



Catalyst 4500 Series Switch Cisco IOS Software Configuration Guide

Cisco IOS XE Release 3.11.xE

Americas Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Customer Order Number:

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

CCVP, the Cisco logo, and Welcome to the Human Network are trademarks of Cisco Systems, Inc.; Changing the Way We Work, Live, Play, and Learn is a service mark of Cisco Systems, Inc.; and Access Registrar, Aironet, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Enterprise/Solver, EtherChannel, EtherFast, EtherSwitch, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, IP/TV, iQ Expertise, the iQ logo, iQ Net Readiness Scorecard, iQuick Study, LightStream, Linksys, MeetingPlace, MGX, Networkers, Networking Academy, Network Registrar, PIX, ProConnect, ScriptShare, SMARTnet, StackWise, The Fastest Way to Increase Your Internet Quotient, and TransPath are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or Website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0711R)

Catalyst 4500 Series Switch Cisco IOS Software Configuration Guide
Copyright © 2019–2020 Cisco Systems, Inc. All rights reserved.



Audience	liii
Conventions	liii
Related Documentation	liv
Notices	lvi
Obtaining Documentation and Submitting a Service Request	i-lviii
Layer 2 Software Features	1-1
802.1Q Tunneling, VLAN Mapping, and Layer 2 Protocol Tunneling	1-2
Cisco IOS Auto Smartport Macros	1-2
Cisco Discovery Protocol	1-3
Cisco Group Management Protocol (CGMP) server	1-3
EtherChannel Bundles	1-3
Ethernet CFM	1-3
Ethernet OAM Protocol	1-3
Flex Links and MAC Address-Table Move Update	1-4
Flexible NetFlow (Supervisor Engine 9-E, 8-E, 8L-E, 7-E, and 7L-E only)	1-4
Internet Group Management Protocol (IGMP) Snooping	1-4
IPv6 Multicast BSR and BSR Scoped Zone Support	1-5
IPv6 Multicast Listen Discovery (MLD) and Multicast Listen Discovery Snooping	1-6
Jumbo Frames	1-6
Link Aggregation Control Protocol	1-7
Link Layer Discovery Protocol	1-7
Link State Tracking	1-8
Location Service	1-8
Multiple Spanning Tree	1-8
Per-VLAN Rapid Spanning Tree	1-8
Quality of Service	1-9
Resilient Ethernet Protocol	1-10
SmartPort Macros	1-10
Spanning Tree Protocol	1-10
Stateful Switchover	1-10
SVI Autostate	1-11
Unidirectional Link Detection	1-11
VLANs	1-11

Virtual Switching Systems (Catalyst 4500-X and Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, and 7-E)	1-12
Virtual Switch System Client	1-12
Y.1731 (AIS and RDI)	1-13
Layer 3 Software Features	1-13
Bidirectional Forwarding Detection	1-14
Cisco Express Forwarding	1-14
Device Sensor	1-14
EIGRP Stub Routing	1-14
Enhanced Object Tracking	1-15
GLBP	1-15
HSRP	1-16
NHRP	1-17
IP Routing Protocols	1-17
In Service Software Upgrade	1-20
IPv6	1-20
Multicast Services	1-20
NSF with SSO	1-21
OSPF for Routed Access	1-22
Policy-Based Routing	1-22
Unicast Reverse Path Forwarding	1-22
Unidirectional Link Routing	1-23
VRF-lite	1-23
Virtual Router Redundancy Protocol	1-23
Management Features	1-23
Cisco Call Home	1-24
Cisco Energy Wise	1-25
Cisco IOS IP Service Level Agreements	1-25
Cisco Media Services Proxy	1-25
Cisco Medianet AutoQoS	1-26
Cisco Medianet Flow Metadata	1-26
Cisco IOS Mediatrace and Performance Monitor	1-27
Cisco Network Assistant	1-28
Dynamic Host Control Protocol	1-28
Easy Virtual Network	1-29
Embedded CiscoView	1-29
Embedded Event Manager	1-29
Ethernet Management Port	1-30
File System Management (Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, and 7-E)	1-30
FAT File Management System on Supervisor Engine 6-E, Supervisor Engine 6L-E	1-30

Forced 10/100 Autonegotiation	1-30
Intelligent Power Management	1-30
MAC Address Notification	1-31
MAC Notify MIB	1-31
Power over Ethernet	1-31
Secure Shell	1-31
Simple Network Management Protocol	1-31
Smart Install	1-32
SPAN and RSPAN	1-32
Universal Power over Ethernet	1-33
Web Content Coordination Protocol	1-33
Wireshark	1-33
XML-PI	1-34
Security Features	1-34
802.1X Identity-Based Network Security	1-35
Cisco TrustSec MACsec Encryption	1-36
Cisco TrustSec Security Architecture	1-36
Cisco TrustSec Security Groups, SGTs and SGACLs	1-37
Dynamic ARP Inspection	1-37
Dynamic Host Configuration Protocol Snooping	1-37
Flood Blocking	1-38
Hardware-Based Control Plane Policing	1-38
IP Source Guard	1-38
IP Source Guard for Static Hosts	1-38
IPv6 First Hop Security	1-39
IPsec VPN	1-40
Local Authentication, RADIUS, and TACACS+ Authentication	1-40
Network Admission Control	1-40
Network Security with ACLs	1-41
Port Security	1-41
PPPoE Intermediate Agent	1-42
Session Aware Networking	1-42
Storm Control	1-42
uRPF Strict Mode	1-43
Utilities	1-43
Web-based Authentication	1-44
Accessing the Switch CLI	2-2
Accessing the CLI Using the EIA/TIA-232 Console Interface	2-2
Accessing the CLI Through Telnet	2-2

Performing Command-Line Processing	2-3
Performing History Substitution	2-4
About Cisco IOS Command Modes	2-4
Getting a List of Commands and Syntax	2-5
Virtual Console for Standby Supervisor Engine	2-6
ROMMON Command-Line Interface	2-7
Archiving Crashfiles Information	2-8
Displaying a Crash Dump for Supervisor Engine 6-E and 6L-E	2-8
Default Switch Configuration	3-1
Configuring DHCP-Based Autoconfiguration	3-2
About DHCP-Based Autoconfiguration	3-2
DHCP Client Request Process	3-3
Configuring the DHCP Server	3-4
Configuring the TFTP Server	3-4
Configuring the DNS Server	3-5
Configuring the Relay Device	3-5
Obtaining Configuration Files	3-6
Example Configuration	3-7
Configuring the Switch	3-8
Using Configuration Mode to Configure Your Switch	3-9
Verifying the Running Configuration Settings	3-9
Saving the Running Configuration Settings to Your Start-Up File	3-10
Reviewing the Configuration in NVRAM	3-10
Configuring a Default Gateway	3-11
Configuring a Static Route	3-11
Controlling Access to Privileged EXEC Commands	3-13
Setting or Changing a Static enable Password	3-13
Using the enable password and enable secret Commands	3-14
Setting or Changing a Privileged Password	3-14
Controlling Switch Access with TACACS+	3-15
Encrypting Passwords	3-22
Configuring Multiple Privilege Levels	3-23
Recovering a Lost Enable Password	3-25
Modifying the Supervisor Engine Startup Configuration	3-25
Understanding the Supervisor Engine Boot Configuration	3-25
Configuring the Software Configuration Register	3-26
Specifying the Startup System Image	3-31
Controlling Environment Variables	3-32

Replacing and Rolling-Back Configuration	3-33
Resetting a Switch to Factory Default Settings	3-34
Managing the System Time and Date	4-1
System Clock	4-2
Understanding Network Time Protocol	4-2
Configuring NTP	4-3
Configuring Time and Date Manually	4-11
Managing Software Licenses Using Right-To-Use Licenses	4-15
RTU License Levels	4-15
RTU License Types	4-16
Ordering with Smart Accounts	4-16
Benefits of an RTU License	4-16
Guidelines for the RTU Licensing Model	4-17
Applying an RTU License	4-17
Activating an RTU License	4-18
Deactivating an RTU License	4-19
Displaying Software License Information	4-19
Configuring a System Name and Prompt	4-25
Understanding the Domain Name System	4-25
Default DNS Configuration	4-26
Setting Up DNS	4-26
Displaying the DNS Configuration	4-27
DNS for IPv6	4-27
IPv6 Router Advertisement Options for DNS Configuration	4-28
Configuring DNS Server Using IPv6 Router Advertisement Options	4-29
Configuring DNS Search List Using IPv6 Router Advertisement Options	4-30
Troubleshooting DNS Servers and DNS Search Lists	4-31
Creating a Banner	4-32
Default Banner Configuration	4-33
Configuring a Message-of-the-Day Login Banner	4-33
Configuring a Login Banner	4-35
Managing the MAC Address Table	4-36
Building the Address Table	4-36
MAC Addresses and VLANs	4-37
Default MAC Address Table Configuration	4-38
Changing the Address Aging Time	4-38
Removing Dynamic Address Entries	4-39
Configuring MAC Change Notification Traps	4-39
Configuring MAC Move Notification Traps	4-41

Configuring MAC Threshold Notification Traps	4-43
Adding and Removing Static Address Entries	4-44
Configuring Unicast MAC Address Filtering	4-45
Disabling MAC Address Learning on a VLAN	4-47
Displaying Address Table Entries	4-52
Managing the ARP Table	4-52
Configuring Embedded CiscoView Support	4-52
Understanding Embedded CiscoView	4-53
Installing and Configuring Embedded CiscoView	4-53
Displaying Embedded CiscoView Information	4-56
Restrictions for Virtual Switching Systems	5-1
Understanding Virtual Switching Systems	5-2
VSS Overview	5-2
VSS Redundancy	5-10
Multichassis EtherChannels	5-13
Packet Handling	5-15
System Monitoring	5-19
Dual-Active Detection	5-23
Configuring a Recovery IP Address	5-25
VSS Initialization	5-26
VSS Configuration Guidelines and Restrictions	5-28
General VSS Restrictions and Guidelines	5-28
Multichassis EtherChannel Restrictions and Guidelines	5-29
Dual-Active Detection Restrictions and Guidelines	5-30
Configuring a VSS	5-30
Configuring Easy VSS	5-30
Converting to a VSS	5-32
Converting to Quad-Supervisor VSS	5-37
Displaying VSS Information	5-39
Converting a VSS to Standalone Switch	5-41
Configuring VSS Parameters	5-42
Configuring Dual-Active Detection	5-52
In-Service Software Upgrade (ISSU) on a VSS	5-56
VSS ISSU Concept	5-56
Traffic and Network Protocol Disruption During ISSU in a VSS	5-57
Related Documents	5-58
Prerequisites to Performing ISSU	5-58
About Performing ISSU	5-59
Compatibility Verification Using Cisco Feature Navigator	5-64

How to Perform the ISSU Process	5-64
License Upgrade on a VSS	5-85
About Programmability	6-1
Overview	6-1
Programmability Components	6-2
Protocols and Data Models for Programmatic Device Configuration	6-2
Default Configuration	6-3
Configuring Programmability Components	6-4
Prerequisites for Configuring Programmability	6-4
Restrictions and Limitations for Configuring Programmability	6-5
Zero-Touch Provisioning Requirements	6-5
Installing and Activating the DMI Container	6-9
Configuring One Platform Kit (OnePK)	6-10
Providing Privilege Access to Use NETCONF and RESTCONF	6-11
Enabling the NETCONF Interface	6-12
Enabling Cisco IOS HTTP Services for RESTCONF	6-13
Using NETCONF and RESTCONF Protocols	6-14
Examples for NETCONF RPCs	6-14
Examples for RESTCONF RPCs	6-15
Using ODM Models	6-15
Enabling SSHv2	6-17
Activating and Deactivating the ODM	6-17
Enabling the Polling Mode	6-19
Displaying Supported Parsers and Polling Intervals	6-20
Monitoring Programmability	6-23
Troubleshooting Programmability	6-25
Sample Configuration and Reference Information	6-28
DHCP Server Settings on Linux	6-28
Configuring DHCP Option 43 (for Microsoft Windows)	6-32
Microsoft Windows DHCP Server Configuration	6-33
Autoboot Process Output	6-37
Prerequisites to Performing ISSU	7-1
About ISSU	7-2
Stateful Switchover Overview	7-3
NSF Overview	7-5
ISSU Process Overview	7-6
Performing an ISSU Upgrade: 2 Methods	7-11
Changeversion Process	7-12
Guidelines for Performing ISSU	7-13

Versioning Capability in Cisco IOS Software to Support ISSU	7-13
SNMP Support for ISSU	7-15
Compatibility Verification Using Cisco Feature Navigator	7-15
Performing the ISSU Process	7-15
Upgrading ISSU to Cisco IOS XE 3.4.0SG/15.1(2)SG from a Prior Release	7-16
Downgrading ISSU from Cisco IOS XE 3.4.0SG/15.1(2)SG to a Prior Release	7-17
Verifying the ISSU Software Installation	7-18
Verifying Redundancy Mode Before Beginning the ISSU Process	7-19
Verifying the ISSU State Before Beginning the ISSU Process	7-20
Loading New Cisco IOS Software on the Standby Supervisor Engine	7-21
Switching to the Standby Supervisor Engine	7-24
Stopping the ISSU Rollback Timer (Optional)	7-26
Loading New Cisco IOS Software on the New Standby Supervisor Engine	7-27
Using changeversion to Automate an ISSU Upgrade	7-29
Aborting a Software Upgrade During ISSU	7-34
Configuring the Rollback Timer to Safeguard Against Upgrade Issues	7-35
Displaying ISSU Compatibility Matrix Information	7-36
Displaying ISSU Compatibility Matrix Information	7-40
Related Documents	7-42
Related Documents	8-2
Prerequisites to Performing ISSU	8-2
About Performing ISSU	8-3
Stateful Switchover	8-4
NSF	8-6
ISSU Process	8-7
Performing an ISSU Upgrade: 2 Methods	8-12
Changeversion Process	8-13
Guidelines for Performing ISSU	8-14
SNMP Support for ISSU	8-15
Compatibility Verification Using Cisco Feature Navigator	8-15
How to Perform the ISSU Process	8-16
Upgrading ISSU to Cisco IOS XE 3.4.0SG/15.1(2)SG from a Prior Release	8-16
Downgrading ISSU from Cisco IOS XE 3.4.0SG/15.1(2)SG to a Prior Release	8-18
Verifying the ISSU Software Installation	8-19
Verifying Redundancy Mode Before Beginning the ISSU Process	8-19
Verifying the ISSU State Before Beginning the ISSU Process	8-21
Loading New Cisco IOS XE Software on the Standby Supervisor Engine	8-21
Switching to the Standby Supervisor Engine	8-25
Stopping the ISSU Rollback Timer (Optional)	8-27

Loading New Cisco IOS XE Software on the New Standby Supervisor Engine	8-28
Using changeversion to Automate an ISSU Upgrade	8-30
Aborting a Software Upgrade During ISSU	8-36
Configuring the Rollback Timer to Safeguard Against Upgrade Issues	8-37
Displaying ISSU Compatibility Matrix Information	8-39
Cisco High Availability Features in Cisco IOS XE 3.1.0SG	8-41
Restrictions for Configuring Interfaces	9-2
About Interface Configuration	9-2
Using the interface Command	9-2
Configuring a Range of Interfaces	9-5
Using the Ethernet Management Port	9-7
Understanding the Ethernet Management Port	9-7
Supported Features on the Ethernet Management Port	9-12
Configuring the Ethernet Management Port	9-13
Defining and Using Interface-Range Macros	9-13
Deploying SFP+ in X2 Ports	9-14
Deploying 10-Gigabit Ethernet and Gigabit Ethernet SFP Ports on Supervisor Engine V-10GE	9-14
Deploying 10-Gigabit Ethernet or Gigabit Ethernet Ports	9-15
Port Numbering TwinGig Convertors	9-15
Limitations on Using a TwinGig Convertor	9-16
Selecting X2/TwinGig Convertor Model	9-16
Configuring MultiGigabit Ports on WS-X4748-12X48U+E	9-18
Module Modes on WS-X4748-12X48U+E Overview	9-18
Configuring Module Modes on WS-X4748-12X48U+E	9-21
Upgrading the Line Card FPGA Image on WS-X4748-12X48U+E	9-21
Invoking Shared-Backplane Uplink Mode on Supervisor Engine 6-E and Supervisor Engine 6L-E	9-22
Selecting Uplink Mode on a Supervisor Engine 6-E	9-23
Support for WS-X46490-CSFP-E on a 10-slot Chassis	9-23
Limitation and Restrictions on Supervisor Engine 7-E and Supervisor Engine 7L-E	9-24
Selecting the Uplink Port on a Supervisor Engine 7L-E	9-24
Single Supervisor Mode	9-25
Redundant Supervisor Mode	9-25
Limitations and Restrictions on Supervisor Engine 8-E	9-25
Configuring Supervisor Engine 7-E Uplink Mode on Supervisor Engine 8-E	9-26
Supervisor Engine 7-E Mode on Supervisor Engine 8-E	9-26
Supervisor Engine 8-E with Daughter Card Enabled	9-26
Supervisor Engine 8-E Uplink Configurations	9-27

Restrictions for Configuring Sup 7-E Uplink Mode on Supervisor Engine 8-E	9-28
Configuring Supervisor Engine 7-E Mode on Supervisor Engine 8-E	9-28
Supervisor Engine 9-E Uplink Configurations	9-29
Limitations and Restrictions on Supervisor Engine 9-E	9-30
Selecting the Uplink Mode on Supervisor Engine 9-E	9-31
Digital Optical Monitoring Transceiver Support	9-32
Configuring Optional Interface Features	9-32
Configuring Ethernet Interface Speed and Duplex Mode	9-32
Configuring Flow Control	9-38
Configuring Jumbo Frame Support	9-40
Interacting with Baby Giants	9-44
Configuring the Port Debounce Timer	9-44
Configuring Auto-MDIX on a Port	9-45
Understanding Online Insertion and Removal	9-47
Online Insertion and Removal on a WS-4500X-32	9-48
Shutting down a Module	9-48
Booting a Module After if it has been Stopped	9-49
Common Scenarios	9-50
Monitoring and Maintaining the Interface	9-50
Monitoring Interface and Controller Status	9-50
Clearing and Resetting the Interface	9-51
Shutting Down and Restarting an Interface	9-51
Configuring Interface Link Status and Trunk Status Events	9-52
Resetting the Interface to the Default Configuration	9-55
Checking Module Status	10-1
Checking Interfaces Status	10-2
Displaying MAC Addresses	10-3
Checking Cable Status Using Time Domain Reflectometer	10-3
Overview	10-3
Running the TDR Test	10-4
TDR Guidelines	10-5
Using Telnet	10-5
Changing the Logout Timer	10-6
Monitoring User Sessions	10-6
Using Ping	10-7
Understanding How Ping Works	10-7
Running Ping	10-8
Using IP Traceroute	10-8

Understanding How IP Traceroute Works	10-8
Running IP Traceroute	10-9
Using Layer 2 Traceroute	10-9
Layer 2 Traceroute Usage Guidelines	10-10
Running Layer 2 Traceroute	10-11
Configuring ICMP	10-11
Enabling ICMP Protocol Unreachable Messages	10-12
Enabling ICMP Redirect Messages	10-12
Enabling ICMP Mask Reply Messages	10-13
About Supervisor Engine Redundancy	11-2
Overview	11-2
RPR Operation	11-2
SSO Operation	11-3
About Supervisor Engine Redundancy Synchronization	11-4
RPR Supervisor Engine Configuration Synchronization	11-4
SSO Supervisor Engine Configuration Synchronization	11-5
Supervisor Engine Redundancy Guidelines and Restrictions	11-5
Configuring Supervisor Engine Redundancy	11-7
Configuring Redundancy	11-8
Virtual Console for Standby Supervisor Engine	11-10
Synchronizing the Supervisor Engine Configurations	11-11
Performing a Manual Switchover	11-12
Performing a Software Upgrade	11-13
Manipulating Bootflash on the Redundant Supervisor Engine	11-14
About Supervisor Engine Redundancy	12-2
Overview	12-2
RPR Operation	12-2
SSO Operation	12-3
About Supervisor Engine Redundancy Synchronization	12-4
RPR Supervisor Engine Configuration Synchronization	12-5
SSO Supervisor Engine Configuration Synchronization	12-5
Supervisor Engine Redundancy Guidelines and Restrictions	12-5
Configuring Supervisor Engine Redundancy	12-7
Configuring Redundancy	12-7
Virtual Console for Standby Supervisor Engine	12-9
Synchronizing the Supervisor Engine Configurations	12-10
Performing a Manual Switchover	12-11
Performing a Software Upgrade	12-12

Manipulating Bootflash on the Standby Supervisor Engine	12-14
About NSF with SSO Supervisor Engine Redundancy	13-1
About Cisco IOS NSF-Aware and NSF-Capable Support	13-2
NSF with SSO Supervisor Engine Redundancy Overview	13-3
SSO Operation	13-4
NSF Operation	13-4
Cisco Express Forwarding	13-5
Routing Protocols	13-5
NSF Guidelines and Restrictions	13-9
Configuring NSF with SSO Supervisor Engine Redundancy	13-9
Configuring SSO	13-10
Configuring CEF NSF	13-11
Verifying CEF NSF	13-11
Configuring BGP NSF	13-11
Verifying BGP NSF	13-12
Configuring OSPF NSF	13-13
Verifying OSPF NSF	13-13
Configuring IS-IS NSF	13-14
Verifying IS-IS NSF	13-15
Configuring EIGRP NSF	13-16
Verifying EIGRP NSF	13-16
Cisco High Availability Features in Cisco IOS XE 3.1.0SG	13-17
About Environmental Monitoring	14-1
Using CLI Commands to Monitor your Environment	14-2
Displaying Environment Conditions	14-2
Displaying On Board Failure Logging (OBFL) information for 9000W AC	14-4
Emergency Actions	14-5
System Alarms	14-6
Power Management	14-7
Power Management for the Catalyst 4500 series switches	14-7
Powering Down a Module	14-22
IEEE 802.3az Energy Efficient Ethernet	14-23
Determining EEE Capability	14-23
Enabling EEE	14-24
Determining EEE Status	14-24
About Power over Ethernet	15-2
Hardware Requirements	15-2
Power Management Modes	15-3
Intelligent Power Management	15-5

Configuring Power Consumption for Powered Devices on an Interface	15-5
Displaying the Operational Status for an Interface	15-7
Displaying all PoE Detection and Removal Events	15-8
Displaying the PoE Consumed by a Module	15-8
PoE Policing and Monitoring	15-12
PoE Policing Modes	15-13
Configuring Power Policing on an Interface	15-13
Displaying Power Policing on an Interface	15-14
Configuring Errdisable Recovery	15-15
Configuring Universal PoE	15-16
Additional References for Power over Ethernet	15-18
Related Documents	15-18
MIBs	15-18
Technical Assistance	15-18
Feature Information for Power over Ethernet	15-18
About Network Assistant	16-2
Community Overview	16-2
Clustering Overview	16-2
Network Assistant-Related Parameters and Their Defaults	16-2
Network Assistant CLI Commands	16-3
Configuring Your Switch for Network Assistant	16-4
(Minimum) Required Configuration	16-4
(Additional) Configuration Required to Use Community	16-5
(Additional) Configuration Required to Use Clustering	16-5
Managing a Network Using Community	16-6
Candidate and Member Requirements	16-7
Automatic Discovery of Candidates and Members	16-7
Community Names	16-8
Hostnames	16-8
Passwords	16-8
Communication Protocols	16-8
Access Modes in Network Assistant	16-9
Community Information	16-9
Adding Devices	16-9
Converting a Cluster into a Community	16-10
Managing a Network Using Cluster	16-11
Understanding Switch Clusters	16-11
Using the CLI to Manage Switch Clusters	16-13

Configuring Network Assistant in Community or Cluster Mode	16-13
Configuring Network Assistant on a Networked Switch in Community Mode	16-13
Configuring Network Assistant in a Networked Switch in Cluster Mode	16-17
VLANs	17-1
About VLANs	17-1
VLAN Configuration Guidelines and Restrictions	17-3
VLAN Default Configuration	17-4
Configuring VLANs	17-5
VLAN Trunking Protocol	17-7
About VTP	17-7
VTP Configuration Guidelines and Restrictions	17-12
VTP Default Configuration	17-13
Configuring VTP	17-13
VLAN Membership Policy Server	17-20
About VMPS	17-20
Overview of VMPS Clients	17-22
Dynamic Port VLAN Membership Configuration Example	17-28
VMPS Database Configuration File Example	17-31
About IP Unnumbered Interface Support	18-1
IP Unnumbered Interface Support with DHCP Server and Relay Agent	18-2
DHCP Option 82	18-2
IP Unnumbered Interface with Connected Host Polling	18-3
IP Unnumbered Configuration Guidelines and Restrictions	18-3
Configuring IP Unnumbered Interface Support with DHCP Server	18-4
Configuring IP Unnumbered Interface Support on LAN and VLAN Interfaces	18-4
Configuring IP Unnumbered Interface Support on a Range of Ethernet VLANs	18-5
Configuring IP Unnumbered Interface Support with Connected Host Polling	18-6
Displaying IP Unnumbered Interface Settings	18-7
Troubleshooting IP Unnumbered Interface	18-8
Related Documents	18-8
About Layer 2 Ethernet Switching	19-1
Layer 2 Ethernet Switching	19-2
VLAN Trunks	19-3
Layer 2 Interface Modes	19-3
Default Layer 2 Ethernet Interface Configuration	19-4
Layer 2 Interface Configuration Guidelines and Restrictions	19-4
Configuring Ethernet Interfaces for Layer 2 Switching	19-5
Configuring an Ethernet Interface as a Layer 2 Trunk	19-5

Configuring an Interface as a Layer 2 Access Port	19-7
Clearing Layer 2 Configuration	19-8
About EVC-Lite	20-1
How to Configure EVC-Lite	20-2
About SmartPort Macros and Static SmartPort	21-1
Configuring SmartPort Macros	21-2
Passing Parameters Through the Macro	21-2
Default SmartPort Macro Configuration	21-3
SmartPort Macro Configuration Guidelines	21-6
Creating SmartPort Macros	21-7
Applying SmartPort Macros	21-8
Displaying SmartPort Macros	21-12
Configuring Static SmartPort Macros	21-13
Default Static SmartPort Configuration	21-13
Static SmartPort Configuration Guidelines	21-13
Applying Static SmartPort Macros	21-14
About Auto Smartport Macros	22-1
Device Classifier	22-2
Configuring Auto Smartport Macros	22-3
Enabling Auto Smartport Macros	22-3
Auto Smartport Default Configuration	22-4
Auto Smartport Configuration Guidelines	22-5
Configuring Auto Smartport Built-in Macro Parameters	22-6
Configuring Mapping Between Event Triggers and Built-in Macros	22-8
Configuring User-Defined Event Triggers	22-9
Configuring Mapping Between User-Defined Triggers and Built-in Macros	22-10
Configuring Auto Smartport User-Defined Macros	22-11
Displaying Auto Smartport	22-14
About STP	23-1
Understanding the Bridge ID	23-2
Bridge Protocol Data Units	23-3
Election of the Root Bridge	23-4
STP Timers	23-4
Creating the STP Topology	23-5
STP Port States	23-5
MAC Address Allocation	23-6
STP and IEEE 802.1Q Trunks	23-6
Per-VLAN Rapid Spanning Tree	23-6

Default STP Configuration	23-7
Configuring STP	23-7
Enabling STP	23-8
Enabling the Extended System ID	23-9
Configuring the Root Bridge	23-9
Configuring a Secondary Root Switch	23-12
Configuring STP Port Priority	23-13
Configuring STP Port Cost	23-15
Configuring the Bridge Priority of a VLAN	23-17
Configuring the Hello Time	23-17
Configuring the Maximum Aging Time for a VLAN	23-18
Configuring the Forward-Delay Time for a VLAN	23-19
Disabling Spanning Tree Protocol	23-20
Enabling Per-VLAN Rapid Spanning Tree	23-20
About MST	23-22
IEEE 802.1s MST	23-22
IEEE 802.1w RSTP	23-23
MST-to-SST Interoperability	23-24
Common Spanning Tree	23-25
MST Instances	23-26
MST Configuration Parameters	23-26
MST Regions	23-26
Message Age and Hop Count	23-28
MST Configuration Restrictions and Guidelines	23-28
Configuring MST	23-28
Enabling MST	23-29
Configuring MST Instance Parameters	23-30
Configuring MST Instance Port Parameters	23-31
Restarting Protocol Migration	23-32
Displaying MST Configurations	23-32
About MST-to-PVST+ Interoperability (PVST+ Simulation)	23-35
Configuring PVST+ Simulation	23-36
About Detecting Unidirectional Link Failure	23-40
About Flex Links	24-1
Flex Links	24-1
VLAN Flex Links Load Balancing and Support	24-2
Flex Links Failover Actions	24-3
MAC Address-Table Move Update	24-3
Configuring Flex Links	24-5

Default Configuration	24-5
Configuration Guidelines	24-5
Configuring Flex Links	24-6
Configuring VLAN Load Balancing on Flex Links	24-8
Configuring MAC Address-Table Move Update	24-9
Default Configuration	24-9
Configuration Guidelines	24-9
Configuring the MAC Address-Table Move Update Feature	24-10
Monitoring Flex Links and the MAC Address-Table Move Update	24-12
24-12	
About REP	25-1
Link Integrity	25-4
Fast Convergence	25-4
VLAN Load Balancing	25-4
Spanning Tree Interaction	25-6
REP Ports	25-6
Configuring REP	25-7
Default REP Configuration	25-7
REP Configuration Guidelines	25-7
Configuring the REP Administrative VLAN	25-8
Configuring REP Interfaces	25-10
Setting Manual Preemption for VLAN Load Balancing	25-13
Configuring SNMP Traps for REP	25-14
Monitoring REP	25-14
About Root Guard	26-2
Enabling Root Guard	26-2
About Loop Guard	26-3
Enabling Loop Guard	26-5
About EtherChannel Guard	26-6
Enabling EtherChannel Guard (Optional)	26-6
About STP PortFast Port Types	26-7
Enabling PortFast Port Types	26-8
Configuring the PortFast Default State Globally	26-8
Configuring a PortFast Edge Port on a Specified Interface	26-8
Configuring a PortFast Network Port on a Specified Interface	26-10
About Bridge Assurance	26-11
Configuring Bridge Assurance	26-13
About BPDU Guard	26-15

Enabling BPDU Guard	26-15
Enabling BPDU Guard Globally	26-15
Enabling BPDU Guard on a Specified Interface	26-16
About PortFast Edge BPDU Filtering	26-16
Enabling PortFast Edge BPDU Filtering	26-17
Enabling PortFast Edge BPDU Filtering Globally	26-17
Enabling PortFast Edge BPDU Filtering on a Specified Interface	26-18
About UplinkFast	26-19
Enabling UplinkFast	26-20
About BackboneFast	26-21
Enabling BackboneFast	26-23
About EtherChannel	27-1
Port Channel Interfaces	27-2
Configuring EtherChannels	27-2
Load Balancing	27-6
EtherChannel Configuration Guidelines and Restrictions	27-6
Configuring EtherChannel	27-7
Configuring Layer 3 EtherChannels	27-7
Configuring Layer 2 EtherChannels	27-11
Configuring LACP Standalone or Independent Mode	27-13
Configuring LACP Port Channel Min-links	27-14
Configuring the LACP System Priority and System ID	27-16
Configuring LACP Fast Rate Timer	27-17
Configuring Auto-LAG Globally	27-17
Configuring Auto-LAG on a Port Interface	27-18
Configuring Persistence with Auto-LAG	27-18
Configuring EtherChannel Load Balancing	27-18
Removing an Interface from an EtherChannel	27-19
Removing an EtherChannel	27-20
Displaying EtherChannel to a Virtual Switch System	27-20
Understanding VSS Client	27-21
Displaying EtherChannel Links to VSS	27-23
Understanding Link-State Tracking	27-23
Configuring Link-State Tracking	27-26
Default Link-State Tracking Configuration	27-26
Link-State Tracking Configuration Guidelines	27-26
Configuring Link-State Tracking	27-26
Displaying Link-State Tracking Status	27-27

About IGMP Snooping	28-1
Immediate-Leave Processing	28-3
IGMP Configurable-Leave Timer	28-4
IGMP Snooping Querier	28-4
Explicit Host Tracking	28-4
Configuring IGMP Snooping	28-5
Default IGMP Snooping Configuration	28-5
Enabling IGMP Snooping Globally	28-6
Enabling IGMP Snooping on a VLAN	28-6
Configuring Learning Methods	28-7
Configuring a Static Connection to a Multicast Router	28-8
Enabling IGMP Immediate-Leave Processing	28-8
Configuring the IGMP Leave Timer	28-9
Configuring IGMP Snooping Querier	28-10
Configuring Explicit Host Tracking	28-11
Configuring a Host Statically	28-11
Suppressing Multicast Flooding	28-12
Displaying IGMP Snooping Information	28-14
Displaying Querier Information	28-15
Displaying IGMP Host Membership Information	28-15
Displaying Group Information	28-16
Displaying Multicast Router Interfaces	28-17
Displaying MAC Address Multicast Entries	28-18
Displaying IGMP Snooping Information on a VLAN Interface	28-18
Displaying IGMP Snooping Querier Information	28-19
Understanding Multicast VLAN Registration	28-20
Using MVR in a Multicast Television Application	28-21
Configuring MVR	28-23
Default MVR Configuration	28-23
MVR Configuration Guidelines and Limitations	28-23
Configuring MVR Global Parameters	28-24
Configuring MVR on Access Ports	28-26
Configuring MVR on a Trunk Port	28-27
Displaying MVR Information	28-29
Configuring IGMP Filtering	28-30
Default IGMP Filtering Configuration	28-30
Configuring IGMP Profiles	28-31
Applying IGMP Profiles	28-32
Setting the Maximum Number of IGMP Groups	28-33

Displaying IGMP Filtering Configuration	28-34
Feature Overview	29-1
Dual-Homed Remote	29-3
Benefits	29-5
Restrictions	29-5
Related Features and Technologies	29-5
Supported Platforms	29-6
Supported Standards, MIBs, and RFCs	29-6
Configuration Tasks	29-7
Configuring EIGRP Stub Routing	29-7
Verifying EIGRP Stub Routing	29-7
Monitoring and Maintaining EIGRP Stub Routing	29-8
Configuration Examples	29-8
About MLD Snooping	30-1
MLD Messages	30-2
MLD Queries	30-3
Multicast Client Aging	30-3
Multicast Router Discovery	30-3
MLD Reports	30-4
MLD Done Messages and Immediate-Leave	30-4
Topology Change Notification Processing	30-4
Configuring IPv6 MLD Snooping	30-5
Default MLD Snooping Configuration	30-5
MLD Snooping Configuration Guidelines	30-6
Enabling or Disabling MLD Snooping	30-6
Configuring a Static Multicast Group	30-7
Configuring a Multicast Router Port	30-7
Enabling MLD Immediate Leave	30-8
Configuring MLD Snooping Queries	30-9
Disabling MLD Listener Message Suppression	30-10
Displaying MLD Snooping Information	30-10
About 802.1Q Tunneling	31-1
Configuring 802.1Q Tunneling	31-3
802.1Q Tunneling Configuration Guidelines	31-3
802.1Q Tunneling and Other Features	31-5
Configuring an 802.1Q Tunneling Port	31-5
About VLAN Mapping	31-6
Deployment Example	31-7

Mapping Customer VLANs to Service-Provider VLANs	31-8
Configuring VLAN Mapping	31-9
Default VLAN Mapping Configuration	31-9
VLAN Mapping Configuration Guidelines	31-9
Configuring VLAN Mapping	31-10
About Layer 2 Protocol Tunneling	31-13
Configuring Layer 2 Protocol Tunneling	31-15
Default Layer 2 Protocol Tunneling Configuration	31-16
Layer 2 Protocol Tunneling Configuration Guidelines	31-16
Configuring Layer 2 Tunneling	31-17
Configuring Layer 2 Tunneling for EtherChannels	31-19
Monitoring and Maintaining Tunneling Status	31-23
About Cisco Discovery Protocol	32-2
Configuring Cisco Discovery Protocol	32-2
Enabling Cisco Discovery Protocol Globally	32-2
Displaying the Cisco Discovery Protocol Global Configuration	32-2
Enabling Cisco Discovery Protocol on an Interface	32-3
Displaying the Cisco Discovery Protocol Interface Configuration	32-3
Monitoring and Maintaining Cisco Discovery Protocol	32-4
About Cisco Discovery Protocol Bypass	32-5
Configuring Cisco Discovery Protocol Bypass	32-5
Enabling Cisco Discovery Protocol Bypass	32-6
Displaying Cisco Discovery Protocol Neighbors	32-7
Disabling Cisco Discovery Protocol Bypass	32-7
About LLDP, LLDP-MED, and Location Service	33-1
Restrictions for LLDP	33-1
LLDP	33-2
LLDP-MED	33-2
Location Service	33-3
Configuring LLDP and LLDP-MED, and Location Service	33-4
Default LLDP Configuration	33-5
Configuring LLDP Characteristics	33-5
Disabling and Enabling LLDP Globally	33-6
Disabling and Enabling LLDP on an Interface	33-7
Configuring LLDP-MED TLVs	33-8
Configuring Network-Policy Profile	33-9
Configuring LLDP Power Negotiation	33-11
Configuring Location TLV and Location Service	33-12

Monitoring and Maintaining LLDP, LLDP-MED, and Location Service	33-14
Cisco IOS Carries Ethernet Features in Cisco IOS XE 3.1.0SG	33-15
About UDLD	34-1
UDLD Topology	34-2
Fast UDLD Topology	34-2
Operation Modes	34-3
Default States for UDLD	34-3
Default UDLD Configuration	34-3
Configuring UDLD on the Switch	34-4
Fast UDLD Guidelines and Restrictions	34-4
Enabling UDLD Globally	34-5
Enabling UDLD on Individual Interfaces	34-6
Disabling UDLD on Individual Interfaces	34-7
Disabling UDLD on a Fiber-Optic Interface	34-7
Configuring a UDLD Probe Message Interval Globally	34-8
Enabling Fast UDLD Error Reporting	34-8
Resetting Disabled LAN Interfaces	34-8
Displaying UDLD Link Status	34-9
About Unidirectional Ethernet	35-1
Configuring Unidirectional Ethernet	35-1
About Layer 3 Interfaces	36-1
Logical Layer 3 VLAN Interfaces	36-2
Physical Layer 3 Interfaces	36-2
Understanding SVI Autostate Exclude	36-3
Understanding Layer 3 Interface Counters	36-3
Configuration Guidelines	36-5
Configuring Logical Layer 3 GRE Tunnel Interfaces	36-6
Configuring Logical Layer 3 VLAN Interfaces	36-7
Configuring VLANs as Layer 3 Interfaces	36-8
Configuring SVI Autostate Exclude	36-9
Configuring IP MTU Sizes	36-10
Configuring Layer 3 Interface Counters	36-11
Configuring Physical Layer 3 Interfaces	36-13
Configuring Multipoint GRE	36-14
About Multipoint GRE	36-14
Configuring Unicast mGRE at Hub	36-16
Configuring Unicast mGRE at Spoke	36-18
Sample mGRE Configuration at Hub and Spokes	36-18

About CEF	37-1
CEF Features	37-1
Forwarding Information Base	37-2
Adjacency Tables	37-2
Catalyst 4500 Series Switch Implementation of CEF	37-3
Hardware and Software Switching	37-4
Load Balancing	37-6
Software Interfaces	37-6
CEF Configuration Restrictions	37-6
Configuring CEF	37-6
Enabling CEF	37-6
Configuring Load Balancing for CEF	37-7
Monitoring and Maintaining CEF	37-8
Displaying IP Statistics	37-8
About Unicast Reverse Path Forwarding	38-1
How Unicast RPF Works	38-2
Unicast Reverse Path Forwarding with ACL Support	38-4
Implementing Unicast RPF	38-4
Restrictions	38-8
Limitation	38-8
Related Features and Technologies	38-8
Prerequisites to Configuring Unicast RPF	38-9
Unicast RPF Configuration Tasks	38-9
Configuring Unicast RPF	38-9
Configuring Unicast RPF with ACL Support	38-10
38-11	
Verifying Unicast RPF	38-11
Monitoring and Maintaining Unicast RPF	38-11
Unicast RPF Configuration Example: Inbound and Outbound Filters	38-12
Unicast RPF with ACL Support Configuration Example:	38-13
Feature Information for Unicast Reverse Path Forwarding	38-13
38-14	
About IP Multicast	39-1
IP Multicast Protocols	39-2
IP Multicast Implementation on the Catalyst 4500 Series Switch	39-4
Configuring IP Multicast Routing	39-13
Default Configuration in IP Multicast Routing	39-14
Enabling IP Multicast Routing	39-14

Enabling PIM on an Interface	39-15
Enabling Bidirectional Mode	39-16
Enabling PIM-SSM Mapping	39-17
Configuring a Rendezvous Point	39-17
Configuring a Single Static RP	39-21
Load Splitting of IP Multicast Traffic	39-22
Monitoring and Maintaining IP Multicast Routing	39-23
Displaying System and Network Statistics	39-24
Displaying the Multicast Routing Table	39-24
Displaying IP MFIB	39-26
Displaying Bidirectional PIM Information	39-27
Displaying PIM Statistics	39-28
Clearing Tables and Databases	39-28
Configuration Examples	39-29
PIM Dense Mode Example	39-29
PIM Sparse Mode Example	39-29
Bidirectional PIM Mode Example	39-29
Sparse Mode with a Single Static RP Example	39-30
Sparse Mode with Auto-RP: Example	39-30
About ANCP Client	40-1
Enabling and Configuring ANCP Client	40-2
Identifying a Port with the ANCP Protocol	40-2
Identifying a Port with DHCP Option 82	40-4
ANCP Guidelines and Restrictions	40-5
Finding Feature Information	41-1
Contents	41-2
Prerequisites for Bidirectional Forwarding Detection	41-2
Restrictions for Bidirectional Forwarding Detection	41-2
Information About Bidirectional Forwarding Detection	41-3
BFD Operation	41-3
Benefits of Using BFD for Failure Detection	41-7
Hardware Support for BFD	41-7
How to Configure Bidirectional Forwarding Detection	41-8
Configuring BFD Session Parameters on the Interface	41-8
Configuring BFD Support for Dynamic Routing Protocols	41-9
Configuring BFD Support for Static Routing	41-14
Configuring BFD Echo Mode	41-15
Monitoring and Troubleshooting BFD	41-17

Configuration Examples for Bidirectional Forwarding Detection	41-17
Example: Configuring BFD in an EIGRP Network with Echo Mode Enabled by Default	41-17
Example: Configuring BFD in an OSPF Network	41-22
Example: Configuring BFD Hardware-Offload support in a BGP Network Network	41-25
Example: Configuring BFD Support for Static Routing	41-27
Additional References	41-28
Related Documents	41-28
Standards	41-28
MIBs	41-29
RFCs	41-29
Technical Assistance	41-29
Policy-Based Routing	42-1
Route Maps	42-2
Policy-Based Routing with Object Tracking	42-6
IPv4 and IPv6 Policy-Based Routing for VRF Instances	42-6
Policy-Based Routing Configuration Tasks	42-7
Enabling IPv4 PBR	42-7
Enabling IPv6 PBR	42-10
Enabling Local IPv4 and Local IPv6 PBR	42-12
Configuring IPv4 and IPv6 PBR for VRF Instances	42-12
Verifying Next-Hop IP using Object Tracking	42-14
Unsupported Commands	42-15
Policy-Based Routing Configuration Examples	42-16
Equal Access	42-16
Differing Next Hops	42-17
Deny ACE	42-17
Examples of the show Command	42-17
About VRF-lite	43-2
VRF-lite Configuration Guidelines	43-3
Configuring VRF-lite for IPv4	43-5
Configuring VRFs	43-5
Configuring VRF-Aware Services	43-6
Configuring Per-VRF for TACACS+ Servers	43-6
Configuring Multicast VRFs	43-7
Configuring a VPN Routing Session	43-8
Configuring BGP PE to CE Routing Sessions	43-9
VRF-lite Configuration Example	43-10
Displaying VRF-lite Status	43-14
Configuring VRF-lite for IPv6	43-15

Configuring VRF-Aware Services	43-15
VRF-lite Configuration Example	43-17
Displaying VRF-lite Status	43-21
Configuring IPv6 VRF-lite	43-22
VPN Co-existence Between IPv4 and IPv6	43-28
Migrating from the Old to New CLI Scheme	43-28
Overview of QoS	44-1
Prioritization	44-2
QoS Terminology	44-3
Basic QoS Model	44-5
Classification	44-6
Policing and Marking	44-8
Queueing and Scheduling	44-8
Packet Modification	44-9
Per Port Per VLAN QoS	44-10
Flow-based QoS	44-10
Using Metadata in QoS Policy	44-11
Configuring System Queue Limit	44-12
Configuring QoS with a 40-Gigabit Ethernet Interface	44-13
Configuring VSS QoS	44-14
MQC-based QoS Configuration	44-14
Platform-supported Classification Criteria and QoS Features	44-15
Platform Hardware Capabilities	44-16
Prerequisites for Applying a QoS Service Policy	44-16
Restrictions for Applying a QoS Service Policy	44-16
Classification	44-16
Policing	44-18
Marking Network Traffic	44-19
Shaping, Sharing (Bandwidth), Priority Queueing, Queue-limiting and DBL	44-26
Enabling Per-Port Per-VLAN QoS	44-37
Applying Flow-based QoS Policy	44-42
Configuring CoS Mutation	44-46
Configuring System Queue Limit	44-47
Configuring QoS on a Standalone Supervisor Engine 6-E, 6L-E or Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, and 7-E	44-48
MQC-based QoS Configuration	44-49
Platform-supported Classification Criteria and QoS Features	44-49
Platform Hardware Capabilities	44-50
Prerequisites for Applying a QoS Service Policy	44-50

Restrictions for Applying a QoS Service Policy	44-51
Classification	44-51
Policing	44-52
Marking Network Traffic	44-53
Shaping, Sharing (Bandwidth), Priority Queuing, Queue-limiting and DBL	44-60
Enabling Per-Port Per-VLAN QoS	44-71
Applying Flow-based QoS Policy	44-76
Configuring CoS Mutation	44-80
Configuring System Queue Limit	44-81
Configuring VSS Auto-QoS	44-82
Auto-QoS Overview	44-82
Auto-QoS Policy and Class Maps	44-83
Auto-Qos Compact	44-88
Effects of Auto-QoS and Auto-Qos Compact on Running Configuration	44-89
Configuring Auto-QoS on a Standalone Supervisor Engine 6-E/6L-E or Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, and 7-E	44-89
Auto-QoS Overview	44-89
Auto-QoS Policy and Class Maps	44-90
Auto-QoS Compact	44-97
Effects of Auto-QoS and Auto-Qos Compact on Running Configuration	44-98
About AVC with DNS-AS	45-3
Overview	45-3
Key Concepts	45-3
AVC with DNS-AS Process Flow	45-5
High Availability and ISSU for AVC with DNS-AS	45-6
Default Configuration	45-7
Configuring AVC with DNS-AS	45-7
Prerequisites for Configuring AVC with DNS-AS	45-7
Restrictions and Guidelines for Configuring AVC with DNS-AS	45-7
Generating Metadata Streams	45-8
Configuring a DNS Server as the Authoritative Server	45-10
Enabling AVC with DNS-AS	45-10
Making an Entry in the Trusted Domain List	45-11
Configuring QoS for AVC with DNS-AS	45-12
Configuring FNF for AVC with DNS-AS	45-16
Monitoring AVC with DNS-AS	45-21
Troubleshooting AVC with DNS-AS	45-25
About Voice Interfaces	46-1
Cisco IP Phone Voice Traffic	46-2

Cisco IP Phone Data Traffic	46-2
Configuring a Port to Connect to a Cisco 7960 IP Phone	46-2
Configuring Voice Ports for Voice and Data Traffic	46-3
Overriding the CoS Priority of Incoming Frames	46-4
Configuring Power	46-5
About Private VLANs	47-1
Purpose of a PVLAN	47-2
PVLAN Terminology	47-3
PVLANs across Multiple Switches	47-5
PVLAN Modes Over Gigabit Etherchannel	47-8
Private-VLAN Interaction with Other Features	47-8
PVLAN Commands	47-10
Configuring PVLANS	47-11
Basic PVLAN Configuration Procedure	47-12
Default Private-VLAN Configuration	47-12
PVLAN Configuration Guidelines and Restrictions	47-12
Configuring a VLAN as a PVLAN	47-15
Associating a Secondary VLAN with a Primary VLAN	47-16
Configuring a Layer 2 Interface as a PVLAN Promiscuous Port	47-17
Configuring a Layer 2 Interface as a PVLAN Host Port	47-18
Configuring a Layer 2 Interface as an Isolated PVLAN Trunk Port	47-19
Configuring a Layer 2 Interface as a Promiscuous PVLAN Trunk Port	47-21
Permitting Routing of Secondary VLAN Ingress Traffic	47-23
Configuring PVLAN over EtherChannel	47-24
Understanding Media Access Control Security and MACsec Key Agreement	48-2
MKA Policies	48-3
Key Lifetime and Hitless Key Rollover	48-3
Encryption Algorithms for MKA Control Packets	48-4
Virtual Ports	48-4
MACsec	48-5
MACsec, MKA, and 802.1X Host Modes	48-5
Configuring MACsec and MACsec Key Agreement	48-7
Default MKA MACsec Configuration	48-7
Configuring an MKA Policy	48-7
Configuring MACsec on an Interface	48-9
Configuring MKA Pre-Shared Key	48-10
Example: Connectivity Association Key Rekey	48-11
Understanding MKA MACsec with EAP-TLS	48-12

Prerequisites for MKA MACsec with EAP-TLS	48-12
Limitations for MKA MACsec with EAP-TLS	48-12
Understanding Certificate Enrollment	48-13
Configuring MKA MACsec Using EAP-TLS	48-16
Understanding Cisco TrustSec MACsec	48-20
Configuring Cisco TrustSec MACsec	48-22
Configuring Cisco TrustSec Credentials on the Switch	48-22
Configuring Cisco TrustSec Switch-to-Switch Link Security in 802.1X Mode	48-23
Configuring Cisco TrustSec Switch-to-Switch Link Security in Manual Mode	48-24
Cisco TrustSec Switch-to-Switch Link Security Configuration Example	48-25
About 802.1X Port-Based Authentication	49-1
Device Roles	49-2
802.1X and Network Access Control	49-3
Authentication Initiation and Message Exchange	49-4
Ports in Authorized and Unauthorized States	49-5
802.1X Host Mode	49-6
802.1X Violation Mode	49-9
Using MAC Move	49-9
Using MAC Replace	49-9
Using 802.1X with VLAN Assignment	49-10
Using 802.1X for Guest VLANs	49-11
Using 802.1X with MAC Authentication Bypass	49-12
Using 802.1X with Web-Based Authentication	49-14
Using 802.1X with Inaccessible Authentication Bypass	49-14
Using 802.1X with Unidirectional Controlled Port	49-15
Using 802.1X with VLAN User Distribution	49-16
Using 802.1X with Authentication Failed VLAN Assignment	49-17
Using 802.1X with Port Security	49-19
Using 802.1X Authentication with ACL Assignments and Redirect URLs	49-19
Using 802.1X with RADIUS-Provided Session Timeouts	49-20
Using 802.1X with Voice VLAN Ports	49-21
Using Voice Aware 802.1x Security	49-21
Using Multiple Domain Authentication and Multiple Authentication	49-22
Limiting Login for Users	49-23
802.1X Supplicant and Authenticator Switches with Network Edge Access Topology	49-23
How 802.1X Fails on a Port	49-24
Supported Topologies	49-25
Configuring 802.1X Port-Based Authentication	49-26
Default 802.1X Configuration	49-27

802.1X Configuration Guidelines	49-28
Enabling 802.1X Authentication	49-28
Configuring Switch-to-RADIUS-Server Communication	49-32
Configuring Multiple Domain Authentication and Multiple Authorization	49-33
Configuring Limiting Login for Users	49-37
Configuring 802.1X Authentication with ACL Assignments and Redirect URLs	49-37
Configuring 802.1X Authentication with Per-User ACL and Filter-ID ACL	49-46
Configuring RADIUS-Provided Session Timeouts	49-55
Configuring MAC Move	49-56
Configuring MAC Replace	49-57
Configuring Violation Action	49-58
Configuring 802.1X with Guest VLANs	49-58
Configuring 802.1X with MAC Authentication Bypass	49-62
Configuring 802.1X with Inaccessible Authentication Bypass	49-64
Configuring 802.1X with Unidirectional Controlled Port	49-68
Configuring 802.1X with VLAN User Distribution	49-69
Configuring 802.1X with Authentication Failed	49-72
Configuring 802.1X with Voice VLAN	49-74
Configuring Voice Aware 802.1x Security	49-75
Configuring 802.1X with VLAN Assignment	49-77
Enabling Fallback Authentication	49-79
Enabling Periodic Reauthentication	49-83
Enabling Multiple Hosts	49-84
Changing the Quiet Period	49-86
Changing the Switch-to-Client Retransmission Time	49-87
Setting the Switch-to-Client Frame-Retransmission Number	49-88
Configuring an Authenticator and a Supplicant Switch with NEAT	49-89
Manually Reauthenticating a Client Connected to a Port	49-96
Initializing the 802.1X Authentication State	49-96
Removing 802.1X Client Information	49-96
Resetting the 802.1X Configuration to the Default Values	49-96
Controlling Switch Access with RADIUS	49-97
Understanding RADIUS	49-97
RADIUS Operation	49-98
RADIUS Change of Authorization	49-99
Configuring RADIUS	49-104
Displaying the RADIUS Configuration	49-117
Configuring Device Sensor	49-117
About Device Sensor	49-118
MSP-IOS Sensor Device Classifier Interaction	49-119

Configuring Device Sensor	49-119
Configuration Examples for the Device Sensor Feature	49-125
Displaying 802.1X Statistics and Status	49-126
Displaying Authentication Details	49-126
Determining the Authentication Methods Registered with the Auth Manager	49-126
Displaying the Auth Manager Summary for an Interface	49-127
Displaying the Summary of All Auth Manager Sessions on the Switch	49-127
Displaying a Summary of All Auth Manager Sessions on the Switch Authorized for a Specified Authentication Method	49-127
Verifying the Auth Manager Session for an Interface	49-127
Displaying MAB Details	49-129
EPM Logging	49-130
Cisco IOS Security Features	49-131
Prerequisites for X.509v3 Certificates for SSH Authentication	50-1
Restrictions for X.509v3 Certificates for SSH Authentication	50-2
Information About X.509v3 Certificates for SSH Authentication	50-2
X.509v3 Certificates for SSH Authentication Overview	50-2
Server and User Authentication Using X.509v3	50-2
OCSP Response Stapling	50-3
How to Configure X.509v3 Certificates for SSH Authentication	50-3
Configuring Digital Certificates for Server Authentication	50-3
Configuring Digital Certificates for User Authentication	50-4
Configuration Examples for X.509v3 Certificates for SSH Authentication	50-5
Example: Configuring Digital Certificates for Server Authentication	50-5
Example: Configuring Digital Certificate for User Authentication	50-5
Verifying Server and User Authentication Using Digital Certificates	50-6
Additional References for X.509v3 Certificates for SSH Authentication	50-9
Related Documents	50-9
Standards & MIBs	50-10
RFCs	50-10
Technical Assistance	50-10
Feature Information for X.509v3 Certificates for SSH Authentication	50-11
Prerequisites for SSH File Transfer Protocol	51-1
Restrictions for SSH File Transfer Protocol	51-1
Information About SSH File Transfer Protocol	51-2
How to Configure SSH File Transfer Protocol	51-2
Configuring SFTP	51-2
Perform an SFTP Copy Operation	51-2

Example: Configuring SSH File Transfer Protocol	51-3
Related Documents	52-2
RFCs	52-2
About PPPoE Intermediate Agent	52-2
Enabling PPPoE IA on a Switch	52-2
Configuring the Access Node Identifier for PPPoE IA on a Switch	52-2
Configuring the Identifier String, Option, and Delimiter for PPPoE IA on an Switch	52-3
Configuring the Generic Error Message for PPPoE IA on an Switch	52-3
Enabling PPPoE IA on an Interface	52-4
Configuring the PPPoE IA Trust Setting on an Interface	52-4
Configuring PPPoE IA Rate Limiting Setting on an Interface	52-4
Configuring PPPoE IA Vendor-tag Stripping on an Interface	52-5
Configuring PPPoE IA Circuit-ID and Remote-ID on an Interface	52-5
Enabling PPPoE IA for a Specific VLAN on an Interface	52-5
Configuring PPPoE IA Circuit-ID and Remote-ID for a VLAN on an Interface	52-6
Displaying Configuration Parameters	52-6
Clearing Packet Counters	52-8
Debugging PPPoE Intermediate Agent	52-8
Troubleshooting Tips	52-9
About Web-Based Authentication	53-1
Device Roles	53-2
Host Detection	53-2
Session Creation	53-3
Authentication Process	53-3
Customization of the Authentication Proxy Web Pages	53-4
Web-Based Authentication Interactions with Other Features	53-4
Configuring Web-Based Authentication	53-6
Default Web-Based Authentication Configuration	53-6
Web-Based Authentication Configuration Guidelines and Restrictions	53-6
Web-Based Authentication Configuration Task List	53-7
Configuring the Authentication Rule and Interfaces	53-7
Configuring AAA Authentication	53-9
Configuring Switch-to-RADIUS-Server Communication	53-9
Configuring the HTTP Server	53-11
Configuring the Web-Based Authentication Parameters	53-13
Removing Web-Based Authentication Cache Entries	53-14
Displaying Web-Based Authentication Status	53-14
Information About Auto Identity	54-1
Auto Identity Overview	54-2

Auto Identity Global Template	54-2
Auto Identity Interface Templates	54-3
Auto Identity Built-in Policies	54-4
Auto Identity Class Map Templates	54-4
Auto Identity Parameter Maps	54-5
Auto Identity Service Templates	54-5
How to Configure Auto Identity	54-5
Configuring Auto Identity Globally	54-5
Configuring Auto Identity at an Interface Level	54-6
Configuration Examples for Auto Identity	54-6
Example: Configuring Auto Identity Globally	54-7
Example: Configuring Auto Identity at an Interface Level	54-7
Verifying Auto Identity	54-7
Port Security Commands	55-1
About Port Security	55-3
Secure MAC Addresses	55-3
Maximum Number of Secure MAC Addresses	55-4
Aging Secure MAC Addresses	55-5
Sticky Addresses on a Port	55-5
Forbidden MAC Addresses	55-6
Violation Actions	55-6
Invalid Packet Handling	55-6
Configuring Port Security on Access Ports	55-7
Configuring Port Security on Access Ports	55-7
Examples of Port Security on Access Ports	55-10
Configuring Port Security on PVLAN Ports	55-14
Configuring Port Security on an Isolated Private VLAN Host Port	55-14
Example of Port Security on an Isolated Private VLAN Host Port	55-16
Configuring Port Security on a Private VLAN Promiscuous Port	55-16
Example of Port Security on a Private VLAN Promiscuous Port	55-17
Configuring Port Security on Trunk Ports	55-17
Configuring Trunk Port Security	55-17
Examples of Trunk Port Security	55-19
Trunk Port Security Configuration Guidelines and Restrictions	55-21
Configuring Port Security on Voice Ports	55-22
Configuring Port Security on Voice Ports	55-23
Examples of Voice Port Security	55-25
Voice Port Security Configuration Guidelines and Restrictions	55-27
Displaying Port Security Settings	55-27

Examples of Security Settings	55-28
Configuring Port Security with Other Features/Environments	55-31
DHCP and IP Source Guard	55-31
802.1X Authentication	55-32
Configuring Port Security in a Wireless Environment	55-32
Port Security Configuration Guidelines and Restrictions	55-33
About Auto Security	56-1
Feature Interaction	56-1
Configuring Auto Security	56-2
Enabling auto security globally	56-2
Disabling auto security globally	56-2
Enabling Auto Security Feature for Access (End Hosts) or Trunk (Uplink) Ports	56-3
Guidelines and Restrictions	56-6
Configuring Control Plane Policing	57-2
About Control Plane Policing	57-2
General Guidelines for Control Plane Policing	57-3
Default Configuration	57-4
Configuring CoPP for Control Plane Traffic	57-4
Configuring CoPP for Data Plane and Management Plane Traffic	57-6
Control Plane Policing Configuration Guidelines and Restrictions	57-8
Monitoring CoPP	57-9
Configuring Layer 2 Control Packet QoS	57-11
Understanding Layer 2 Control Packet QoS	57-11
Default Configuration	57-12
Enabling Layer 2 Control Packet QoS	57-12
Disabling Layer 2 Control Packet QoS	57-13
Layer 2 Control Packet QoS Configuration Examples	57-14
Layer 2 Control Packet QoS Guidelines and Restrictions	57-16
Policing IPv6 Control Traffic	57-16
About Dynamic ARP Inspection	58-1
ARP Cache Poisoning	58-2
Purpose of Dynamic ARP Inspection	58-2
Interface Trust State, Security Coverage and Network Configuration	58-3
Relative Priority of Static Bindings and DHCP Snooping Entries	58-4
Logging of Dropped Packets	58-4
Rate Limiting of ARP Packets	58-4
Port Channels Function	58-5
Configuring Dynamic ARP Inspection	58-5

Configuring Dynamic ARP Inspection in DHCP Environments	58-5
DAI Configuration Example	58-7
Configuring ARP ACLs for Non-DHCP Environments	58-11
Configuring the Log Buffer	58-14
Limiting the Rate of Incoming ARP Packets	58-16
Performing Validation Checks	58-19
Finding Feature Information	59-1
Prerequisites for Configuring the DHCP Server	59-1
Information About Cisco IOS DHCP Server	59-2
Overview of the DHCP Server	59-2
DHCP Attribute Inheritance	59-2
DHCP Server Address Allocation Using Option 82	59-3
Disabling Conflict Logging	59-4
DHCP Address Pools	59-5
Manual Bindings	59-6
DHCP Static Mapping	59-7
DHCP Server Operation	59-8
Static Route with the Next-Hop Dynamically Obtained Through DHCP	59-9
How to Configure the Cisco IOS DHCP Server	59-9
Configuring a DHCP Database Agent or Disabling Conflict Logging	59-10
Excluding IP Addresses	59-10
Configuring DHCP Address Pools	59-10
Configuring Manual Bindings	59-17
Configuring the DHCP Server to Read a Static Mapping Text File	59-18
Customizing DHCP Server Operation	59-19
Configuring a Remote Device to Import DHCP Server Options from a Central DHCP Server	59-19
Configuring DHCP Address Allocation Using Option 82	59-21
Configuring Static Route with the Next-Hop Dynamically Obtained Through DHCP	59-23
Clearing DHCP Server Variables	59-24
Configuration Examples for the Cisco IOS DHCP Server	59-24
Example: Configuring a DHCP Database Agent or Disabling Conflict Logging	59-24
Example: Excluding IP Addresses	59-24
Example: Configuring a DHCP Address Pool	59-25
Example: Configuring Manual Bindings	59-27
Example: Configuring Static Mapping	59-28
Example: Customizing DHCP Server Operation	59-29
Example: Configuring a Remote Device to Import DHCP Server Options from a Central DHCP Server	59-30
Example: Configuring DHCP Address Allocation Using Option 82	59-31

Example: Configuring a Static Route with the Next-Hop Dynamically Obtained Through DHCP	59-32
Additional References for the Cisco IOS DHCP Server	59-33
Related Documents	59-33
Standards & RFCs	59-33
MIBs	59-33
Technical Assistance	59-33
Feature Information for the IOS DHCP Server	59-34
About DHCP Snooping	60-1
Trusted and Untrusted Sources	60-2
About the DHCP Snooping Database Agent	60-2
Option 82 Data Insertion	60-3
Configuring DHCP Snooping	60-6
Default Configuration for DHCP Snooping	60-7
Enabling DHCP Snooping	60-7
Enabling DHCP Snooping on the Aggregation Switch	60-9
Enabling DHCP Snooping and Option 82	60-10
Enabling DHCP Snooping on Private VLAN	60-12
Configuring DHCP Snooping on Private VLAN	60-12
Configuring DHCP Snooping with an Ethernet Channel Group	60-12
Enabling the DHCP Snooping Database Agent	60-13
Limiting the Rate of Incoming DHCP Packets	60-13
Configuration Examples for the Database Agent	60-15
Displaying DHCP Snooping Information	60-18
Displaying a Binding Table	60-19
Displaying the DHCP Snooping Configuration	60-19
About IP Source Guard	60-19
Configuring IP Source Guard	60-20
Configuring IP Source Guard on Private VLANs	60-22
Displaying IP Source Guard Information	60-22
Displaying IP Source Binding Information	60-23
Configuring IP Source Guard for Static Hosts	60-24
About IP Source Guard for Static Hosts	60-24
Configuring IPSG for Static Hosts on a Layer 2 Access Port	60-24
Configuring IPSG for Static Hosts on a PVLAN Host Port	60-28
Restrictions for DHCPv6 Options Support	61-1
Information About DHCPv6 Options Support	61-2
DHCPv6 Relay Agent Overview	61-2
DHCPv6 Relay Options: Remote-ID	61-2

DHCPv6 Interface-ID	61-3
Lightweight DHCPv6 Relay Agent	61-3
Interoperability between DHCPv6 Relay Agents and LDRA	61-3
LDRA for VLANs and Interfaces	61-4
CAPWAP Access Controller DHCPv6 Option	61-4
How to Configure DHCPv6 Options Support	61-5
Configuring the DHCPv6 Relay Agent	61-5
Configuring LDRA Functionality on a VLAN	61-5
Configuring LDRA Functionality on an Interface	61-6
Verifying the LDRA Configuration	61-7
Configuring CAPWAP Access Points	61-8
Configuration Examples for DHCPv6 Options Support	61-9
Example: Configuring the DHCPv6 Relay Agent	61-9
Example: Configuring LDRA Functionality on a VLAN	61-9
Example: Configuring LDRA Functionality on an Interface	61-9
Example: Configuring CAPWAP Access Points	61-10
Additional References for DHCPv6 Options Support	61-10
Related Documents	61-10
Standards and RFCs	61-10
MIBs	61-10
Technical Assistance	61-11
Feature Information for DHCPv6 Options Support	61-12
About ACLs	62-2
Overview	62-2
Supported Features That Use ACLs	62-3
Router ACLs	62-3
Port ACLs	62-4
Dynamic ACLs	62-5
VLAN Maps	62-5
Hardware and Software ACL Support	62-6
Troubleshooting High CPU Due to ACLs	62-7
Selecting Mode of Capturing Control Packets	62-7
TCAM Programming and ACLs	62-10
Layer 4 Operators in ACLs	62-11
Restrictions for Layer 4 Operations	62-11
Configuration Guidelines for Layer 4 Operations	62-12
Using ACLs to Filter TCP Flags and How ACL Processing Impacts CPU	62-13
Configuring Unicast MAC Address Filtering	62-16

Configuring Named MAC Extended ACLs	62-16
Configuring EtherType Matching	62-17
Configuring Named IPv6 ACLs	62-18
Applying IPv6 ACLs to Layer 2 and 3 Interface	62-20
Configuring VLAN Maps	62-21
VLAN Map Configuration Guidelines	62-22
Creating and Deleting VLAN Maps	62-22
Applying a VLAN Map to a VLAN	62-25
Using VLAN Maps in Your Network	62-25
Displaying VLAN Access Map Information	62-28
Using VLAN Maps with Router ACLs	62-28
Guidelines for Using Router ACLs and VLAN Maps on the Same VLAN	62-29
Examples of Router ACLs and VLAN Maps Applied to VLANs	62-29
Configuring PACLS	62-31
Creating a PACL	62-31
PACL Configuration Guidelines	62-32
Removing the Requirement for a Port ACL	62-32
Webauth Fallback	62-33
Configuring IPv4, IPv6, and MAC ACLs on a Layer 2 Interface	62-33
Using PACL with Access-Group Mode	62-34
Configuring Access-group Mode on Layer 2 Interface	62-35
Applying ACLs to a Layer 2 Interface	62-35
Displaying an ACL Configuration on a Layer 2 Interface	62-36
Using PACL with VLAN Maps and Router ACLs	62-36
Configuring Object Group ACLs	62-39
Overview	62-39
Configuring IPv4 OG ACLs	62-40
Configuring IPv6 OG ACLs	62-46
Configuring RA Guard	62-50
Introduction	62-50
Deployment	62-51
Configuring RA Guard	62-51
Examples	62-52
Usage Guidelines	62-53
Prerequisites for Authorization and Revocation of Certificates	63-1
Restrictions for Authorization and Revocation of Certificates	63-2
Information About Authorization and Revocation of Certificates	63-2
PKI Authorization	63-2

PKI and AAA Server Integration for Certificate Status	63-2
RADIUS or TACACS+ Choosing a AAA Server Protocol	63-3
Attribute-Value Pairs for PKI and AAA Server Integration	63-3
CRLs or OCSP Server Choosing a Certificate Revocation Mechanism	63-5
What Is a CRL	63-6
What Is OCSP	63-7
When to Use Certificate-Based ACLs for Authorization or Revocation	63-8
PKI Certificate Chain Validation	63-9
High-Availability Support	63-10
How to Configure Authorization and Revocation of Certificates for Your PKI	63-11
Configuring PKI Integration with a AAA Server	63-11
Configuring a Revocation Mechanism for PKI Certificate Status Checking	63-15
The revocation-check Command	63-15
Nonces and Peer Communications with OCSP Servers	63-16
Configuring a Revocation Mechanism for PKI Certificate Status Checking	63-16
Configuring Certificate Authorization and Revocation Settings	63-18
Configuring Certificate-Based ACLs to Ignore Revocation Checks	63-18
Manually Overriding CDPs in a Certificate	63-18
Manually Overriding the OCSP Server Setting in a Certificate	63-18
Configuring CRL Cache Control	63-19
Configuring Certificate Serial Number Session Control	63-19
Configuring Certificate Chain Validation	63-25
Configuring Certificate Servers for High Availability	63-26
Configuring SCTP on the Active and Standby Certificate Servers	63-28
Synchronizing the Active and Standby Certificate Servers	63-30
Configuration Examples for Setting Up Authorization and Revocation of Certificates	63-31
Configuring and Verifying PKI AAA Authorization Examples	63-31
Debug of a Successful PKI AAA Authorization Example	63-33
Debugs of a Failed PKI AAA Authorization Example	63-34
Configuring a Revocation Mechanism Examples	63-35
Configuring an OCSP Server Example	63-36
Specifying a CRL and Then an OCSP Server Example	63-36
Specifying an OCSP Server Example	63-36
Disabling Nonces in Communications with the OCSP Server Example	63-36
Configuring a Hub Router at a Central Site for Certificate Revocation Checks Example	63-36
Configuring Certificate Authorization and Revocation Settings Examples	63-40
Configuring Certificate Authorization and Revocation Settings Examples	63-40
Configuring CRL Cache Control	63-40

Configuring Certificate Serial Number Session Control	63-41
Configuring Certificate Chain Validation Examples	63-42
Configuring Certificate Chain Validation from Peer to Root CA	63-43
Configuring Certificate Chain Validation from Peer to Subordinate CA	63-43
Configuring Certificate Chain Validation Through a Gap	63-43
Additional References	63-44
Finding Feature Information	64-1
About IPv6	64-1
DHCP	64-2
Security	64-2
First-Hop Security	64-3
QoS	64-3
Management	64-3
Multicast	64-4
Static Routes	64-4
First-Hop Redundancy Protocols	64-5
Unicast Routing	64-5
Tunneling	64-7
IPv6 Default States	64-7
About IPv6 Addressing and Basic Connectivity	65-1
IPv6 for Cisco Software	65-2
Large IPv6 Address Space for Unique Addresses	65-2
IPv6 Address Formats	65-3
IPv6 Address Output Display	65-4
Simplified IPv6 Packet Header	65-4
Path MTU Discovery for IPv6	65-8
IPv6 Prefix Aggregation	65-8
IPv6 Site Multihoming	65-9
IPv6 Data Links	65-9
Dual IPv4 and IPv6 Protocol Stacks	65-9
Cisco Discovery Protocol IPv6 Address Support	65-10
ICMP for IPv6	65-11
IPv6 Neighbor Discovery	65-11
IPv6 Neighbor Solicitation Message	65-11
Configuring IPv6 Addressing and Basic Connectivity	65-13
Guidelines for Implementing IPv6 Addressing and Basic Connectivity	65-13
Configuring IPv6 Addressing and Enabling IPv6 Routing	65-14
Mapping Hostnames to IPv6 Addresses	65-15
Displaying IPv6 Redirect Messages	65-16

Configuration Examples for IPv6 Addressing and Basic Connectivity	65-16
About SISF-Based Device Tracking	66-1
Guidelines for Enabling SISF-Based Device Tracking	66-2
Manually Enabling the Device Tracking	66-2
Programmatically Enabling Device Tracking	66-3
Migrating from Legacy Commands to SISF-Based Device Tracking	66-3
No IPDT and No IPv6 Snooping Configuration Exists	66-3
Only IPDT Configuration Exists	66-3
Only IPv6 Snooping Configuration Exists	66-4
Both IPDT and IPv6 Snooping Configurations Exist	66-4
Manually Enabling SISF-Based Device Tracking	66-4
Applying the Default Device Tracking Policy to a Target	66-5
Creating a Custom Device Tracking Policy with Custom Settings	66-6
Attaching a Device Tracking Policy to an Interface	66-9
Attaching a Device Tracking Policy to a VLAN	66-9
Programmatically Enabling SISF-Based Device Tracking	66-10
Configuration Examples for SISF-Based Device Tracking	66-12
Example: Configuring a Multi-Switch Network to Stop Creating Binding Entries from a Trunk Port	66-12
Example: Applying a Merged (IPv4 and IPv6) Device Tracking Policy to the Same Target	66-12
Example: Manually Attaching an IPv4 Device Tracking Policy to a VLAN	66-14
Example: Disabling IPv6 Device Tracking	66-15
Example: Enabling IPv6 for SVI on VLAN (To Mitigate the Duplicate Address Problem)	66-15
Example: Mitigating the IPv4 Duplicate Address Problem	66-15
Example: Avoiding a Short Device-Tracking Binding Reachable Time	66-17
About Flood Blocking	67-1
Configuring Port Blocking	67-1
Blocking Flooded Traffic on an Interface	67-2
Resuming Normal Forwarding on a Port	67-3
About Storm Control	68-1
Hardware-Based Storm Control Implementation	68-1
Software-Based Storm Control Implementation	68-2
Enabling Broadcast Storm Control	68-3
Enabling Multicast Storm Control	68-4
Disabling Broadcast Storm Control	68-5
Disabling Multicast Storm Control	68-6
Displaying Storm Control	68-6
About SPAN and RSPAN	69-1

SPAN and RSPAN Concepts and Terminology	69-3
SPAN and RSPAN Session Limits	69-6
Default SPAN and RSPAN Configuration	69-6
Configuring SPAN	69-7
SPAN Configuration Guidelines and Restrictions	69-7
Configuring SPAN Sources	69-8
Configuring SPAN Destinations	69-9
Monitoring Source VLANs on a Trunk Interface	69-9
Configuration Scenario	69-10
Verifying a SPAN Configuration	69-10
CPU Port Sniffing	69-10
Encapsulation Configuration	69-11
Ingress Packets	69-12
Access List Filtering	69-13
ACL Configuration Guidelines	69-13
Configuring Access List Filtering	69-14
Packet Type Filtering	69-14
Configuration Example	69-15
Configuring RSPAN	69-16
RSPAN Configuration Guidelines	69-16
Creating an RSPAN Session	69-17
Creating an RSPAN Destination Session	69-18
Creating an RSPAN Destination Session and Enabling Ingress Traffic	69-19
Removing Ports from an RSPAN Session	69-20
Specifying VLANs to Monitor	69-21
Specifying VLANs to Filter	69-23
Displaying SPAN and RSPAN Status	69-24
Prerequisites for ERSPAN	70-1
Restrictions for ERSPAN	70-2
Information About ERSPAN	70-2
ERSPAN Overview	70-2
ERSAN Sources	70-4
How to Configure ERSPAN	70-5
Configuring an ERSPAN Source Session	70-5
Configuration Examples for ERSPAN	70-6
Example: Configuring an ERSPAN Source Session	70-6
Verifying ERSPAN	70-6
Additional References for Configuring ERSPAN	70-8

Related Documents	70-8
Standards & MIBs	70-8
RFCs	70-8
Technical Assistance	70-8
Feature Information for ERSPAN	70-9
Finding Feature Information	71-1
Prerequisites for Wireshark	71-2
Guidelines for Wireshark	71-2
Restrictions for Wireshark	71-4
Information about Wireshark	71-5
Capture Points	71-6
Attachment Points	71-6
Filters	71-6
Input and Output Classification	71-7
Actions	71-8
Storing Captured Packets to Buffer in Memory	71-8
Decoding and Displaying Packets	71-9
Activating and Deactivating Wireshark Capture Points	71-10
Wireshark Features used in Switches	71-10
Wireshark on VSS	71-11
How to Configure Wireshark	71-11
Default Wireshark Configuration	71-11
Defining, Modifying, or Deleting a Capture Point	71-11
Activating and Deactivating a Capture Point	71-13
Configuring Wireshark on VSS	71-14
Monitoring Wireshark	71-14
Configuration Examples for Wireshark	71-14
Example: Displaying a Brief Output from a .pcap File	71-14
Example: Displaying Detailed Output from a .pcap File	71-15
Example: Displaying a Hexadecimal Dump Output from a .pcap File	71-16
Example: Displaying Packets from a .pcap File with a Display Filter	71-17
Usage Examples for Wireshark	71-18
Example: Simple Capture and Display	71-18
Example: Simple Capture and Store	71-19
Example: Using Buffer Capture	71-20
Example: Capture Sessions	71-24
Example: Capture and Store in Lock-step Mode	71-28
Example: Simple Capture and Store in Lock-step with High-speed Mode	71-29
Example: Simple Capture and Store of Packets in Egress Direction	71-30

VSS Specific Examples	71-31
Example: Capturing and Storing in a file	71-31
Example: Capturing and Storing in a File with Display	71-31
Example: Circular Buffer Usage	71-32
Understanding Enhanced Object Tracking	72-1
Configuring Enhanced Object Tracking Features	72-2
Default Configuration	72-2
Tracking Interface Line-Protocol or IP Routing State	72-2
Configuring a Tracked List	72-3
Configuring HSRP Object Tracking	72-7
Configuring Other Tracking Characteristics	72-8
Configuring IP SLAs Object Tracking	72-8
Configuring Static Routing Support	72-10
Monitoring Enhanced Object Tracking	72-12
About System Message Logging	73-1
Configuring System Message Logging	73-2
System Log Message Format	73-2
Default System Message Logging Configuration	73-3
Disabling Message Logging	73-3
Setting the Message Display Destination Device	73-4
Synchronizing Log Messages	73-5
Enabling and Disabling Timestamps on Log Messages	73-6
Enabling and Disabling Sequence Numbers in Log Messages (Optional)	73-7
Defining the Message Severity Level (Optional)	73-8
Limiting Syslog Messages Sent to the History Table and to SNMP (Optional)	73-9
Configuring UNIX Syslog Servers	73-10
Displaying the Logging Configuration	73-12
Prerequisites for OBFL	74-1
Restrictions for OBFL	74-2
Information About OBFL	74-2
Overview of OBFL	74-2
Information about Data Collected by OBFL	74-2
Default Settings for OBFL	74-8
Enabling OBFL	74-8
Configuration Examples for OBFL	74-9
Enabling OBFL Message Logging: Example	74-9
OBFL Message Log: Example	74-9
OBFL Component Uptime Report: Example	74-10

OBFL Report for a Specific Time: Example	74-10
About SNMP	75-1
SNMP Versions	75-2
SNMP Manager Functions	75-3
SNMP Agent Functions	75-4
SNMP Community Strings	75-4
Using SNMP to Access MIB Variables	75-4
SNMP Notifications	75-5
Configuring SNMP	75-5
Default SNMP Configuration	75-5
SNMP Configuration Guidelines	75-6
Disabling the SNMP Agent	75-7
Configuring Community Strings	75-7
Configuring SNMP Groups and Users	75-9
Configuring SNMP Notifications	75-11
Setting the Agent Contact and Location Information	75-14
Limiting TFTP Servers Used Through SNMP	75-15
SNMP Examples	75-15
Displaying SNMP Status	75-16
VSS Environment	76-1
Non-VSS Environment	76-8
About Ethernet CFM	77-2
Ethernet CFM and OAM Definitions	77-2
CFM Domain	77-3
Maintenance Associations and Maintenance Points	77-4
CFM Messages	77-5
Crosscheck Function and Static Remote MEPs	77-5
SNMP Traps and Fault Alarms	77-5
Configuration Error List	77-6
IP SLAs Support for CFM	77-6
Configuring Ethernet CFM	77-6
Ethernet CFM Default Configuration	77-7
Ethernet CFM Configuration Guidelines	77-7
Configuring the CFM Domain	77-8
Configuring Ethernet CFM Crosscheck	77-11
Configuring Static Remote MEP	77-13
Configuring a Port MEP	77-14
Configuring SNMP Traps	77-16
Configuring Fault Alarms	77-16

Configuring IP SLAs CFM Operation	77-18
Configuring CFM on C-VLAN (Inner VLAN)	77-24
Understanding CFM ITU-T Y.1731 Fault Management	77-27
Y.1731 Terminology	77-27
Alarm Indication Signals	77-28
Ethernet Remote Defect Indication	77-28
Multicast Ethernet Loopback	77-29
Configuring Y.1731 Fault Management	77-29
Default Y.1731 Configuration	77-29
Configuring ETH-AIS	77-29
Using Multicast Ethernet Loopback	77-31
Managing and Displaying Ethernet CFM Information	77-31
About Ethernet OAM Protocol	77-33
OAM Features	77-34
OAM Messages	77-34
Enabling and Configuring Ethernet OAM	77-35
Ethernet OAM Default Configuration	77-35
Ethernet OAM Configuration Guidelines	77-35
Enabling Ethernet OAM on an Interface	77-36
Enabling Ethernet OAM Remote Loopback	77-37
Configuring Ethernet OAM Link Monitoring	77-38
Configuring Ethernet OAM Remote Failure Indications	77-42
Configuring Ethernet OAM Templates	77-45
Displaying Ethernet OAM Protocol Information	77-49
Ethernet CFM and Ethernet OAM Interaction	77-51
Configuring Ethernet OAM Interaction with CFM	77-51
Example: Configuring Ethernet OAM and CFM	77-53
AIS and RDI Terminology	78-1
About Y.1731	78-2
Server MEP	78-2
Alarm Indication Signal	78-2
Ethernet Remote Defect Indication	78-3
Configuring Y.1731	78-4
Y.1731 Configuration Guidelines	78-4
Configuring AIS Parameters	78-4
Clearing MEP from the AIS Defect Condition	78-5
Clearing SMEP from the AIS Defect Condition	78-5
Displaying Y.1731 Information	78-5

About Call Home	79-1
Obtaining Smart Call Home	79-2
Configuring Call Home	79-3
Configuring Contact Information	79-4
Configuring Destination Profiles	79-5
Subscribing to Alert Groups	79-6
Configuring General E-Mail Options	79-9
Enabling Call Home	79-10
Testing Call Home Communications	79-10
Configuring and Enabling Smart Call Home	79-13
Displaying Call Home Configuration Information	79-13
Call Home Default Settings	79-18
Alert Group Trigger Events and Commands	79-18
Message Contents	79-21
Syslog Alert Notification in Long-Text Format Example	79-25
Syslog Alert Notification in XML Format Example	79-28
Understanding Cisco IOS IP SLAs	80-2
Using Cisco IOS IP SLAs to Measure Network Performance	80-3
IP SLAs Responder and IP SLAs Control Protocol	80-4
Response Time Computation for IP SLAs	80-4
IP SLAs Operation Scheduling	80-5
IP SLAs Operation Threshold Monitoring	80-5
Configuring IP SLAs Operations	80-6
IP SLA Default Configuration	80-6
IP SLA Configuration Guidelines	80-6
Configuring the IP SLAs Responder	80-7
Analyzing IP Service Levels by Using the UDP Jitter Operation	80-8
Analyzing IP Service Levels by Using the ICMP Echo Operation	80-11
Monitoring IP SLAs Operations	80-12
About RMON	81-1
Configuring RMON	81-2
Default RMON Configuration	81-2
Configuring RMON Alarms and Events	81-2
Configuring RMON Collection on an Interface	81-4
Displaying RMON Status	81-5
Configuring Online Diagnostics	82-1
Performing Diagnostics	82-3
Power-On Self-Test Diagnostics	82-10

POST Result Example	82-11
Power-On Self-Test Results	82-12
Troubleshooting the Test Failures	82-18
Understanding WCCP	83-1
Overview	83-2
Hardware Acceleration	83-2
Understanding WCCP Configuration	83-3
WCCP Features	83-3
Restrictions for WCCP	83-5
Configuring WCCP	83-5
Configuring a Service Group Using WCCP	83-5
Using Access Lists for a WCCP Service Group	83-8
Setting a Password for a Switch and Cache Engine	83-8
Verifying and Monitoring WCCP Configuration Settings	83-9
WCCP Configuration Examples	83-9
Example: Performing a General WCCP Configuration	83-10
Example: Running a Web Cache Service	83-10
Example: Running a Reverse Proxy Service	83-10
Example: Running TCP-Promiscuous Service	83-11
Example: Running Redirect Access List	83-12
Example: Using Access Lists	83-12
Example: Setting a Password for a Switch and Content Engines	83-13
Example: Verifying WCCP Settings	83-13
Determining MIB Support for Cisco IOS Releases	84-1
Using Cisco IOS MIB Tools	84-1
Downloading and Compiling MIBs	84-2
Guidelines for Working with MIBs	84-2
Downloading MIBs	84-3
Compiling MIBs	84-4
Enabling SNMP Support	84-4
Prerequisites for Configuring Easy Virtual Network	85-1
Restrictions for EVN	85-1
About Easy Virtual Network	85-2
Virtual Network Tags Provide Path Isolation	85-2
Virtual Network Tags	85-4
vnet Global	85-4
Edge Interfaces and EVN Trunk Interfaces	85-5
Identifying Trunk Interfaces in Display Output	85-6

Single IP Address on Trunk Interfaces	85-6
Relationship Between VRFs Defined and VRFs Running on a Trunk Interface	85-6
VRF Awareness	85-7
Routing Protocols Supported by EVN	85-8
Packet Flow in a Virtual Network	85-8
Command Inheritance on EVN Trunk Interfaces	85-10
Overriding Command Inheritance Virtual Network Interface Mode	85-10
Removing Overrides and Restoring Values Inherited from EVN Trunk	85-12
Determining if No Form of Commands Appear in Configuration Files	85-13
EVN Compatibility with VRF-Lite	85-13
Example: VRF-Lite Subinterface Configuration EVN Trunk Configuration	85-13
SQoS and EVN	85-14
Configuring Easy Virtual Networks	85-14
Enabling a Subset of VRFs over a Trunk Interface	85-15
Configuring EVN Edge Interfaces	85-16
Verifying EVN Configuration	85-16
Changing the Inherited IP Address for Subinterfaces	85-17
Configuration Examples for Configuring EVN	85-18
Example: Overriding Command Inheritance	85-19
Example: Enabling an Attribute to vnet Global Only	85-20
Example: Command Inheritance and Virtual Network Interface Mode Override in a Multicast Environment	85-20
Example: EVN Using IP Multicast	85-21
Troubleshooting EVN Configuration	85-22
Routing Context for EXEC Mode Reduces Repetitive VRF Specification	85-22
traceroute Output Indicates VRF Name and VRF Tag	85-22
Debug Output Filtering Per VRF	85-23
CISCO-VRF-MIB	85-23
Entering the ROM Monitor	86-1
ROM Monitor Commands	86-2
ROM Monitor Command Descriptions	86-3
Configuration Register	86-3
Changing the Configuration Register Manually	86-3
Changing the Configuration Register Using Prompts	86-4
Console Download	86-4
Error Reporting	86-5
Debug Commands	86-5
Exiting the ROM Monitor	86-6
Information About Zero-Touch Provisioning	87-1

Zero-Touch Provisioning Overview	87-1
DHCP Server Configuration for Zero-Touch Provisioning	87-2
Sample Zero-Touch Provisioning Configurations	87-2
Sample DHCP Server Configuration on a Management Port	87-2
Zero-Touch Provisioning Boot Log	87-2



Preface

This preface describes who should read this document, how it is organized, and its conventions. The preface also tells you how to obtain Cisco documents, as well as how to obtain technical assistance.

Audience

This guide is for experienced network administrators who are responsible for configuring and maintaining Catalyst 4500 series switches.

Conventions

This document uses the following typographical conventions:

Convention	Description
boldface font	Commands, command options, and keywords are in boldface .
<i>italic font</i>	Command arguments for which you supply values are in <i>italics</i> .
[]	Command elements in square brackets are optional.
{ x y z }	Alternative keywords in command lines are grouped in braces and separated by vertical bars.
[x y z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A unquoted set of characters. Do not use quotation marks around the string because the string will include the quotation marks.
screen font	System displays are in <code>screen font</code> .
boldface screen font	Information you must enter verbatim is in boldface screen font .
<i>italic screen font</i>	Arguments for which you supply values are in <i>italic screen font</i> .
→	This pointer highlights an important line of text in an example.
^	Represents the key labeled Control—for example, the key combination ^D in a screen display means hold down the Control key while you press the D key.
< >	Nonprinting characters such as passwords are in angle brackets.

Notes use the following conventions:



Note

Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the publication.

Cautions use the following conventions:



Caution

Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Related Documentation

Refer to the following documents for additional information about Catalyst 4500 series switches:

- Catalyst 4500 Series Switch Documentation Home
http://www.cisco.com/en/US/products/hw/switches/ps4324/tsd_products_support_series_home.html

Hardware Documents

Installation guides and notes including specifications and relevant safety information are available at the following URLs:

Cisco Catalyst 4500E Series Switches

- *Catalyst 4500 E-series Switches Installation Guide*
<http://www.cisco.com/en/US/docs/switches/lan/catalyst4500/hardware/catalyst4500e/installation/guide/Eseries.html>
- For information about individual switching modules and supervisors, refer to the *Catalyst 4500 Series Module Installation Guide* at:
http://www.cisco.com/en/US/docs/switches/lan/catalyst4500/hardware/configuration/notes/OL_25315.html
- Installation notes for specific supervisor engines or for accessory hardware are available at:
http://www.cisco.com/en/US/products/hw/switches/ps4324/prod_installation_guides_list.html
- *Regulatory Compliance and Safety Information for the Catalyst 4500 series switches*
http://www.cisco.com/en/US/docs/switches/lan/catalyst4500/hardware/regulatory/compliance/78_13233.html

Cisco Catalyst 4500-X Series Switches

- Catalyst 4500-X hardware installation information is available at:
http://www.cisco.com/en/US/products/ps12332/prod_installation_guides_list.html
- *Regulatory Compliance and Safety Information for the Cisco Catalyst 4500-X Series Switches*

http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst4500x/regulatory_compliance/OL_26538.html

Software Documentation

Software release notes, configuration guides, command references, and system message guides are available at the following URLs:

- Cisco 4500-X release notes are available at:
http://www.cisco.com/en/US/products/ps12332/prod_release_notes_list.html
- Catalyst 4500E release notes are available at:
http://www.cisco.com/en/US/products/hw/switches/ps4324/prod_release_notes_list.html

Software documents for the Catalyst 4500 E-Series and Catalyst 4500-X Series switches are available at the following URLs:

- *Catalyst 4500 Series Software Configuration Guide*
http://www.cisco.com/en/US/products/hw/switches/ps4324/products_installation_and_configuration_guides_list.html
- *Catalyst 4500 Series Software Command Reference*
http://www.cisco.com/en/US/products/hw/switches/ps4324/prod_command_reference_list.html
- *Catalyst 4500 Series Software System Message Guide*
http://www.cisco.com/en/US/products/hw/switches/ps4324/products_system_message_guides_list.html

Cisco IOS Documentation

Platform-independent Cisco IOS documentation may also apply. These documents and tools are available at the following URLs:

Cisco IOS XE 3E

Cisco IOS XE Configuration Guides	http://www.cisco.com/c/en/us/support/ios-nx-os-software/ios-xe-3e/products-installation-and-configuration-guides-list.html
-----------------------------------	---

Cisco IOS 12.4

Cisco IOS Configuration Guides	http://www.cisco.com/en/US/products/ps6350/products_installation_and_configuration_guides_list.html
Cisco IOS Command References	http://www.cisco.com/en/US/products/ps6350/prod_command_reference_list.html
Cisco IOS System Messages	http://www.cisco.com/en/US/products/ps6350/products_system_message_guides_list.html

Tools

Command Lookup

<http://tools.cisco.com/Support/CLILookup/cltSearchAction.do>

Commands in Task Tables

Commands listed in task tables show only the relevant information for completing the task and not all available options for the command. For a complete description of a command, refer to the command reference guide.

Notices

The following notices pertain to this software license.

OpenSSL/Open SSL Project

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>).

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product includes software written by Tim Hudson (tjh@cryptsoft.com).

License Issues

The OpenSSL toolkit stays under a dual license; that is, both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

OpenSSL License:

Copyright © 1998-2007 The OpenSSL Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>)".
4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.
5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project.

6. Redistributions of any form whatsoever must retain the following acknowledgment:

“This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>)”.

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT “AS IS” AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young (ey@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Original SSLeay License:

Copyright © 1995-1998 Eric Young (ey@cryptsoft.com). All rights reserved.

This package is an SSL implementation written by Eric Young (ey@cryptsoft.com).

The implementation was written so as to conform with Netscapes SSL.

This library is free for commercial and non-commercial use as long as the following conditions are adhered to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young’s, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:

“This product includes cryptographic software written by Eric Young (ey@cryptsoft.com)”.

The word ‘cryptographic’ can be left out if the routines from the library being used are not cryptography-related.

4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement: “This product includes software written by Tim Hudson (tjh@cryptsoft.com)”.

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO

EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The license and distribution terms for any publicly available version or derivative of this code cannot be changed, that is, this code cannot be copied and put under another distribution license [including the GNU Public License].

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

<http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html>

Subscribe to the *What's New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS Version 2.0.



Product Overview

This chapter provides an overview of Catalyst 4500 series switches and includes the following major sections:

- [Layer 2 Software Features, page 1-1](#)
- [Layer 3 Software Features, page 1-13](#)
- [Management Features, page 1-23](#)
- [Security Features, page 1-34](#)

Layer 2 Software Features

The following subsections describe the key Layer 2 switching software features on the switch:

- [802.1Q Tunneling, VLAN Mapping, and Layer 2 Protocol Tunneling, page 1-2](#)
- [Cisco IOS Auto Smartport Macros, page 1-2](#)
- [Cisco Discovery Protocol, page 1-3](#)
- [Cisco Group Management Protocol \(CGMP\) server, page 1-3](#)
- [EtherChannel Bundles, page 1-3](#)
- [Ethernet CFM, page 1-3](#)
- [Ethernet OAM Protocol, page 1-3](#)
- [Flex Links and MAC Address-Table Move Update, page 1-4](#)
- [Flexible NetFlow \(Supervisor Engine 9-E, 8-E, 8L-E, 7-E, and 7L-E only\), page 1-4](#)
- [Internet Group Management Protocol \(IGMP\) Snooping, page 1-4](#)
- [IPv6 Multicast BSR and BSR Scoped Zone Support, page 1-5](#)
- [IPv6 Multicast Listen Discovery \(MLD\) and Multicast Listen Discovery Snooping, page 1-6](#)
- [Jumbo Frames, page 1-6](#)
- [Link Aggregation Control Protocol, page 1-7](#)
- [Link Layer Discovery Protocol, page 1-7](#)
- [Link State Tracking, page 1-8](#)
- [Location Service, page 1-8](#)
- [Multiple Spanning Tree, page 1-8](#)

- [Per-VLAN Rapid Spanning Tree, page 1-8](#)
- [Quality of Service, page 1-9](#)
- [Resilient Ethernet Protocol, page 1-10](#)
- [SmartPort Macros, page 1-10](#)
- [Spanning Tree Protocol, page 1-10](#)
- [Stateful Switchover, page 1-10](#)
- [SVI Autostate, page 1-11](#)
- [Unidirectional Link Detection, page 1-11](#)
- [VLANs, page 1-11](#)
- [Virtual Switching Systems \(Catalyst 4500-X and Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, and 7-E\), page 1-12](#)
- [Virtual Switch System Client, page 1-12](#)
- [Y.1731 \(AIS and RDI\), page 1-13](#)

802.1Q Tunneling, VLAN Mapping, and Layer 2 Protocol Tunneling

802.1Q tunneling is a Q-in-Q technique that expands the VLAN space by retagging the tagged packets that enter the service provider infrastructure. 802.1Q tunneling allows service providers to assign a VLAN to each customer without losing the original customer VLAN IDs inside the tunnel. All data traffic that enters the tunnel is encapsulated with the tunnel VLAN ID. Layer 2 Protocol Tunneling is a similar technique for all Layer 2 control traffic.

To map customer VLANs to service-provider VLANs, you can configure VLAN mapping (or VLAN ID translation) on trunk ports connected to a customer network. Packets entering the port are mapped to a service provider VLAN (S-VLAN) based on the port number and the original customer VLAN-ID (C-VLAN) of the packet.

For information on configuring 802.1Q tunneling and VLAN Mapping, see [Chapter 31, “Configuring 802.1Q Tunneling, VLAN Mapping, and Layer 2 Protocol Tunneling.”](#)

Cisco IOS Auto Smartport Macros

Cisco IOS Auto SmartPort macros dynamically configure ports based on the device type detected on the port. When the switch detects a new device on a port it applies the appropriate Cisco IOS Auto Smartports macro. When a link-down event occurs on the port, the switch removes the macro. For example, when you connect a Cisco IP phone to a port, Cisco IOS Auto SmartPorts automatically applies the IP phone macro. The IP phone macro enables quality of service (QoS), security features, and a dedicated voice VLAN to ensure proper treatment of delay-sensitive voice traffic.

For information on configuring SmartPort macros, see [Chapter 22, “Configuring Cisco IOS Auto Smartport Macros.”](#)

Cisco Discovery Protocol

The Cisco Discovery Protocol (CDP) is a device-discovery protocol that is both media- and protocol-independent. CDP is available on all Cisco products, including routers, switches, bridges, and access servers. Using CDP, a device can advertise its existence to other devices and receive information about other devices on the same LAN. CDP enables Cisco switches and routers to exchange information, such as their MAC addresses, IP addresses, and outgoing interfaces. CDP runs over the data-link layer only, allowing two systems that support different network-layer protocols to learn about each other. Each device configured for CDP sends periodic messages to a multicast address. Each device advertises at least one address at which it can receive Simple Network Management Protocol (SNMP) messages.

For information on configuring CDP, see [Chapter 32, “Configuring Cisco Discovery Protocol.”](#)

Cisco Group Management Protocol (CGMP) server

CGMP server manages multicast traffic. Multicast traffic is forwarded only to ports with attached hosts that request the multicast traffic.

EtherChannel Bundles

EtherChannel port bundles allow you to create high-bandwidth connections between two switches by grouping multiple ports into a single logical transmission path.

For information on configuring EtherChannel, see [Chapter 27, “Configuring EtherChannel and Link State Tracking.”](#)

Ethernet CFM

Ethernet CFM is an end-to-end per-service-instance (per-VLAN) Ethernet layer OAM protocol that includes proactive connectivity monitoring, fault verification, and fault isolation. End-to-end can be provider-edge-to-provider-edge (PE-to-PE) device or customer-edge-to-customer-edge (CE-to-CE) device. Ethernet CFM, as specified by IEEE 802.1ag, is the standard for Layer 2 ping, Layer 2 traceroute, and end-to-end connectivity check of the Ethernet network.

For information about CFM, see [Chapter 77, “Configuring Ethernet OAM and CFM.”](#)

Ethernet OAM Protocol

Ethernet Operations, Administration, and Maintenance (OAM) is a protocol for installing, monitoring, and troubleshooting Ethernet networks to increase management capability within the context of the overall Ethernet infrastructure. You can implement Ethernet OAM on any full-duplex, point-to-point, or emulated point-to-point Ethernet link for a network or part of a network (specified interfaces).

For information about OAM, see [Chapter 77, “Configuring Ethernet OAM and CFM.”](#)

Flex Links and MAC Address-Table Move Update

Flex Links are a pair of Layer 2 interfaces (switch ports or port channels) where one interface is configured to act as a backup to the other. The feature provides an alternative solution to the Spanning Tree Protocol (STP). Flex Links are typically configured in service provider or enterprise networks where customers do not want to run STP on the switch.

MAC Address-Table Move Update allows a switch to provide rapid bidirectional convergence when a primary (forwarding) link goes down and the standby link begins forwarding traffic.

For information about Flex Links and MAC Address-Table Move Update, see [Chapter 24, “Configuring Flex Links and MAC Address-Table Move Update.”](#)

Flexible NetFlow (Supervisor Engine 9-E, 8-E, 8L-E, 7-E, and 7L-E only)

Flow is defined as unique set of key fields attributes, which might include fields of packet, packet routing attributes, and input and output interface information. A NetFlow feature defines a flow as a sequence of packets that have the same values for the feature key fields. Flexible NetFlow (FNF) allows a flow record that specifies various flow attributes to be collected and optionally exported. NetFlow collection supports IP, IPv6 and Layer 2 traffic.

For information on configuring Flexible NetFlow, see [Chapter 76, “Configuring Flexible NetFlow.”](#)

Internet Group Management Protocol (IGMP) Snooping

IGMP snooping manages multicast traffic. The switch software examines IP multicast packets and forwards packets based on their content. Multicast traffic is forwarded only to ports with attached hosts that request multicast traffic.

Support for IGMPv3 provides constrained flooding of multicast traffic in the presence of IGMPv3 hosts or routers. IGMPv3 snooping listens to IGMPv3 query and membership report messages to maintain host-to-multicast group associations. It enables a switch to propagate multicast data only to ports that need it. IGMPv3 snooping is fully interoperable with IGMPv1 and IGMPv2.

Explicit Host Tracking (EHT) is an extension to IGMPv3 snooping. EHT enables immediate leave operations on a per-port basis. EHT can be used to track per host membership information or to gather statistics about all IGMPv3 group members.

The IGMP Snooping Querier is a Layer 2 feature required to support IGMP snooping in a VLAN where PIM and IGMP are not configured because the multicast traffic does not require routing.

With SSO support, Stateful IGMP Snooping propagates the IGMP data learned by the active supervisor engine to the redundant supervisor engine so that when a switchover occurs, the newly active supervisor engine is aware of the multicast group membership, which alleviates a disruption to multicast traffic during a switchover.

Beginning with Release IOS XE 3.5.0E and IOS 15.2(1)E, the Catalyst 4500 series switch supports an application of local IGMP snooping, Multicast VLAN Registration (MVR). MVR is designed for applications using wide-scale deployment of multicast traffic across an Ethernet ring-based service provider network (for example, the broadcast of multiple television channels over a service-provider network). MVR allows a subscriber on a port to subscribe and unsubscribe to a multicast stream on the network-wide multicast VLAN. It allows the single multicast VLAN to be shared in the network while subscribers remain in separate VLANs.

For information on configuring IGMP snooping and MVR, see [Chapter 28, “Configuring IGMP Snooping and Filtering, and MVR.”](#)

IPv6 Multicast BSR and BSR Scoped Zone Support

The bootstrap router (BSR) protocol for PIM-SM provides a dynamic, adaptive mechanism to distribute group-to-RP mapping information rapidly throughout a domain. With the IPv6 BSR feature, if an RP becomes unreachable, it will be detected and the mapping tables will be modified so that the unreachable RP is no longer used, and the new tables will be rapidly distributed throughout the domain.

BSR provides scoped zone support by distributing group-to-RP mappings in networks using administratively scoped multicast. The user can configure candidate BSRs and a set of candidate RPs for each administratively scoped region in the user's domain.

For information on BSR and BSR Scoped Zone Support, see this URL with the following caveats related to support on a Catalyst 4500 series switch:

http://www.cisco.com/en/US/docs/ios-xml/ios/ipmulti_pim/configuration/xr-3s/ip6-mcast-bsr.html

- In the section “IPv6 BSR: Scoped Zone Support,” a paragraph starts as follows:

Unless the C-RP is configured with a scope, it discovers the existence of the administratively scoped zone and its group range through reception of a BSM from the scope zone's elected BSR containing the scope zone's group range.

A C-RP can no longer be configured with a scope. So, the sentence should read:

A C-RP discovers the existence of the administratively scoped zone and its group range through reception of a BSM from the scope zone's elected BSR containing the scope zone's group range.

- In the section “Configuring a BSR and Verifying BSR Information” in Step 3 under Summary Steps and Detailed Steps, the command for configuring a C-BSR is listed as:

ipv6 pim [vrf vrf-name] bsr candidate bsr ipv6-address [hash-mask-length] [priority priority-value]

Because the original syntax mistakenly excludes **scope scope-value** and the “new” option (**accept-rp-candidate access-list-name**) is supported with this release.

ipv6 pim [vrf vrf-name] bsr candidate bsr ipv6-address [hash-mask-length] [priority priority-value] [scope scope-value] [accept-rp-candidate access-list-name]

- In the section “Sending PIM RP Advertisements to the BSR” in Step 3 under Summary and Detailed Steps, the keyword **scope scope-value** should be removed. The **scope** keyword no longer exists for C-RPs.

- In the section “Configuring BSR for Use Within Scoped Zones,” several changes apply.

The following paragraph:

If scope is specified on the candidate RP, then this device will advertise itself as C-RP only to the BSR for the specified scope. If the group list is specified along with the scope, then only prefixes in the access list with the same scope as that configured will be advertised.

Should read:

The candidate RP will advertise the different ranges it serves to the respective elected BSRs. If a group list is specified, for each of the prefixes in the group list, it will verify that there is an elected scoped BSR for the scope of the prefix. If none exists, the prefix will be announced to the elected non-scoped BSR, provided one is present.

Note: If a prefix is not scope specific (for example, FF00::/8), it will only be announced to a non-scoped BSR. If the candidate RP is not configured with a group list, it will behave as if a group list with only the prefix FF00::/8 is configured.

Under the Summary Steps, steps 3 and 4 should read as follows:

```
ipv6 pim [vrf vrf-name] bsr candidate bsr ipv6-address [hash-mask-length] [priority  
priority-value] [scope scope-value] [accept-rp-candidate access-list-name]
```

```
ipv6 pim [vrf vrf-name] bsr candidate rp ipv6-address [group-list access-list-name] [priority  
priority-value] [interval seconds] [bidir]
```

Under the Details Steps, Step 3 should read:

```
ipv6 pim [vrf vrf-name] bsr candidate bsr ipv6-address [hash-mask-length] [priority  
priority-value] [scope scope-value] [accept-rp-candidate access-list-name]
```

Example:

```
Device(config)# ipv6 pim bsr candidate bsr 2001:DB8:1:1:4 scope 6
```

Under the Details Steps, Step 4 should read:

```
ipv6 pim [vrf vrf-name] bsr candidate rp ipv6-address [group-list access-list-name] [priority  
priority-value] [interval seconds] [bidir]
```

Example:

```
Device(config)# ipv6 pim bsr candidate rp 2001:DB8:1:1:1 group-list list
```

- In the section "Configuring BSR Devices to Announce Scope-to-RP Mappings," the keyword **scope scope-value** should be removed from Step 3, both under Summary and Detail Steps.
- In the section "Additional References section," it would be helpful to reference RFC 5059.

IPv6 Multicast Listen Discovery (MLD) and Multicast Listen Discovery Snooping

MLD is a protocol used by IPv6 multicast devices to discover the presence of multicast listeners (nodes that want to receive IPv6 multicast packets) on its directly attached links and to discover which multicast packets are of interest to neighboring nodes. MLD snooping is supported in two different versions: MLD v1 and MLD v2. Network switches use MLD snooping to limit the flood of multicast traffic, causing IPv6 multicast data to be selectively forwarded to a list of ports that want to receive the data, instead of being flooded to all ports in a VLAN. This lessens the load on devices in the network, minimizing unnecessary bandwidth on links, enabling efficient distribution of IPv6 multicast data.

For information on configuring multicast services, see [Chapter 39, "Configuring IP Multicast."](#)

Jumbo Frames

The jumbo frames feature allows the switch to forward packets as large as 9216 bytes (larger than the IEEE Ethernet MTU), rather than declare those frames "oversize" and discard them. This feature is typically used for large data transfers. The jumbo frames feature can be configured on a per-port basis on Layer 2 and Layer 3 interfaces. The feature is supported only on the following hardware:

- WS-X4306-GB: all ports
- WS-X4232-GB-RJ: ports 1-2

- WS-X4418-GB: ports 1-2
- WS-X4412-2GB-TX: ports 13-14
- WS-4648-RJ45V-E
- WS-X4648+RJ45V+E
- WS-C4500X-16
- WS-C4500X-32
- WS-X4706-10GE linecards
- supervisor engine uplink ports

For information on Jumbo Frames, see [Chapter 9, “Configuring Interfaces.”](#)

Link Aggregation Control Protocol

LACP supports the automatic creation of EtherChannels by exchanging LACP packets between LAN ports. LACP packets are exchanged only between ports in passive and active modes. The protocol "learns" the capabilities of LAN port groups dynamically and informs the other LAN ports. After LACP identifies correctly matched Ethernet links, it facilitates grouping the links into an EtherChannel. Then the EtherChannel is added to the spanning tree as a single bridge port.

Cisco IOS XE IP Application Services Features in Cisco IOS XE 3.1.0SG

This section lists the IP Application Services software features that are supported in Cisco IOS XE 3.1.0SG. Links to the feature documentation are included.

Feature guides may contain information about more than one feature. To find information about a specific feature within a feature guide, see the Feature Information table at the end of the guide.

Feature guides document features that are supported on many different software releases and platforms. Your Cisco software release or platform may not support all the features documented in a feature guide. See the Feature Information table at the end of the feature guide for information about which features in that guide are supported in your software release. Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

IEEE 802.3ad Link Aggregation (LACP)

http://www.cisco.com/en/US/docs/ios/cether/configuration/guide/ce_lnkbnld.html

Link Aggregation Control Protocol (LACP) (802.3ad) for Gigabit Interfaces

http://www.cisco.com/en/US/docs/ios/ios_xe/cether/configuration/guide/ce_lnkbnld_xe.html

Link Layer Discovery Protocol

To support non-Cisco devices and to allow for interoperability between other devices, the switch supports the IEEE 802.1AB LLDP. Link Layer Discovery Protocol (LLDP) is a neighbor discovery protocol that is used for network devices to advertise information about themselves to other devices on the network. This protocol runs over the data-link layer, which allows two systems running different network layer protocols to learn about each other.

LLDP supports a set of attributes that it uses to discover neighbor devices. These attributes contain type, length, and value descriptions and are referred to as *TLVs*. LLDP supported devices can use TLVs to receive and send information to their neighbors. Details such as configuration information, device capabilities, and device identity can be advertised using this protocol.

For information on configuring LLDP, see [Chapter 33, “Configuring LLDP, LLDP-MED, and Location Service.”](#)

Link State Tracking

Link-state tracking, also known as trunk failover, is a feature that binds the link state of multiple interfaces. For example, link-state tracking provides redundancy in the network when used with server NIC adapter teaming. When server network adapters are configured in a primary or secondary relationship known as teaming, if the link is lost on the primary interface, connectivity is transparently changed to the secondary interface.

For information on configuring Link State Tracking, see [Chapter 27, “Configuring EtherChannel and Link State Tracking.”](#)

Location Service

The location service feature allows the switch to provide location and attachment tracking information for its connected devices to a Cisco Mobility Services Engine (MSE). The tracked device can be a wireless endpoint, a wired endpoint, or a wired switch or controller. The switch informs device link up and link down events through encrypted Network Mobility Services Protocol (NMSP) location and attachment notifications to the MSE.

For information on configuring LLDP, see [Chapter 33, “Configuring LLDP, LLDP-MED, and Location Service.”](#)

Multiple Spanning Tree

IEEE 802.1s Multiple Spanning Tree (MST) allows for multiple spanning tree instances within a single 802.1Q or Inter-Switch Link (ISL) VLAN trunk. MST extends the IEEE 802.1w Rapid Spanning Tree (RST) algorithm to multiple spanning trees. This extension provides both rapid convergence and load balancing within a VLAN environment.

MST allows you to build multiple spanning trees over trunks. You can group and associate VLANs to spanning tree instances. Each instance can have a topology independent of other spanning tree instances. This new architecture provides multiple forwarding paths for data traffic and enables load balancing. Network fault tolerance is improved because a failure in one instance (forwarding path) does not affect other instances (forwarding paths).

For information on configuring MST, see [Chapter 23, “Configuring STP and MST.”](#)

Per-VLAN Rapid Spanning Tree

Per-VLAN Rapid Spanning Tree (PVRST+) is the implementation of 802.1w on a per-VLAN basis. It is the same as PVST+ with respect to STP mode and runs RSTP protocol based on 802.1w.

For information on configuring PVRST+, see [Chapter 23, “Configuring STP and MST.”](#)

Quality of Service

**Note**

QoS functionality on Supervisor Engine 6-E and Supervisor Engine 6L-E are equivalent.

The quality of service (QoS) feature prevents congestion by selecting network traffic and prioritizing it according to its relative importance. Implementing QoS in your network makes network performance more predictable and bandwidth use more effective.

The switch supports the following QoS features:

- Classification and marking
- Ingress and egress policing, including per-port per-VLAN policing
- Sharing and shaping

The Catalyst 4500 series switch supports trusted boundary, which uses the Cisco Discovery Protocol (CDP) to detect the presence of a Cisco IP phone (such as the Cisco IP Phone 7910, 7935, 7940, and 7960) on a switch port. If the telephone is not detected, the trusted boundary feature disables the trusted setting on the switch port and prevents misuse of a high-priority queue.

The Catalyst 4500 series switch also supports QoS Automation (Auto QoS), which simplifies the deployment of existing QoS features through automatic configuration.

Cisco Modular QoS Command-Line-Interface

Cisco Modular QoS CLI (MQC) is the framework that implements Cisco IOS software QoS. MQC allows the user to define a traffic class, create a traffic policy (containing the QoS feature to be applied to the traffic class), and attach the traffic policy to an interface. MQC is a cross-Cisco baseline that provides a consistent syntax and behavior of QoS features across multiple product families. Cisco IOS Software Release 12.2(40)SG complies to MQC for configuration of QoS features on the Supervisor Engine 6-E. MQC enables rapid deployment of new features and technology innovations and facilitates the management of network performance with respect to bandwidth, delay, jitter, and packet loss, enhancing the performance of mission-critical business applications. The rich and advanced QoS features are enabled using Cisco MQC.

Two-Rate Three-Color Policing

The Two-Rate Three-Color Policing feature (also termed *Hierarchical QoS*) limits the input or output transmission rate of a class of traffic based on user-defined criteria and marks or colors packets by setting the applicable differentiated services code point (DSCP) values. This feature is often configured on the interfaces at the edge of a network to limit the rate of traffic entering or leaving the network. Using this feature, traffic that conforms to user-defined criteria can be sent through the interfaces, while traffic that exceeds or violates these criteria is sent out with a decreased priority setting or even dropped.

For information on QoS and Auto QoS, see [Chapter 44, “Configuring Quality of Service.”](#)

Resilient Ethernet Protocol

Resilient Ethernet Protocol (REP) is a Cisco proprietary protocol that provides an alternative to Spanning Tree Protocol (STP) to control network loops, handle link failures, and improve convergence time. REP controls a group of ports connected in a segment, ensures that the segment does not create any bridging loops, and responds to link failures within the segment. REP provides a basis for constructing more complex networks and supports VLAN load balancing.

For information on REP, see [Chapter 25, “Configuring Resilient Ethernet Protocol.”](#)

SmartPort Macros

SmartPort macros provide a convenient way to save and share common configurations. You can use SmartPort macros to enable features and settings based on the location of a switch in the network and for mass configuration deployments across the network.

For information on configuring SmartPort macros, see [Chapter 21, “Configuring SmartPort Macros.”](#)

Spanning Tree Protocol

The Spanning Tree Protocol (STP) allows you to create fault-tolerant internetworks that ensure an active, loop-free data path between all nodes in the network. STP uses an algorithm to calculate the best loop-free path throughout a switched network.

For information on configuring STP, see [Chapter 23, “Configuring STP and MST.”](#)

The Catalyst 4500 series switch supports the following STP enhancements:

- Spanning tree PortFast—PortFast allows a port with a directly attached host to transition to the forwarding state directly, bypassing the listening and learning states.
- Spanning tree UplinkFast—UplinkFast provides fast convergence after a spanning-tree topology change and achieves load balancing between redundant links using uplink groups. Uplink groups provide an alternate path in case the currently forwarding link fails. UplinkFast is designed to decrease spanning-tree convergence time for switches that experience a direct link failure.
- Spanning tree BackboneFast—BackboneFast reduces the time needed for the spanning tree to converge after a topology change caused by an indirect link failure. BackboneFast decreases spanning-tree convergence time for any switch that experiences an indirect link failure.
- Spanning tree root guard—Root guard forces a port to become a designated port so that no switch on the other end of the link can become a root switch.

For information on the STP enhancements, see [Chapter 26, “Configuring Optional STP Features.”](#)

Stateful Switchover

Stateful switchover (SSO) enables you to propagate configuration and state information from the active to the redundant supervisor engine so that sub-second interruptions in Layer 2 traffic occur when the active supervisor engine switches over to the redundant supervisor engine.

- Stateful IGMP Snooping

This feature propagates the IGMP data learned by the active supervisor engine to the redundant supervisor engine so that when a switchover occurs, the newly active supervisor engine is aware of the multicast group membership, which alleviates a disruption to multicast traffic during a switchover.

- Stateful DHCP Snooping

This feature propagates the DHCP-snooped data from the active supervisor engine to the redundant supervisor engine so that when a switchover occurs, the newly active supervisor engine is aware of the DHCP data that was already snooped, and the security benefits continue uninterrupted.

For information about SSO, see [Chapter 13, “Configuring Cisco NSF with SSO Supervisor Engine Redundancy.”](#)

SVI Autostate

When an SVI has multiple ports on a VLAN, normally the SVI will go down when all the ports in the VLAN go down. You can design your network so that some ports are not counted in the calculation of SVI “going up or down.” SVI Autostate provides a knob to mark a port so that it is not counted in the SVI “going up and down” calculation and applies to all VLANs that are enabled on that port.

Unidirectional Link Detection

The Unidirectional Link Detection (UDLD) protocol allows devices connected through fiber-optic or copper Ethernet cables to monitor the physical configuration of the cables and detect a unidirectional link.

With standard UDLD, the time to detect a unidirectional link can vary from a few seconds to several minutes depending on how the timers are configured. Link status messages are exchanged every couple of seconds. With Fast UDLD, you can detect unidirectional links in under one second (this also depends on how the timers are configured). Link status messages are exchanged every couple of hundred milliseconds.

For information about UDLD and Fast UDLD, see [Chapter 34, “Configuring UDLD.”](#)

VLANs

A VLAN configures switches and routers according to logical, rather than physical, topologies. Using VLANs, you can combine any collection of LAN segments within an internetwork into an autonomous user group, such that the segments appear as a single LAN in the network. VLANs logically segment the network into different broadcast domains so that packets are switched only between ports within the VLAN. Typically, a VLAN corresponds to a particular subnet, although not necessarily.

For more information about VLANs, VTP, and Dynamic VLAN Membership, see [Chapter 17, “Configuring VLANs, VTP, and VMPS.”](#)

The following VLAN-related features also are supported:

- VLAN Trunking Protocol (VTP)—VTP maintains VLAN naming consistency and connectivity between all devices in the VTP management domain. You can have redundancy in a domain by using multiple VTP servers, through which you can maintain and modify the global VLAN information. Only a few VTP servers are required in a large network.

- **Private VLANs**—Private VLANs are sets of ports that have the features of normal VLANs and also provide some Layer 2 isolation from other ports on the switch.
For information about private VLANs, see [Chapter 47, “Configuring Private VLANs.”](#)
- **Private VLAN Trunk Ports**—Private VLAN trunk ports allow a secondary port on a private VLAN to carry multiple secondary VLANs.
- **Private VLAN Promiscuous Trunk Ports**—Private VLAN promiscuous trunk extends the promiscuous port to a 802.1Q trunk port, carrying multiple primary VLANs (hence multiple subnets). Private VLAN promiscuous trunk is typically used to offer different services or content on different primary VLANs to isolated subscribers. Secondary VLANs can not be carried over the private VLAN promiscuous trunk.
- **Dynamic VLAN Membership**—Dynamic VLAN Membership allows you to assign switch ports to VLANs dynamically, based on the source Media Access Control (MAC) address of the device connected to the port. When you move a host from a port on one switch in the network to a port on another switch in the network, that switch dynamically assigns the new port to the proper VLAN for that host. With the VMPS Client feature, you can convert a dynamic access port to a VMPS client. VMPS clients can use VQP queries to communicate with the VMPS server to obtain a VLAN assignment for the port based on the MAC address of the host attached to that port.

Virtual Switching Systems (Catalyst 4500-X and Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, and 7-E)

Network operators increase network reliability by configuring switches and by provisioning links to the redundant pairs. Redundant network elements and redundant links can add complexity to network design and operation. Virtual switching simplifies the network by reducing the number of network elements and hiding the complexity of managing redundant switches and links.

A VSS combines a pair of Catalyst 4500 or 4500-X series switches into a single network element. The VSS manages the redundant links, which externally act as a single port channel. Starting with Cisco Release IOS XE 3.4.0SG, the Catalyst 4500 or 4500-X series switches support VSS.

**Note**

Smart Install Director is not supported with VSS.

For information on VSS, see [Chapter 5, “Configuring Virtual Switching Systems.”](#)

Virtual Switch System Client

Catalyst 4500 series switches support enhanced PAgP. If a Catalyst 4500 series switch is connected to a Catalyst 6500 series Virtual Switch System (VSS) with a PAgP EtherChannel, the Catalyst 4500 series switch will automatically serve as a VSS client, using enhanced PAgP on this EtherChannel for dual-active detection. This VSS client feature has no impact on the performance of Catalyst 4500 series switches and does not require any user configuration.

For more details, see [Chapter 27, “Configuring EtherChannel and Link State Tracking.”](#)

Y.1731 (AIS and RDI)

Y.1731 ETH-AIS (Ethernet Alarm Indication Signal function) and ETH-RDI (Ethernet Remote Defect Indication function) provides fault and performance management for service providers in large networks.

ETH-AIS suppresses alarms following detection of defect conditions at the server (sub) layer. Due to independent restoration capabilities provided within the Spanning Tree Protocol (STP) environments, ETH-AIS is not expected to be applied in the STP environments. In this case, AIS is configurable, and the administrator describes how to enable and disable AIS in STP environment or not.

ETH-RDI can be used by a MEP to communicate to its peer MEPs that a defect condition has been encountered. ETH-RDI is used only when ETH-CC transmission is enabled.

For information about Y.1731, see [Chapter 78, “Configuring Y.1731 \(AIS and RDI\).”](#)

Layer 3 Software Features

A Layer 3 switch is a high-performance switch that has been optimized for a campus LAN or an intranet, and it provides both wire-speed Ethernet routing and switching services. Layer 3 switching improves network performance with two software functions: route processing and intelligent network services.

Compared to conventional software-based switches, Layer 3 switches process more packets faster by using application-specific integrated circuit (ASIC) hardware instead of microprocessor-based engines.

The following sections describe the key Layer 3 switching software features on the switch:

- [Bidirectional Forwarding Detection, page 1-14](#)
- [Cisco Express Forwarding, page 1-14](#)
- [Device Sensor, page 1-14](#)
- [EIGRP Stub Routing, page 1-14](#)
- [Enhanced Object Tracking, page 1-15](#)
- [GLBP, page 1-15](#)
- [HSRP, page 1-16](#)
- [In Service Software Upgrade, page 1-20](#)
- [IP Routing Protocols, page 1-17](#)
- [IPv6, page 1-20](#)
- [Multicast Services, page 1-20](#)
- [NSF with SSO, page 1-21](#)
- [OSPF for Routed Access, page 1-22](#)
- [Policy-Based Routing, page 1-22](#)
- [Unicast Reverse Path Forwarding, page 1-22](#)
- [Unicast Reverse Path Forwarding, page 1-22](#)
- [Unidirectional Link Routing, page 1-23](#)
- [VRF-lite, page 1-23](#)
- [Virtual Router Redundancy Protocol, page 1-23](#)

Bidirectional Forwarding Detection

**Note**

Starting with Cisco IOS XE 3.5.0E and IOS 15.2(1)E, Bidirectional Forwarding Detection (BFD) support was supported on Supervisor Engine 7-E, and Supervisor Engine 7L-E. With Cisco IOS XE 3.6.0E and IOS 15.2(2)E, support was extended to Supervisor Engine 8-E. Starting with Cisco IOS XE Release 3.10.0E, the feature is supported on Supervisor Engine 9-E.

Bidirectional Forwarding Detection (BFD) protocol. BFD is a detection protocol designed to provide fast forwarding path failure detection times for all media types, encapsulations, topologies, and routing protocols. It includes a description of how to configure multihop BFD sessions. BFD provides a consistent failure detection method for network administrators in addition to fast forwarding path failure detection.

For information on configuring BFD, see [Chapter 41, “Configuring Bidirectional Forwarding Detection.”](#)

Cisco Express Forwarding

Cisco Express Forwarding (CEF) is an advanced Layer 3 IP-switching technology. CEF optimizes network performance and scalability in networks with large and dynamic traffic patterns, such as the Internet, and on networks that use intensive web-based applications or interactive sessions. Although you can use CEF in any part of a network, it is designed for high-performance, highly resilient Layer 3 IP-backbone switching.

For information on configuring CEF, see [Chapter 37, “Configuring Cisco Express Forwarding.”](#)

Device Sensor

Device Sensor uses protocols such as Cisco Discovery Protocol (CDP), Link Layer Discovery Protocol (LLDP), and DHCP to obtain endpoint information from network devices and make this information available to its clients. Device Sensor has internal clients, such as the embedded Device Classifier (local analyzer), Auto Smartports (ASP), MediaNet Service Interface (MSI)-Proxy, and EnergyWise. Device Sensor also has an external client, Identity Services Engine (ISE), which uses RADIUS accounting to receive and analyze endpoint data. When integrated with ISE, Device Sensor provides central policy management and device-profiling capabilities.

For more information on Device Sensor, see [Chapter 49, “Configuring 802.1X Port-Based Authentication.”](#)

EIGRP Stub Routing

The EIGRP stub routing feature, available in all images, reduces resource utilization by moving routed traffic closer to the end user.

The IP base image contains only EIGRP stub routing. The IP services image contains complete EIGRP routing.

In a network using EIGRP stub routing, the only route for IP traffic to follow to the user is through a switch that is configured with EIGRP stub routing. The switch sends the routed traffic to interfaces that are configured as user interfaces or are connected to other devices.

For information on configuring EIGRP Stub Routing, see [Chapter 36, “About Layer 3 Interfaces.”](#)

Enhanced Object Tracking

Before the introduction of the Enhanced Object Tracking feature, the Hot Standby Router Protocol (HSRP) had a simple tracking mechanism that allowed you to track the interface line-protocol state only. If the line-protocol state of the interface went down, the HSRP priority of the router was reduced, allowing another HSRP router with a higher priority to become active.

The Enhanced Object Tracking (EOT) feature separates the tracking mechanism from HSRP and creates a separate standalone tracking process that can be used by other Cisco IOS processes as well as HSRP. This feature allows tracking of other objects in addition to the interface line-protocol state.

A client process, such as HSRP, Virtual Router Redundancy Protocol (VRRP), or Gateway Load Balancing Protocol (GLBP), can now register its interest in tracking objects and then be notified when the tracked object changes state.

For details on EOT, refer to this URL:

For platform specific information on Enhanced Object Tracking, see [Chapter 72, “Configuring Enhanced Object Tracking.”](#)

For more detailed information on Enhanced Object Tracking, see the URL:

<http://www.cisco.com/en/US/docs/ios-xml/ios/ipapp/configuration/12-4t/iap-eot.html>

GLBP

The Gateway Load Balancing Protocol (GLBP) feature provides automatic router backup for IP hosts configured with a single default gateway on a LAN. Multiple first hop routers on the LAN combine to offer a single virtual first hop IP router while sharing the IP packet forwarding load. GLBP devices share packet-forwarding responsibilities, optimizing resource usage, thereby reducing costs. Other routers on the LAN may act as redundant GLBP routers that will become active if any of the existing forwarding routers fail. This improves the resiliency of the network and reduces administrative burden.

For details on GLBP, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_2t/12_2t15/feature/guide/ft_glb.html

Cisco IOS XE IP Application Services Features in Cisco IOS XE 3.1.0SG

This section lists the IP Application Services software features that are supported in Cisco IOS XE 3.1.0SG. Links to the feature documentation are included.

Feature guides may contain information about more than one feature. To find information about a specific feature within a feature guide, see the Feature Information table at the end of the guide.

Feature guides document features that are supported on many different software releases and platforms. Your Cisco software release or platform may not support all the features documented in a feature guide. See the Feature Information table at the end of the feature guide for information about which features in that guide are supported in your software release. Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Gateway Load Balancing Protocol (GLBP), GLBP MD5 Authentication

http://www.cisco.com/en/US/docs/ios/12_2sx/12_2sxh/feature/guide/sxglbpm.html

HSRP

The Hot Standby Router Protocol (HSRP) provides high network availability by routing IP traffic from hosts on Ethernet networks without relying on the availability of any single Layer 3 switch. This feature is particularly useful for hosts that do not support a router discovery protocol and do not have the functionality to switch to a new router when their selected router reloads or loses power.

For information on configuring HSRP, refer to the following URL:

http://www.cisco.com/en/US/tech/tk648/tk362/tk321/tsd_technology_support_sub-protocol_home.html

Cisco IOS XE IP Application Services: HSRP Features in Cisco IOS XE 3.1.0SG

This section lists the IP Application Services:HSRP software features that are supported in Cisco IOS XE 3.1.0SG. Links to the feature documentation are included.

Feature guides may contain information about more than one feature. To find information about a specific feature within a feature guide, see the Feature Information table at the end of the guide.

Feature guides document features that are supported on many different software releases and platforms. Your Cisco software release or platform may not support all the features documented in a feature guide. See the Feature Information table at the end of the feature guide for information about which features in that guide are supported in your software release. Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

HSRP—Hot Standby Router Protocol

http://www.cisco.com/en/US/docs/ios/ipapp/configuration/guide/ipapp_hsrp.html

HSRP MD5 Authentication

http://www.cisco.com/en/US/docs/ios/ipapp/configuration/guide/ipapp_hsrp.html

HSRP Support for ICMP Redirects

http://www.cisco.com/en/US/docs/ios/ipapp/configuration/guide/ipapp_hsrp.html

IP Precedence Accounting

http://www.cisco.com/en/US/docs/ios/12_2/ipaddr/command/reference/1rfip2.html

ISSU—HSRP

http://www.cisco.com/en/US/docs/ios/ipapp/configuration/guide/ipapp_hsrp.html

SSO—HSRP

http://www.cisco.com/en/US/docs/ios/ipapp/configuration/guide/ipapp_hsrp.html

SSO Aware HSRP

SSO Aware HSRP offers continuous data packet forwarding during a supervisor engine switchover without a path change to the standby HSRP router. During supervisor engine switchover, NSF with SSO continues forwarding data packets along known routes using the HSRP virtual IP address. When both supervisor engines fail on the active HSRP router, the standby HSRP router takes over as the active HSRP router. It further extends reliability and availability offered by the NSF with SSO to Layer 3. SSO aware HSRP is available for Supervisor Engine IV, V, and V-10GE on Catalyst 4507R and 4510R chassis with supervisor redundancy.

NHRP

Beginning in Cisco IOS XE Release 3.7.1E, NHRP is supported on Catalyst 4500 series switches.

The Next Hop Resolution Protocol (NHRP) is an ARP-like protocol that dynamically maps a Non-Broadcast Multi-Access (NBMA) network. With NHRP, systems attached to an NBMA network can dynamically learn the NBMA (physical) address of the other systems that are part of that network, allowing these systems to directly communicate.

NHRP is a client and server protocol where the hub is the Next Hop Server (NHS) and the spokes are the Next Hop Clients (NHCs). Catalyst 4500 series switches act as NHRP clients that communicate with the NHRP hub for registration, and to request the resolution of other spoke addresses.

**Note**

Catalyst 4500 series switches cannot be used as NHRP hubs.

For information on configuring NHRP, see

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipaddr_nhrp/configuration/xe-3s/asr1000/nhrp-xe-3s-asr1000-book/config-nhrp.html

IP Routing Protocols

The following routing protocols are supported on the Catalyst 4500 series switch:

- [BGP, page 1-17](#)
- [EIGRP, page 1-18](#)
- [IS-IS, page 1-18](#)
- [OSPF, page 1-19](#)
- [RIP, page 1-19](#)

BGP

The Border Gateway Protocol (BGP) is an exterior gateway protocol that allows you to set up an interdomain routing system to automatically guarantee the loop-free exchange of routing information between autonomous systems. In BGP, each route consists of a network number, a list of autonomous systems that information has passed through (called the autonomous system path), and a list of other path attributes.

The Catalyst 4500 series switch supports BGP version 4, including classless interdomain routing (CIDR). CIDR lets you reduce the size of your routing tables by creating aggregate routes, resulting in supernets. CIDR eliminates the concept of network classes within BGP and supports the advertising of IP prefixes. CIDR routes can be carried by OSPF, EIGRP, and RIP.

BGP Route-Map Continue

The BGP Route-Map Continue feature introduces the continue clause to the BGP route-map configuration. The continue clause provides more programmable policy configuration and route filtering. It introduces the capability to execute additional entries in a route map after an entry is executed with successful match and set clauses. Continue clauses allow configuring and organizing more modular policy definitions to reduce the number of policy configurations that are repeated within the same route map.

For details on BGP, refer to this URL:

http://www.cisco.com/en/US/docs/ios/12_4t/ip_route/configuration/guide/t_brbbas.html

EIGRP

The Enhanced Interior Gateway Routing Protocol (EIGRP) is a version of IGRP that combines the advantages of link-state protocols with distance-vector protocols. EIGRP incorporates the Diffusing Update Algorithm (DUAL). EIGRP includes fast convergence, variable-length subnet masks, partially bounded updates, and multiple network-layer support. When a network topology change occurs, EIGRP checks its topology table for a suitable new route to the destination. If such a route exists in the table, EIGRP updates the routing table instantly. You can use the fast convergence and partial updates that EIGRP provides to route Internetwork Packet Exchange (IPX) packets.

EIGRP saves bandwidth by sending routing updates only when routing information changes. The updates contain information only about the link that changed, not the entire routing table. EIGRP also takes into consideration the available bandwidth when determining the rate at which it transmits updates.



Note

Layer 3 switching does not support the Next Hop Resolution Protocol (NHRP).



Note

Customers can configure Enhanced Interior Gateway Routing Protocol (EIGRP) to route IPv6 prefixes. EIGRP configuration and protocol behavior for both IPv4 and IPv6 prefixes are similar, providing operational familiarity and continuity. EIGRP support for IPv6 will enable customers to use their existing EIGRP knowledge and processes, allowing them to deploy an IPv6 network at a low cost.

For details on EIGRP, refer to this URL:

http://www.cisco.com/en/US/products/ps6630/products_ios_protocol_option_home.html

IS-IS

The Intermediate System-to-Intermediate System Protocol (IS-IS Protocol) uses a link-state routing algorithm. It closely follows the Open Shortest Path First (OSPF) routing protocol used within the TCP/IP environment. The operation of ISO IS-IS Protocol requires each router to maintain a full topology map of the network (that is, which intermediate systems and end systems are connected to which other intermediate systems and end systems). Periodically, the router runs an algorithm over its map to calculate the shortest path to all possible destinations.

The IS-IS Protocol uses a two-level hierarchy. Intermediate Systems (or routers) are classified as Level 1 and Level 2. Level 1 intermediate systems deal with a single routing area. Traffic is relayed only within that area. Any other internetwork traffic is sent to the nearest Level 2 intermediate systems, which also acts as a Level 1 intermediate systems. Level 2 intermediate systems move traffic between different routing areas within the same domain.

An IS-IS with multi-area support allows multiple Level 1 areas within in a single intermediate system, thus allowing an intermediate system to be in multiple areas. A single Level 2 area is used as backbone for inter-area traffic.

For details on IS-IS, refer to this URL:

http://www.cisco.com/en/US/products/ps6632/products_ios_protocol_option_home.html

OSPF

The Open Shortest Path First (OSPF) protocol is a standards-based IP routing protocol designed to overcome the limitations of RIP. Because OSPF is a link-state routing protocol, it sends link-state advertisements (LSAs) to all other routers within the same hierarchical area. Information on the attached interfaces and their metrics is used in OSPF LSAs. As routers accumulate link-state information, they use the shortest path first (SPF) algorithm to calculate the shortest path to each node. Additional OSPF features include equal-cost multipath routing and routing based on the upper-layer type of service (ToS) requests.

OSPF uses the concept of an *area*, which is a group of contiguous OSPF networks and hosts. OSPF areas are logical subdivisions of OSPF autonomous systems in which the internal topology is hidden from routers outside the area. Areas allow an additional level of hierarchy different from that provided by IP network classes, and they can be used to aggregate routing information and mask the details of a network. These features make OSPF particularly scalable for large networks.



Note

For Catalyst 4500 series switch, usage of OSPF aggressive timers may lead to session transition and packet loss during CPU-intensive operation. It is recommended to use either default OSPF timers or BFD with hardware offload functionality.

For details on OSPF, refer to this URL:

http://www.cisco.com/en/US/tech/tk365/tk480/tsd_technology_support_sub-protocol_home.html

RIP

The Routing Information Protocol (RIP) is a distance-vector, intradomain routing protocol. RIP works well in small, homogeneous networks. In large, complex internetworks it has many limitations, such as a maximum hop count of 15, lack of support for variable-length subnet masks (VLSMs), inefficient use of bandwidth, and slow convergence. RIP II does support VLSMs.

For details on RIP, refer to this URL:

http://www.cisco.com/en/US/tech/tk365/tk554/tsd_technology_support_sub-protocol_home.html

In Service Software Upgrade

SSO requires the same version of Cisco IOS on both the active and standby supervisor engines. Because of version mismatch during an upgrade or downgrade of the Cisco IOS software, a Catalyst 4500 series switch is forced into operating in RPR mode. In this mode, after the switchover you can observe link-flaps and a disruption in service. This issue is solved by the In-Service Software Upgrade (ISSU) feature that enables you to operate in SSO/NSF mode while performing software upgrade or downgrade.

ISSU allows an upgrade or downgrade of the Catalyst IOS or IOS XE images at different release levels on the both the active and standby supervisor engines by utilizing the Version Transformation Framework between the stateful components running on each supervisor engine.

For details on Cisco IOS ISSU, refer to [Chapter 7, “Configuring the Cisco IOS In-Service Software Upgrade Process.”](#)

For details on Cisco IOS XE ISSU, refer to [Chapter 8, “Configuring the Cisco IOS XE In Service Software Upgrade Process.”](#)

IPv6

IPv6 provides services such as end-to-end security, quality of service (QoS), and globally unique addresses. The IPv6 address space reduces the need for private addresses and Network Address Translation (NAT) processing by border routers at network edges.

For more information about IPv6 services supported on a Catalyst 4500 series switch, see [Chapter 64, “Support for IPv6.”](#)

Multicast Services

Multicast services save bandwidth by forcing the network to replicate packets only when necessary and by allowing hosts to join and leave groups dynamically. The following multicast services are supported:

- ANCP Client —ANCP Multicast enables you to control multicast traffic on a Catalyst 4500 switch using either ANCP (rather than IGMP) or direct static configuration on the CLI.
- Cisco Group Management Protocol (CGMP) server—CGMP server manages multicast traffic. Multicast traffic is forwarded only to ports with attached hosts that request the multicast traffic.
- Internet Group Management Protocol (IGMP) snooping—IGMP snooping manages multicast traffic. The switch software examines IP multicast packets and forwards packets based on their content. Multicast traffic is forwarded only to ports with attached hosts that request multicast traffic.

Support for IGMPv3 provides constrained flooding of multicast traffic in the presence of IGMPv3 hosts or routers. IGMPv3 snooping listens to IGMPv3 query and membership report messages to maintain host-to-multicast group associations. It enables a switch to propagate multicast data only to ports that need it. IGMPv3 snooping is fully interoperable with IGMPv1 and IGMPv2.

Explicit Host Tracking (EHT) is an extension to IGMPv3 snooping. EHT enables immediate leave operations on a per-port basis. EHT can be used to track per host membership information or to gather statistics about all IGMPv3 group members.

The IGMP Snooping Querier is a Layer 2 feature required to support IGMP snooping in a VLAN where PIM and IGMP are not configured because the multicast traffic does not require routing.

For information on configuring IGMP snooping, see [Chapter 28, “Configuring IGMP Snooping and Filtering, and MVR.”](#)

- IPv6 Multicast Listen Discovery (MLD) and Multicast Listen Discovery snooping—MLD is a protocol used by IPv6 multicast devices to discover the presence of multicast listeners (nodes that want to receive IPv6 multicast packets) on its directly attached links and to discover which multicast packets are of interest to neighboring nodes. MLD snooping is supported in two different versions: MLD v1 and MLD v2. Network switches use MLD snooping to limit the flood of multicast traffic, causing IPv6 multicast data to be selectively forwarded to a list of ports that want to receive the data, instead of being flooded to all ports in a VLAN. This lessens the load on devices in the network, minimizing unnecessary bandwidth on links, enabling efficient distribution of IPv6 multicast data.

For information on configuring multicast services, see [Chapter 30, “Configuring IPv6 Multicast Listener Discovery Snooping.”](#)

- Protocol Independent Multicast (PIM)—PIM is protocol-independent because it can leverage whichever unicast routing protocol is used to populate the unicast routing table, including EIGRP, OSPF, BGP, or static route. PIM also uses a unicast routing table to perform the Reverse Path Forwarding (RPF) check function instead of building a completely independent multicast routing table.

For information on PIM-SSM mapping, see the URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/15.1SY/config_guide/sup720/ude_udlr.html

- IP Multicast Load Splitting (Equal Cost Multipath (ECMP) Using S, G and Next Hop)—IP Multicast Load Splitting introduces more flexible support for ECMP multicast load splitting by adding support for load splitting based on source and group address and on source, group, and next-hop address. This feature allows multicast traffic from devices that send many streams to groups or that broadcast many channels, such as IPTV servers or MPEG video servers, to be more effectively load shared across equal-cost paths.

For information on configuring multicast services, see [Chapter 39, “Configuring IP Multicast.”](#)

NSF with SSO

Non-Stop Forwarding with Stateful Switchover (NSF/SSO) offers continuous data packet forwarding in a Layer 3 routing environment during supervisor engine switchover. During supervisor engine switchover, NSF/SSO continues forwarding data packets along known routes while the routing protocol information is recovered and validated, avoiding unnecessary route flaps and network instability. With NSF/SSO, IP phone calls do not drop. NSF/SSO is supported for OSPF, BGP, EIGRP, IS-IS, and Cisco Express Forwarding (CEF). NSF/SSO is typically deployed in the most critical parts of an enterprise or service provider network, such as Layer 3 aggregation/core or a resilient Layer 3 wiring closet design. It is an essential component of single chassis deployment for critical applications. NSF/SSO is available for all shipping supervisor engines on Catalyst 4507R and 4510R chassis with supervisor redundancy.



Note

With the IP Base image, NSF is supported with EIGRP-stub routing and OSPF.



Note

With the Enterprise Services image, NSF is supported on all routing protocols except for RIP.



Note

The LAN Base image does not support NSF.

For information on NSF with SSO, see [Chapter 13, “Configuring Cisco NSF with SSO Supervisor Engine Redundancy.”](#)

OSPF for Routed Access

OSPF for Routed Access is designed specifically to enable customers to extend Layer 3 routing capabilities to the access or wiring closet.

**Note**

OSPF for Routed Access supports only one OSPFv2 and one OSPFv3 instance with a maximum number of 1000 dynamically learned routes.

With the typical topology (hub and spoke) in a campus environment, where the wiring closets (spokes) are connected to the distribution switch (hub) forwarding all nonlocal traffic to the distribution layer, the wiring closet switch does not need to hold a complete routing table. Ideally, the distribution switch sends a default route to the wiring closet switch to reach inter-area and external routes (OSPF stub or totally stub area configuration).

Refer to the following link for more details:

<http://www.cisco.com/en/US/docs/solutions/Enterprise/Campus/routed-ex.html>

With Cisco IOS Release 12.2(53)SG, the IP Base image supports OSPF for routed access. The Enterprise Services image is required if you need multiple OSPFv2 and OSPFv3 instances without route restrictions. Enterprise Services also is required to enable the VRF-lite feature.

Policy-Based Routing

Traditional IP forwarding decisions are based purely on the destination IP address of the packet being forwarded. Policy-Based Routing (PBR) enables forwarding based upon other information associated with a packet, such as the source interface, IP source address, Layer 4 ports, and so on. This feature allows network managers more flexibility in how they configure and design their networks.

Starting with Release IOS XE 3.4.0SG and IOS 15.1(2)SG, the PBR Recursive Next Hop feature enhances route maps to enable configuration of a recursive next-hop IP address. The recursive next-hop IP address can be a subnet that is not directly connected. The routing table will be looked up to find the directly connected next-hop to which to send the packet so that it is routed via the recursive next-hop that has been configured.

For more information on policy-based routing, see [Chapter 42, “Configuring Policy-Based Routing.”](#)

Unicast Reverse Path Forwarding

The Unicast Reverse Path Forwarding (Unicast RPF) feature helps to mitigate problems that are caused by the introduction of malformed or forged (spoofed) IP source addresses into a network by discarding IP packets that lack a verifiable IP source address.

For information on URPF, see [Chapter 38, “Configuring Unicast Reverse Path Forwarding.”](#)

REVIEW DRAFT—CISCO CONFIDENTIAL

Unidirectional Link Routing

Unidirectional link routing (UDLR) provides a way to forward multicast packets over a physical unidirectional interface (such as a satellite link of high bandwidth) to stub networks that have a back channel.

For information on configuring unidirectional link routing, refer to the URL

http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/15.1SY/config_guide/sup720/ude_udlr.html

VRF-lite

VPN routing and forwarding (VRF-lite) is an extension of IP routing that provides multiple routing instances. Along with BGP, it enables the creation of a Layer 3 VPN service by keeping separate IP routing and forwarding tables for each VPN customer. VRF-lite uses input interfaces to distinguish routes for different VPNs. It forms virtual packet-forwarding tables by associating one or more Layer 3 interfaces with each VRF, allowing the creation of multiple Layer 3 VPNs on a single switch. Interfaces in a VRF could be either physical, such as an Ethernet port, or logical, such as a VLAN switch virtual interface (SVI). However, interfaces cannot belong to more than one VRF at any time.

Prior to Release IOS XE 3.5.0E and IOS 15.2(1)E, only IPv4 was available. With Release IOS XE 3.5.0E and IOS 15.2(1)E, VRF-lite support has been extended to IPv6.

For information on VRF-lite, see [Chapter 43, “Configuring VRF-lite.”](#)

Virtual Router Redundancy Protocol

Virtual Router Redundancy Protocol (VRRP) is a standard based first-hop redundancy protocol. With VRRP, a group of routers function as one virtual router by sharing one virtual IP address and one virtual MAC address. The master router performs packet forwarding, while the backup routers stay idle. VRRP is typically used in the multi-vendor first-hop gateway redundancy deployment.

For details on VRRP, refer to this URL:

http://www.cisco.com/en/US/docs/ios/ipapp/configuration/guide/ipapp_vrrp_ps6441_TSD_Products_Configuration_Guide_Chapter.html

Management Features

The Catalyst 4500 series switch offers network management and control using the CLI or through alternative access methods, such as SNMP. The switch software supports these network management features:

- [Cisco Call Home, page 1-24](#)
- [Cisco Energy Wise, page 1-25](#)
- [Cisco IOS IP Service Level Agreements, page 1-25](#)
- [Cisco Media Services Proxy, page 1-25](#)
- [Cisco Medianet AutoQoS, page 1-26](#)
- [Cisco Medianet Flow Metadata, page 1-26](#)

- [Cisco IOS Mediatrace and Performance Monitor](#), page 1-27
- [Cisco Network Assistant](#), page 1-28
- [Dynamic Host Control Protocol](#), page 1-28
- [Easy Virtual Network](#), page 1-29
- [Embedded CiscoView](#), page 1-29
- [Embedded Event Manager](#), page 1-29
- [Ethernet Management Port](#), page 1-30
- [File System Management \(Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, and 7-E\)](#), page 1-30
- [FAT File Management System on Supervisor Engine 6-E, Supervisor Engine 6L-E](#), page 1-30
- [Forced 10/100 Autonegotiation](#), page 1-30
- [Intelligent Power Management](#), page 1-30
- [MAC Address Notification](#), page 1-31
- [MAC Notify MIB](#), page 1-31
- [Power over Ethernet](#), page 1-31
- [Power over Ethernet](#), page 1-31
- [Secure Shell](#), page 1-31
- [Simple Network Management Protocol](#), page 1-31
- [Smart Install](#), page 1-32
- [SPAN and RSPAN](#), page 1-32
- [Universal Power over Ethernet](#), page 1-33
- [Web Content Coordination Protocol](#), page 1-33
- [Wireshark](#), page 1-33
- [XML-PI](#), page 1-34

Cisco Call Home

Call Home provides e-mail-based and web-based notification of critical system events. A versatile range of message formats are available for optimal compatibility with pager services, standard e-mail, or XML-based automated parsing applications. Common uses of this feature may include direct paging of a network support engineer, e-mail notification to a Network Operations Center, XML delivery to a support website, and utilization of Cisco Smart Call Home services for direct case generation with the Cisco Systems Technical Assistance Center (TAC).

The Call Home feature can deliver alert messages containing information on configuration, diagnostics, environmental conditions, inventory, and syslog events.

For more information on Call Home, see [Chapter 79, “Configuring Call Home.”](#)

Cisco Energy Wise

Cisco EnergyWise is an energy-management technology added onto Cisco switching solutions to help you measure, report, and reduce energy consumption across your entire infrastructure. With EnergyWise's management interface, network management applications can communicate with endpoints and each other, using the network as the unifying fabric.

For details refer to the URLs:

http://www.cisco.com/en/US/docs/switches/lan/energywise/phase2/ios/configuration/guide/ew_v2.html

http://www.cisco.com/en/US/docs/switches/lan/energywise/phase2_5/ios/release/notes/ol23554.html#wp604941

Cisco IOS IP Service Level Agreements

Cisco IOS IP Service Level Agreements (SLAs) allows Cisco customers to analyze IP service levels for IP applications and services by using active traffic monitoring—the generation of traffic in a continuous, reliable, and predictable manner—for measuring network performance. With Cisco IOS IP SLA, service provider customers can measure and provide service level agreements, and enterprise customers can verify service levels, verify outsourced service level agreements, and understand network performance. Cisco IOS IP SLAs can perform network assessments, verify quality of service (QoS), ease the deployment of new services, and assist with network troubleshooting.

For platform-specific information on Cisco IOS IP SLA, see [Chapter 80, “Configuring Cisco IOS IP SLA Operations.”](#)

For more detail on Cisco IOS IP SLAs, see the *Cisco IOS IP SLAs Configuration Guide, Release 12.4T*: http://www.cisco.com/en/US/docs/ios/ipsla/configuration/guide/12_4t/sla_12_4t_book.html

A Catalyst 4500 series switch also supports a Built-in Traffic Simulator using Cisco IOS IP SLAs video operations to generate synthetic traffic for a variety of video applications, such as Telepresence, IPTV and IP video surveillance camera. You can use the simulator tool:

- for network assessment before deploying applications that have stringent network performance requirements.
- along with the Cisco IOS Mediatrace for post-deployment troubleshooting for any network related performance issues.

The traffic simulator includes a sophisticated scheduler that allows the user to run several tests simultaneously or periodically and over extended time periods. (Supported only on switches running the Enterprise Services feature set.)

For information on configuring this feature, see the *Configuring Cisco IOS IP SLAs Video Operations* document at:

http://www.cisco.com/en/US/docs/ios-xml/ios/ipsla/configuration/12-2se/sla_video.html

Cisco Media Services Proxy

The Media Services Proxy (MSP) feature identifies various media end points in the network automatically and renders appropriate media services. It acts as a layer that connects appropriate devices with their respective network services automatically.

MSP follows a network-centric model, where the access switches and routers learn information about devices and flow by using mechanisms such as Cisco Discovery Protocol (formerly known as CDP) and DHCP, or by snooping on key protocol packets such as the Session Initiation Protocol (SIP) and H.323. Modifications to the endpoints are not required to achieve the information learning. After the information is gleaned, MSP provides appropriate services to the network devices.

Following are the benefits of MSP:

- Automatic identification of devices and flow in the network.
- Application of appropriate services to the endpoints.
- Configuration control for the administrator, thereby reducing the manual configuration and management of services. For example, configuring the Resource Reservation Protocol (RSVP) in the network for video applications requiring guaranteed bandwidth.



Note The system cannot scale to greater than 512 SIP flows with MSP and Flow Metadata enabled.

For information on configuring this feature, refer to the following documents:

<http://www.cisco.com/en/US/docs/ios-xml/ios/msp/configuration/15-1sg/med-ser-prxy.html>

<http://www.cisco.com/en/US/docs/ios-xml/ios/msp/configuration/xe-3sg/med-ser-prxy-xe.html>

Cisco Medianet AutoQoS

Cisco Medianet AutoQoS provides a default configuration to ease the process of enabling QoS on switches. This process can be difficult given the functional/behavioral differences in QoS across different platforms. This functionality extend AutoQoS functionality for the Catalyst 4500 to support video traffic as well as other kinds of traffic.

The goal of AutoQoS is to simplify the work customers have to undertake while configuring their networks to support QoS. This is done by automating QoS configurations to handle various classes of traffic. AutoQoS for Medianet provides commands, which act as macros that call existing CLI commands to implement desired configurations. You are required to specify the type of device (PC, another switch, ip camera, etc.) connected to a given interface. AutoQoS for Medianet applies a default QoS configuration to that interface, which you can later fine-tune as needed.

For details, refer to [Chapter 44, “Configuring Quality of Service.”](#)

Cisco Medianet Flow Metadata

Flow Metadata is the data that qualifies other data. Flow Metadata aids in supporting an intelligent network by making the network aware about the type, nature, and characteristics of the media stream that flows in the network. Flow Metadata also allows for the network to apply policies on the media streams. Across the Medianet system, Flow Metadata is produced, transported, stored, retrieved, and acted on consistently by a wide variety of Medianet services.

The Flow Metadata infrastructure provides a framework that allows data from one component be available to another component on the same network element as well as across network elements.

Flow Metadata is supported on releases prior to Cisco IOS Release 15.1(1)SG. Flow metadata is the data that describes a flow in the network. This Flow Metadata describes the five tuple flow along with the attributes. Network elements can take action based on the Flow Metadata generated by the endpoints.

The Flow Metadata infrastructure consists of two major components: producers and consumers.

- Flow Metadata producer is any source of Flow Metadata. The producer propagates all the attributes of a given flow. Producers can be anywhere in the network: endpoint, proxy agents, or intermediate nodes. Currently, Flow Metadata generated by the endpoints is supported. Producers use a specific transport protocol, such as RSVP for signalling the Flow Metadata attributes and store the information in a database, referred to as the control plane database, which can then be used by the consumers.
- Flow Metadata consumer is any network element that uses the flow tuple and Flow Metadata provided by the producers. The flow tuple and Flow Metadata can also be propagated along the media path to consumers in different network elements via a transport infrastructure.

For configuration details, refer to the following URLs:

<http://www.cisco.com/en/US/docs/ios-xml/ios/mdata/configuration/xe-3sg/metadata-framework.html>

<http://www.cisco.com/en/US/docs/ios-xml/ios/mdata/configuration/15-1sg/metadata-framework.html>

For details on the Flow Metadata commands, refer to the following URL:

<http://www.cisco.com/en/US/docs/ios-xml/ios/qos/command/qos-cr-book.html>

Cisco IOS Mediatrace and Performance Monitor

Cisco IOS Mediatrace helps to isolate and troubleshoot network degradation problems by enabling a network administrator to discover an IP flow's path, dynamically enable monitoring capabilities on the nodes along the path, and collect information on a hop-by-hop basis. This information includes, among other things, flow statistics; utilization information for incoming and outgoing interfaces, CPUs, and memory; as well as any changes to IP routes or the Cisco IOS Mediatrace monitoring state.

For details, see the following URLs:

http://www.cisco.com/en/US/docs/ios-xml/ios/media_monitoring/configuration/15-1sg/mm-pasv-mon.html

http://www.cisco.com/en/US/docs/ios-xml/ios/media_monitoring/configuration/xe-3sg/mm-pasv-mon.html

http://www.cisco.com/en/US/docs/ios/media_monitoring/command/reference/mm_book.html

http://www.cisco.com/en/US/docs/ios-xml/ios/media_monitoring/configuration/15-1sg/mm-mediatriace.html

http://www.cisco.com/en/US/docs/ios-xml/ios/media_monitoring/configuration/xe-3sg/mm-mediatriace.html

Configuration guidelines for Cisco IOS Mediatrace and Performance Monitor include the following:

- Video monitoring is supported only on physical ports.

Limitations for Cisco IOS Mediatrace and Performance Monitor on a Catalyst 4500 series switch include the following:

- Both features can only be configured to monitor ingress traffic.
- Packets cannot be monitored by both CEure and the rxSPAN session with encapsulation. The first-applied configuration takes precedence.
- Not all packets received by an interface can be monitored. After a packet is received by an ingress interface, it might be either unable to make a forwarding decision or dropped at various stages because of configured security features (like IP Source Guard). The switch attempts to monitor packets close to the switch, but only those that are not dropped before the input classification stage can be monitored.

REVIEW DRAFT—CISCO CONFIDENTIAL

- CPU utilization is impacted when you monitor a high traffic rate. After the internally-determined threshold is crossed, monitored packets are dropped although the original packet is forwarded in hardware intact. Starting with Release IOS XE 3.3.0SG and IOS 15.1(1)SG, monitored packets might be dropped if any of the following apply:
 - The packet rate exceeds 512 PPS per flow.
 - The aggregated bandwidth of the monitor traffic exceeds 10Mbps.
 - Resources are insufficient to enqueue a new monitored packet.

When monitored packets are dropped, the *monitor event* will be set to TRUE if the flow record contains *collect monitor event*. If one minute passes with no new drops, the *monitor event* is set to FALSE but is not reflected in the output of the **show performance monitor status** until the new monitor interval starts.

monitor event is a global flag. This means that any packet drops that would trigger "monitor event" be set to TRUE for all monitored flows at that monitor interval. If a metric depends on the collection of continuous packets, the accuracy of that metric might be impacted when a *monitor event* is TRUE.

Cisco Network Assistant

Cisco Network Assistant manages standalone devices, clusters of devices, or federations of devices from anywhere in your intranet. Using its graphical user interface, you can perform multiple configuration tasks without having to remember command-line interface commands. Embedded CiscoView is a device management application that can be embedded on the switch flash and provides dynamic status, monitoring, and configuration information for your switch.

For more information on Cisco Network Assistant, see [Chapter 16, “Configuring Catalyst 4500 Series Switches with Cisco Network Assistant.”](#)

Dynamic Host Control Protocol

The Catalyst 4500 series switch uses DHCP in the following ways:

- Dynamic Host Control Protocol server—The Cisco IOS DHCP server feature is a full DHCP server implementation that assigns and manages IP addresses from specified address pools within the router to DHCP clients. If the Cisco IOS DHCP server cannot satisfy a DHCP request from its own database, it can forward the request to one or more secondary DHCP servers defined by the network administrator.
- Dynamic Host Control Protocol autoconfiguration—With this feature your switch (the DHCP client) is automatically configured at startup with IP address information and a configuration file.

For DHCP server configuration information, refer to the chapter, “Configuring DHCP,” in the *Cisco IOS IP and IP Routing Configuration Guide* at the following URL:

http://www.cisco.com/en/US/docs/ios/ipaddr/configuration/guide/iad_dhcp_rdmf_ps6350_TSD_Products_Configuration_Guide_Chapter.html

Easy Virtual Network

Easy Virtual Network (EVN) is an IP-based virtualization technology that provides end-to-end virtualization of the network. You can use a single IP infrastructure to provide separate virtual networks whose traffic paths remain isolated from each other. Configure Easy Virtual Network to configure two or more virtual IP networks.

For details on EVN, refer to the following URLs:

<http://www.cisco.com/en/US/docs/ios-xml/ios/evn/configuration/xe-3s/evn-xe-3s-book.html>

The following restrictions/feature interactions apply:

- Multicast

When multicast traffic traverses VRFs, where source and receivers are in different VRFs, multicast counters will not increment on receivers' VRFs.

- NetFlow

When configured on an EVN trunk interface, NetFlow captures traffic information for all VRFs but does not preserve the VRF information.

- SPAN

- When an EVN trunk interface is configured as a SPAN source, traffic belonging to all VRFs carried by the EVN trunk is spanned. By default, the VNET tag is not preserved. To preserve it, configure SPAN destination with the **encapsulation dot1q** option.
- To span traffic belonging to specific VRFs on an EVN trunk, configure **filter vlan** on the SPAN session with the corresponding VNET tags as `vlan_ids` and configure VLANs specified in **filter vlan**.
- To span traffic in specific VRFs on all interfaces, configure **vlan** as the SPAN source with VNET tags as `vlan_ids` and configure VLANs specified as sources.
- If `cpu` is configured as a SPAN source, then transmit packets that will be spanned are tagged by default. If the **encapsulation dot1q** option is set on the SPAN session, then the `cpu` transmitted packets, which are spanned, are double tagged.

Refer to [Chapter 69, “Configuring SPAN and RSPAN,”](#) for information on configuring SPAN sessions.

Embedded CiscoView

A web-based tool to configure the Catalyst 4500 series switch. Embedded CiscoView is a device management application that can be embedded on the switch flash and provides dynamic status, monitoring, and configuration information for your switch.

For more information on Embedded CiscoView, see [Chapter 4, “Administering the Switch.”](#)

Embedded Event Manager

Embedded Event Manager (EEM) is a distributed and customized approach to event detection and recovery offered directly in a Cisco IOS device. EEM offers the ability to monitor events and take informational, corrective, or any desired EEM action when the monitored events occur or when a threshold is reached. An EEM policy is an entity that defines an event and the actions to be taken when that event occurs.

For information on EEM, see the URL:

http://www.cisco.com/en/US/products/ps6815/products_ios_protocol_group_home.html

Ethernet Management Port

The Ethernet management port, also referred to as the *Fal or fastethernet1 port*, is a Layer 3 host port to which you can connect a PC. You can use the Ethernet management port instead of the switch console port for network management. When managing a switch stack, connect the PC to the Ethernet management port on a Catalyst 4500 series switch.

For more information on Ethernet management port, see the “Using the Ethernet Management Port” section in [Chapter 9, “Configuring Interfaces.”](#)

File System Management (Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, and 7-E)

The format command for IOS XE 3.1.0SG changed slightly compared to the classic IOS format because the later does not support ext2 format.

For USB flash under IOS XE 3.1.0SG, there are 3 optional formats, i.e. FAT16, FAT32 and EXT2:

```
Switch# format usb0: ?  
FAT16  FAT16 filesystem type  
FAT32  FAT32 filesystem type  
ext2   ext2 filesystem type
```

For SD card under IOS XE 3.1.0SG, the default format is FAT16:

```
Switch# format slaveusb0: ?  
FAT16  FAT16 filesystem type  
FAT32  FAT32 filesystem type  
ext2   ext2 filesystem type
```

FAT File Management System on Supervisor Engine 6-E, Supervisor Engine 6L-E

The FAT file system is widely used to manage files on devices disks and flash. The support of the FAT file system allows you to easily remove, add, and/or transfer images to and from the flash.

Forced 10/100 Autonegotiation

This feature allows you to configure a port to limit the speed at which it will autonegotiate to a speed lower than the physically maximum speed. This method of reducing the throughput incurs much less overhead than using an ACL.

Intelligent Power Management

Working with powered devices (PDs) from Cisco, this feature uses power negotiation to refine the power consumption of an 802.3af-compliant PD beyond the granularity of power consumption provided by the 802.3af class. Power negotiation also enables the backward compatibility of newer PDs with older modules that do not support either 802.3af or high-power levels as required by IEEE standard.

For more information on Intelligent Power Management, see the “Intelligent Power Management” section in [Chapter 15, “Configuring Power over Ethernet.”](#)

MAC Address Notification

MAC address notification monitors the MAC addresses that are learned by, aged out, or removed from the Catalyst 4500 series switch. Notifications are sent out or retrieved by using the CISCO-MAC-NOTIFICATION MIB. It is typically used by a central network management application to collect such MAC address notification events for host moves. User-configurable MAC table utilization thresholds can be defined to notify any potential DoS or man-in-the-middle attack.

For information on MAC Address Notification, see [Chapter 4, “Administering the Switch.”](#)

MAC Notify MIB

The MAC Notify MIB feature monitors network performance, utilization, and security conditions enabling a network administrator to track the MAC addresses that are learned or removed on the switch forwarding the Ethernet frames.

Power over Ethernet

Power over Ethernet (PoE) allows the LAN switching infrastructure to provide power to an endpoint ("powered device") over a copper Ethernet cable. This capability, once referred to as "inline power," was originally developed by Cisco in 2000 to support emerging IP telephony deployments.

IP telephones need power for operation, and Power over Ethernet supports scalable, manageable power delivery and simplifies IP telephony deployments. As wireless networking emerged, Power over Ethernet began powering wireless devices in locations where local power access did not exist.

For more information on Power over Ethernet, see [Chapter 15, “Configuring Power over Ethernet.”](#)

Secure Shell

Secure Shell (SSH) is a program that enables you to log into another computer over a network, to execute commands remotely, and to move files from one machine to another. The switch may not initiate SSH connections: SSH will be limited to providing a remote login session to the switch and will only function as a server.

Simple Network Management Protocol

Simple Network Management Protocol (SNMP) facilitates the exchange of management information between network devices. The Catalyst 4500 series switch supports these SNMP types and enhancements:

- SNMP—A full Internet standard
- SNMP v2—Community-based administrative framework for version 2 of SNMP
- SNMP v3—Security framework with three levels: noAuthNoPriv, authNoPriv, and authPriv (available only on a crypto image, such as cat4000-i5k91s-mz)

- SNMP trap message enhancements—Additional information with certain SNMP trap messages, including spanning-tree topology change notifications and configuration change notifications

For more information on SNMP, see [Chapter 75, “Configuring SNMP.”](#)

Smart Install



Note

The Smart Install feature is deprecated starting with the following Cisco IOS and IOS XE releases:

- Cisco IOS Release 15.2(2)E9 and later, Cisco IOS XE Release 3.6.9E and later
- Cisco IOS Release 15.2(4)E5 and later, Cisco IOS XE Release 3.8.5E and later
- Cisco IOS Release 15.2(6)E2 and later, Cisco IOS XE Release 3.10.2E and later

For more information, see the Cisco Security Advisory:

<https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-20170214-smi>.

Beginning with Cisco IOS XE 3.4.0SG and 15.1(2)SG, Catalyst 4500 series switches supported Smart Install, which is a plug-and-play configuration and image-management feature that provides zero-touch deployment for new switches. You can ship a switch to a location, place it in the network and power it on with no configuration required on the device.

For details on Smart Install, see the URL:

http://www.cisco.com/en/US/docs/switches/lan/smart_install/configuration/guide/smart_install.html

SPAN and RSPAN

Switched Port Analyzer (SPAN) allows you to monitor traffic on any port for analysis by a network analyzer or Remote Monitoring (RMON) probe. You also can do the following:

- Configure ACLs on SPAN sessions.
- Allow incoming traffic on SPAN destination ports to be switched normally.
- Explicitly configure the encapsulation type of packets that are spanned out of a destination port.
- Restrict ingress sniffing depending on whether the packet is unicast, multicast, or broadcast, and depending on whether the packet is valid.
- Mirror packets sent to or from the CPU out of a SPAN destination port for troubleshooting purposes.

For information on SPAN, see [Chapter 69, “Configuring SPAN and RSPAN.”](#)

Remote SPAN (RSPAN) is an extension of SPAN, where source ports and destination ports are distributed across multiple switches, allowing remote monitoring of multiple switches across the network. The traffic for each RSPAN session is carried over a user-specified RSPAN VLAN that is dedicated for that RSPAN session on all participating switches.

For information on RSPAN, see [Chapter 69, “Configuring SPAN and RSPAN.”](#)

Universal Power over Ethernet

The IEEE 802.3 Power over Ethernet (PoE) standard sets the maximum power that can be sourced by data terminal equipment (DTE) at 30W. This power is sourced over two pairs out of the four twisted pairs of conductors in a Class D, or better, cabling as specified in ISO/IEC 11801:1995.

Cisco® Universal Power over Ethernet (UPOE) is a Cisco proprietary technology that extends the IEEE 802.3 PoE standard to provide the capability to source up to 60W of power over standard Ethernet cabling infrastructure (Class D or better).

For more information on UPOE, see the “Configuring Universal PoE” section in [Chapter 15](#), “Configuring Power over Ethernet.”

Web Content Coordination Protocol

**Note**

WCCP version 1 is *not* supported.

Web Content Communication Protocol (WCCP) Version 2 Layer 2 redirection enables Catalyst 4500 series switches to transparently redirect content requests to the directly connected content engines by using a Layer 2 and MAC address rewrite. The WCCPv2 Layer 2 redirection is accelerated in the switching hardware, and is more efficient than Layer 3 redirection using Generic Routing Encapsulation (GRE). The content engines in a cache cluster transparently store frequently accessed content, and then fulfill successive requests for the same content, eliminating repetitive transmissions of identical content from the original content servers. It supports the transparent redirection of HTTP and non-HTTP traffic with ports or dynamic services, such as Web caching, HTTPS caching, File Transfer Protocol (FTP) caching, proxy caching, media caching, and streaming services. WCCPv2 Layer 2 redirection is typically deployed for transparent caching at network edge, such as regional or branch sites. WCCPv2 Layer 2 redirection cannot be enabled on the same input interface with PBR or VRF-lite. ACL-based classification for Layer 2 redirection is not supported.

For information on WCCP, see [Chapter 83](#), “Configuring WCCP Version 2 Services.”

Wireshark

**Note**

Wireshark is supported only on Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, 7-E, and Catalyst 4500X.

Starting with Cisco IOS Release XE 3.3.0SG and the IP Base and Enterprise Services feature sets, the Catalyst 4500 series switch supports Wireshark. This is a packet analyzer program, formerly known as Ethereal that supports multiple protocols and presents information in a graphical and text-based user interface. Wireshark is applied or enabled on an individual interface; global packet capture is not supported.

For information on Wireshark, see [Chapter 71](#), “Configuring Wireshark.”

XML-PI

eXtensible Markup Language Programmatic Interface (XML-PI) Release 1.0 leverages the Network Configuration Protocol (NETCONF). It provides new data models that collect running configurations and **show** command output down to the keyword level without requiring the technologies or external XML-to-command line interface (CLI) gateways. XML-PI allows you to develop XML-based network management applications to control any number of network devices simultaneously.

Refer to the following link for more details:

http://www.cisco.com/en/US/docs/ios/netmgmt/configuration/guide/nm_xmlpi_v1.html

Security Features

The Catalyst 4500 series switch offers network management and control through the CLI or through alternative access methods, such as SNMP. The switch software supports these security features:

- [802.1X Identity-Based Network Security, page 1-35](#)
- [Dynamic ARP Inspection, page 1-37](#)
- [Cisco TrustSec Security Architecture, page 1-36](#)
- [Cisco TrustSec Security Groups, SGTs and SGACLs, page 1-37](#)
- [Dynamic ARP Inspection, page 1-37](#)
- [Dynamic Host Configuration Protocol Snooping, page 1-37](#)
- [Flood Blocking, page 1-38](#)
- [Hardware-Based Control Plane Policing, page 1-38](#)
- [IP Source Guard, page 1-38](#)
- [IP Source Guard for Static Hosts, page 1-38](#)
- [IPsec VPN, page 1-40](#)
- [IPv6 First Hop Security, page 1-39](#)
- [Local Authentication, RADIUS, and TACACS+ Authentication, page 1-40](#)
- [Network Admission Control, page 1-40](#)
- [Network Security with ACLs, page 1-41](#)
- [Port Security, page 1-41](#)
- [PPPoE Intermediate Agent, page 1-42](#)
- [Session Aware Networking, page 1-42](#)
- [Storm Control, page 1-42](#)
- [uRPF Strict Mode, page 1-43](#)
- [Utilities, page 1-43](#)
- [Web-based Authentication, page 1-44](#)

802.1X Identity-Based Network Security

This security feature consists of the following:

- 802.1X Authentication for Guest VLANs—Allows you to use VLAN assignment to limit network access for certain users.
- 802.1X Authentication Failed Open Assignment—Allows you to configure a switch to handle the case when a device fails to authenticate itself correctly through 802.1X (for example, not providing the correct password).
- 802.1X Authentication with ACL Assignment—Downloads per-host policies such as ACLs and redirect URLs to the switch from the RADIUS server during 802.1X or MAB authentication of the host.
- 802.1X Authentication with Per-User ACL and Filter-ID ACL—Allows ACL policy enforcement using a third-party AAA server.
- 802.1X Convergence—Provides consistency between the switching business units in 802.1X configuration and implementation.
- 802.1X Protocol—Provides a means for a host that is connected to a switch port to be authenticated before it is given access to the switch services.
- 802.1X RADIUS accounting—Allows you to track the use of network devices.
- 802.1X Supplicant and Authenticator Switches with Network Edge Access Topology (NEAT)—Extends identity to areas outside the wiring closet (such as conference rooms). NEAT is designed for deployment scenarios where a switch acting as 802.1X authenticator to end-hosts (PC or Cisco IP-phones) is placed in an unsecured location (outside wiring closet); the authenticator switch cannot always be trusted.
- 802.1X with Authentication Failed VLAN Assignment—Allows you to provide access for authentication failed users on a per-port basis. Authentication failed users are end hosts that are 802.1X-capable but do not have valid credentials in an authentication server or end hosts that do not give any username and password combination in the authentication pop-up window on the user side.
- 802.1X with Inaccessible Authentication Bypass—Applies when the AAA servers are unreachable or nonresponsive. In this situation, 802.1X user authentication typically fails with the port closed, and the user is denied access. Inaccessible Authentication Bypass provides a configurable alternative on the Catalyst 4500 series switch to grant a critical port network access in a locally specified VLAN.
- 802.1X with Port Security—Allows port security on an 802.1X port in either single- or multiple-host mode. When you enable port security and 802.1X on a port, 802.1X authenticates the port, and port security manages the number of MAC addresses allowed on that port, including that of the client.
- 802.1X with MAC Authentication Bypass—Provides network access to agentless devices without 802.1X supplicant capabilities, such as printers. Upon detecting a new MAC address on a switch port, the Catalyst 4500 series switch will proxy an 802.1X authentication request based on the device's MAC address.
- 802.1X with RADIUS-Provided Session Timeouts—Allows you to 802.1X with Unidirectional Controlled Port—Allows the Wake-on-LAN (WoL) magic packets to reach a workstation attached to an unauthorized 802.1X switch port. Unidirectional Controlled Port is typically used to send operating systems or software updates from a central server to workstations at night.
- 802.1X with Violation Mode—This feature allows you to configure 802.1X security violation behavior as either shutdown, restrict, or replace mode, based on the response to the violation.

- 802.1X with VLAN assignment—This feature allows you to enable non-802.1X-capable hosts to access networks that use 802.1X authentication.
- 802.1X with VLAN user distribution—An alternative to dynamically assigning a VLAN ID or a VLAN name, this feature assign a VLAN Group name. It enables you to distribute users belonging to the same group (and characterized by a common VLAN Group name) across multiple VLANs. Ordinarily, you do this to avoid creating an overly large broadcast domain.
- 802.1X with Voice VLAN—This feature allows you to use 802.1X security on a port while enabling it to be used by both Cisco IP phones and devices with 802.1X supplicant support.
- Multi-Domain Authentication—This feature allows both a data device and a voice device, such as an IP phone (Cisco or non-Cisco), to authenticate on the same switch port, which is divided into a data domain and a voice domain.
- RADIUS Change of Authorization—This feature employs Change of Authorization (CoA) extensions defined in RFC 5176 in a push model to allow for the dynamic reconfiguring of sessions from external authentication, authorization, and accounting (AAA) or policy servers.

For more information on 802.1X identity-based network security, see [Chapter 49, “Configuring 802.1X Port-Based Authentication.”](#)

Cisco TrustSec MACsec Encryption

MACsec (Media Access Control Security) is the IEEE 802.1AE standard for authenticating and encrypting packets between two MACsec-capable devices. The Catalyst 4500 series switch supports 802.1AE encryption with MACsec Key Agreement (MKA) on downlink ports for encryption between the switch and host devices. The switch also supports MACsec link layer switch-to-switch security by using Cisco TrustSec Network Device Admission Control (NDAC) and the Security Association Protocol (SAP) key exchange. Link layer security can include both packet authentication between switches and MACsec encryption between switches (encryption is optional).

For more information on TrustSec MACsec encryption, see [Chapter 48, “Configuring MACsec Encryption.”](#)

Cisco TrustSec Security Architecture

The Cisco TrustSec security architecture builds secure networks by establishing domains of trusted network devices. Each device in the domain is authenticated by its peers. Communication on the links between devices in the domain is secured with a combination of encryption, message integrity check, and data-path replay protection mechanisms. Cisco TrustSec uses the device and user credentials acquired during authentication for classifying the packets by security groups (SGs) as they enter the network. This packet classification is maintained by tagging packets on ingress to the Cisco TrustSec network so that they can be properly identified for the purpose of applying security and other policy criteria along the data path. The tag, called the security group tag (SGT), allows the network to enforce the access control policy by enabling the endpoint device to act upon the SGT to filter traffic.

For more information, refer to the following URL:

<http://www.cisco.com/en/US/docs/switches/lan/trustsec/configuration/guide/trustsec.html>

Cisco TrustSec Security Groups, SGTs and SGACLs

**Note**

This support is provided only on Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, and 7-E, and Catalyst 4500X.

A security group is a grouping of users, endpoint devices, and resources that share access control policies. Security groups are defined by the administrator in the Cisco ISE or Cisco Secure ACS. As new users and devices are added to the Cisco TrustSec domain, the authentication server assigns these new entities to appropriate security groups. Once a device is authenticated, Cisco TrustSec tags any packet that originates from that device with a security group tag (SGT) that contains the security group number of the device. The packet carries this SGT throughout the network.

Using security group access control lists (SGACLs), you can control the operations that users can perform based on the security group assignments of users and destination resources.

For more information, refer to the following URL:

<http://www.cisco.com/en/US/docs/switches/lan/trustsec/configuration/guide/trustsec.html>

For Cisco TrustSec SGFT and SGACL guidelines and restrictions that apply on the Catalyst 4500 series switch, refer to "Appendix B. Notes for the Catalyst 4500 Series Switches" in the *Cisco TrustSec Switch Configuration Guide*.

Dynamic ARP Inspection

Dynamic ARP Inspection (DAI) intercepts all ARP requests, replies on untrusted ports, and verifies each intercepted packet for valid IP to MAC bindings. Dynamic ARP Inspection helps to prevent attacks on a network by not relaying invalid ARP replies out to other ports in the same VLAN. Denied ARP packets are logged by the switch for auditing.

For more information on dynamic ARP inspection, see [Chapter 58, "Configuring Dynamic ARP Inspection."](#)

Dynamic Host Configuration Protocol Snooping

Dynamic Host Configuration Protocol (DHCP) Snooping is a security feature that is a component of a DHCP server. DHCP snooping provides security by intercepting untrusted DHCP messages and by building and maintaining a DHCP snooping binding table. An untrusted message is a message that is received from outside the network or firewall that can cause traffic attacks within your network.

DHCP snooping acts like a firewall between untrusted hosts and DHCP servers. It also provides a way to differentiate between untrusted interfaces connected to the end-user and trusted interfaces connected to the DHCP server or another switch.

With SSO support, DHCP Snooping propagates the DHCP-snooped data from the active supervisor engine to the redundant supervisor engine so that when a switchover occurs, the newly active supervisor engine is aware of the DHCP data that was already snooped, and the security benefits continue uninterrupted.

For DHCP server configuration information, refer to the chapter, "Configuring DHCP," in the *Cisco IOS IP and IP Routing Configuration Guide* at the following URL:

http://www.cisco.com/en/US/docs/ios/ipaddr/configuration/guide/iad_dhcp_rdmf_ps6350_TSD_Products_Configuration_Guide_Chapter.html

For information on configuring DHCP snooping, see [Chapter 60, “Configuring DHCP Snooping, IP Source Guard, and IPSG for Static Hosts.”](#)

Flood Blocking

Flood blocking enables users to disable the flooding of unicast and multicast packets on a per-port basis. Occasionally, unknown unicast or multicast traffic from an unprotected port is flooded to a protected port because a MAC address has timed out or has not been learned by the switch.

For information on flood blocking, see [Chapter 67, “Port Unicast and Multicast Flood Blocking.”](#)

Hardware-Based Control Plane Policing

Control Plane Policing provides a unified solution to limit the rate of CPU bound control plane traffic in hardware. It enables users to install system wide control plane ACLs to protect the CPU by limiting rates or filtering out malicious DoS attacks. Control plane policing ensures the network stability, availability and packet forwarding, and prevents network outages such as loss of protocol updates despite an attack or heavy load on the switch. Hardware-based control plane policing is available for all Catalyst 4500 supervisor engines. It supports various Layer 2 and Layer 3 control protocols, such as CDP, EAPOL, STP, DTP, VTP, ICMP, CGMP, IGMP, DHCP, RIPv2, OSPF, PIM, TELNET, SNMP, HTTP, and packets destined to 224.0.0.* multicast link local addresses. Predefined system policies or user-configurable policies can be applied to those control protocols.

Through Layer 2 Control Packet QoS, you can police control packets arriving on a physical port or VLAN; it enables you to apply QoS on Layer 2 control packets

For information on control plane policing and Layer 2 control packet QoS, see [Chapter 57, “Configuring Control Plane Policing and Layer 2 Control Packet QoS.”](#)

IP Source Guard

Similar to DHCP snooping, this feature is enabled on an untrusted Layer 2 port that is configured for DHCP snooping. Initially all IP traffic on the port is blocked except for the DHCP packets, which are captured by the DHCP snooping process. When a client receives a valid IP address from the DHCP server, a PVACL is installed on the port, which restricts the client IP traffic only to clients with assigned IP addresses, so any IP traffic with source IP addresses other than those assigned by the DHCP server will be filtered out. This filtering prevents a malicious host from attacking a network by hijacking neighbor host's IP address.

For information on configuring IP Source Guard, see [Chapter 60, “Configuring DHCP Snooping, IP Source Guard, and IPSG for Static Hosts.”](#)

IP Source Guard for Static Hosts

This feature allows you to secure the IP address learned from static hosts by using ARP packets and then bind that IP address to a given MAC address using the device tracking database, allowing entries to survive through link down events.

IP Source Guard (IPSG) for static hosts allows multiple bindings per-port per-MAC address for both DHCP and static hosts, in both device tracking database and DHCP snooping binding database. The feature allows you to take action when a limit is exceeded.

For information on configuring IPSG for static hosts, see [Chapter 60, “Configuring DHCP Snooping, IP Source Guard, and IPSG for Static Hosts.”](#)

IPv6 First Hop Security

**Note**

IPv6 First Hop Security is supported only on Catalyst 4500-X, Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, 7-E, 6L-E, and 6-E.

IPv6 FHS is a suite of features designed to secure link operations in an IPv6 enabled network as well as address certain scalability issues seen in large L2 domains. IPv6 FHS provides effective counter measures for the following types of attacks or misconfiguration errors that could result in DoS or information theft:

- Router impersonation (MiM attacks)
- Address theft
- Address spoofing
- Remote address resolution cache exhaustion (DoS attacks)

These attacks can come from malicious or mis-configured users and could result in severe disruption to users of the Layer 2 domain and to the network in general.

The following features are supported:

- DAD Proxy
- Data Glean
- Destination Guard
- IPv6 Snooping (DHCP Data Gleaning, per-limit Address Limit)
- IPv6 Address Glean
- IPv6 Device Tracking
- Lightweight DHCPv6 Relay Agent (LDRA)
- NDP Inspection
- Per ND Cache Limit
- Per Port Address Limit
- Source and Prefix Guard

**Note**

IPv6 LDRA is the only FHS feature supported on EtherChannels.

**Note**

Configuring IPv6 FHS on secondary VLANs is not allowed; they inherit the policy from the primary VLAN configuration. Whatever policy is applied on the primary VLANs is programmed automatically on the associated secondary VLANs. The applied policy, however, always overrides the VLAN level configuration.

The following caveats are specific for Data Glean, Prefix Guard, and Source Guard enabled on a Catalyst 4500 series switch:

- First Hop Security (FHS) cannot be configured on the same port or VLAN as dot1X, because the latter asserts control over the MAC table and FHS requires similar control to allow only valid NDP or DHCPv6 hosts.
- If unicast Rpf (unicast reverse path forwarding; uRPF) is configured on box and FHS is enabled, Forward Lookup CAM is populated with routes from FHS and uRPF. Packets that normally fail the uRPF check are admitted provided it passes the Source Guard or Prefix Guard check.
- If Data Glean policy and Source Guard (or Prefix Guard) are applied such that VLAN policies and port policies differ, neither VLAN nor port policy are effective.
- All ICMP and DHCP version 6 control packets are permitted even when Source Guard or Prefix Guard is enabled.

For a brief overview of FHS, see the URL:

http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6553/aag_c45-707354.pdf

For detailed information on how to implement FHS, see the URL:

<http://www.cisco.com/en/US/docs/ios-xml/ios/ipv6/configuration/12-4t/ip6-first-hop-security.html>

IPsec VPN

When a growing organization expands to multiple locations, one of the challenges it faces is how to interconnect remote sites to the corporate network. As network security risks increase and regulatory compliance becomes essential, it is important to address these critical needs.

You can dramatically increase the reach of your network without significantly expanding your infrastructure by using Cisco IOS IPsec VPNs. IPsec is a standards-based encryption technology that enables you to securely connect branch offices and remote users and provides significant cost savings compared to traditional WAN access such as Frame Relay or ATM. IPsec VPNs provide high levels of security through encryption and authentication, protecting data from unauthorized access.

For additional information, refer to the following URL:

http://www.cisco.com/en/US/products/ps6635/products_ios_protocol_group_home.html

Local Authentication, RADIUS, and TACACS+ Authentication

Local Authentication, Remote Authentication Dial-In User Service (RADIUS), and Terminal Access Controller Access Control System Plus (TACACS+) authentication methods control access to the switch. For additional information, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/sec_user_services/configuration/guide/sec_cfg_authntifcn_ps6350_TSD_Products_Configuration_Guide_Chapter.html

Network Admission Control

Network Admission Control consists of two features:

- NAC Layer 2 IP validation

NAC Layer 2 IP is an integral part of Cisco Network Admission Control. It offers the first line of defense for infected hosts (PCs and other devices attached to a LAN port) attempting to connect to the corporate network. NAC Layer 2 IP on the Catalyst 4500 series switch performs posture validation at the Layer 2 edge of the network for non-802.1x-enabled host devices. Host device

posture validation includes antivirus state and OS patch levels. Depending on the corporate access policy and host device posture, a host may be unconditionally admitted, admitted with restricted access, or quarantined to prevent the spread of viruses across the network.

For more information on Layer 2 IP validation, see the URL:

http://www.cisco.com/en/US/docs/net_mgmt/cisco_secure_access_control_server_for_windows/4.1/configuration/guide/nac_conf.html

- NAC Layer 2 802.1X authentication

The Catalyst 4500 series switch extends NAC support to 802.1x-enabled devices. Like NAC Layer 2 IP, the NAC Layer 2 802.1x feature determines the level of network access based on endpoint information.

For more information on 802.1X identity-based network security, see [Chapter 49, “Configuring 802.1X Port-Based Authentication.”](#)

Network Security with ACLs

An access control list (ACL) filters network traffic by controlling whether routed packets are forwarded or blocked at the router interfaces. The Catalyst 4500 series switch examines each packet to determine whether to forward or drop the packet based on the criteria you specified within the access lists.

MAC access control lists (MACLs) and VLAN access control lists (VACLs) are supported. VACLs are also known as VLAN maps in Cisco IOS.

The Catalyst 4500 series switch supports three types of ACLs:

- IP ACLs, which filter IP traffic, including TCP, the User Datagram Protocol (UDP), Internet Group Management Protocol (IGMP), and Internet Control Message Protocol (ICMP)
- IPv6 ACLs
- MAC ACLs which match based on Ethernet addresses and Ether Type

The switch supports the following applications of ACLs to filter traffic:

- MAC address filtering, which enables you to block unicast traffic for a MAC address on a VLAN interface.
- Port ACLs, which enable you to apply ACLs to Layer 2 interfaces on a switch for inbound traffic.
- Router ACLs, which are applied to Layer 3 interfaces to control the access of routed traffic between VLANs.
- VLAN ACLs or VLAN maps to control the access of all packets (bridged and routed).

For information on ACLs, MACLs, VLAN maps, MAC address filtering, and Port ACLs, see [Chapter 62, “Configuring Network Security with ACLs.”](#)

Port Security

Port security restricts traffic on a port based upon the MAC address of the workstation that accesses the port. Trunk port security extends this feature to trunks, including private VLAN isolated trunks, on a per-VLAN basis.

Sticky port security extends port security by saving the dynamically learned MAC addresses in the running configuration to survive port link down and switch reset. It enables a network administrator to restrict the MAC addresses allowed or the maximum number of MAC addresses on each port.

Voice VLAN sticky port security further extends the sticky port security to the voice-over-IP deployment. Voice VLAN sticky port security locks a port and blocks access from a station with a MAC address different from the IP phone and the workstation behind the IP phone.

For information on port security, see [Chapter 55, “Configuring Port Security.”](#)

PPPoE Intermediate Agent

PPPoE Intermediate Agent (PPPoE IA) is placed between a subscriber and BRAS to help the service provider BRAS distinguish between end hosts connected over Ethernet to an access switch. On the access switch, PPPoE IA enables Subscriber Line Identification by appropriately tagging Ethernet frames of different users. (The tag contains specific information such as which subscriber is connected to the switch and VLAN.) PPPoE IA acts as mini-security firewall between host and BRAS by intercepting all PPPoE Active Discovery (PAD) messages on a per-port per-VLAN basis. It provides specific security feature such as verifying the intercepted PAD message from untrusted port, performing per-port PAD message rate limiting, inserting and removing VSA tags into and from PAD messages, respectively.

For information on PPPoE IA, see [Chapter 52, “Configuring the PPPoE Intermediate Agent.”](#)

Session Aware Networking

Session Aware Networking provides an identity-based approach to access management and subscriber management. It offers a consistent way to configure features across technologies, a command interface that allows easy deployment and customization of features, and a robust policy control engine with the ability to apply policies defined locally or received from an external server to enforce policy in the network.

Session Aware Networking allows a single session identifier to be used for web authentication sessions in addition to all 802.1X and MAB authenticated sessions for a client. This session ID is used for all reporting purposes such as show commands, MIBs, and RADIUS messages and allows users to distinguish messages for one session from messages for other sessions. This common session ID is used consistently across all authentication methods and features applied to a session.

**Note**

IPv6 is not supported for web authentication, dot.1X, or MAB.

For additional information, refer to the following URL:

<http://www.cisco.com/en/US/docs/ios-xml/ios/san/configuration/x3e/3850/san-overview.html>

Storm Control

Broadcast suppression is used to prevent LANs from being disrupted by a broadcast storm on one or more switch ports. A LAN broadcast storm occurs when broadcast packets flood the LAN, creating excessive traffic and degrading network performance. Errors in the protocol-stack implementation or in the network configuration can cause a broadcast storm. Multicast and broadcast suppression measures how much broadcast traffic is passing through a port and compares the broadcast traffic with some configurable threshold value within a specific time interval. If the amount of broadcast traffic reaches the threshold during this interval, broadcast frames are dropped, and optionally the port is shut down

Starting with Cisco IOS Release 12.2(40)SG, the Catalyst 4500 series switch allows suppression of broadcast and multicast traffic on a per-port basis.

For information on configuring broadcast suppression, see [Chapter 68, “Configuring Storm Control.”](#)

uRPF Strict Mode

The uRPF feature mitigates problems caused by the introduction of malformed or forged (spoofed) IP source addresses into a network by discarding IP packets that lack a verifiable IP source address. uRPF deflects denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks by forwarding only packets that have source addresses that are valid and consistent with the IP routing table. This helps to protect the network of the customer, the ISP, and the rest of the Internet. When using uRPF in strict mode, the packet must be received on the interface that the router uses to forward the return packet. uRPF strict mode is supported for both IPv4 and IPv6 prefixes.

For information on configuring broadcast suppression, see [Chapter 38, “Configuring Unicast Reverse Path Forwarding.”](#)

Utilities

Supported utilities include the following:

Layer 2 Traceroute

Layer 2 traceroute allows the switch to identify the physical path that a packet takes from a source device to a destination device. Layer 2 traceroute supports only unicast source and destination MAC addresses.

For information about Layer 2 Traceroute, see [Chapter 10, “Checking Port Status and Connectivity.”](#)

Time Domain Reflectometry

Time Domain Reflectometry (TDR) is a technology used for diagnosing the state and reliability of cables. TDR can detect open, shorted, or terminated cable states. The calculation of the distance to the failure point is also supported.

For information about TDR, see [Chapter 10, “Checking Port Status and Connectivity.”](#)

Debugging Features

The switch has several commands to help you debug your initial setup. These commands are included in the following command groups:

- **platform**
- **debug platform**

For more information, refer to the command reference guide.

Web-based Authentication

The web-based authentication feature, known as Web Authentication Proxy, enables you to authenticate end users on host systems that do not run the IEEE 802.1X supplicant. When you initiate an HTTP session, this feature intercepts ingress HTTP packets from the host and sends an HTML login page to your. You key in the credentials, which the web-based authentication feature sends to the AAA server for authentication. If authentication succeeds, web-based authentication sends a Login-Successful HTML page to the host and applies the access policies returned by the AAA server.

For information on configuring web-based authentication, see [Chapter 53, “Configuring Web-Based Authentication.”](#)



Command-Line Interfaces

This chapter describes the CLIs you use to configure the Catalyst 4500 series switch. This chapter includes the following major sections:

Comment - to be deleted.

- [Accessing the Switch CLI, page 2-2](#)
- [Performing Command-Line Processing, page 2-3](#)
- [Performing History Substitution, page 2-4](#)
- [About Cisco IOS Command Modes, page 2-4](#)
- [Getting a List of Commands and Syntax, page 2-5](#)
- [ROMMON Command-Line Interface, page 2-7](#)
- [Archiving Crashfiles Information, page 2-8](#)
- [Displaying a Crash Dump for Supervisor Engine 6-E and 6L-E, page 2-8](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the *Cisco Catalyst 4500 Series Switch Command Reference* and related publications at this location:

<http://www.cisco.com/en/US/products/hw/switches/ps4324/index.html>

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the Cisco IOS library. See related publications at this location:

<http://www.cisco.com/en/US/products/ps6350/index.html>

The following command changes apply only to Supervisor Engines 6-E and 6L-E:

- The **rename** command is supported in the FAT file system for bootflash and slot0.
- The **fsck** command is supported for the slot0 device. It is not supported in the file systems on supervisor engines other than Supervisor Engine 6-E and 6L-E.

The following additional file management commands are supported on Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, and 7-E:

- **verify** <filename>
- **delete** <filename>
- **copy** <source_filename>, target_filename>

Accessing the Switch CLI

The following sections describe how to access the switch CLI:

- [Accessing the CLI Using the EIA/TIA-232 Console Interface, page 2-2](#)
- [Accessing the CLI Through Telnet, page 2-2](#)

Accessing the CLI Using the EIA/TIA-232 Console Interface



Note

EIA/TIA-232 was known as recommended standard 232 (RS-232) before its acceptance as a standard by the Electronic Industries Alliance (EIA) and Telecommunications Industry Association (TIA).

Perform the initial switch configuration over a connection to the EIA/TIA-232 console interface. Refer to the *Catalyst 4500 Series Switch Module Installation Guide* for console interface cable connection procedures.

To access the switch through the console interface, perform this task:

	Command	Purpose
Step 1	Switch> enable	From the user EXEC prompt (>), enter enable to change to enable mode (also known as privileged mode or privileged EXEC mode).
Step 2	Password: <i>password</i> Switch#	At the password prompt, enter the system password. The prompt (#) appears, indicating that you have accessed the CLI in enabled mode.
Step 3	Switch# quit	When you are finished executing the task command, exit the session.

After accessing the switch through the EIA/TIA-232 interface, you see this display:

```
Press Return for Console prompt

Switch> enable
Password:< >
Switch#
```

Accessing the CLI Through Telnet



Note

Before you make a Telnet connection to the switch, you must set the IP address for the switch. See the [“Configuring Physical Layer 3 Interfaces”](#) section on page 36-13.

The switch supports up to eight simultaneous Telnet sessions. Telnet sessions disconnect automatically after remaining idle for the period specified by the **exec-timeout** command.

To make a Telnet connection to the switch, perform this task:

	Command	Purpose
Step 1	<code>telnet {hostname ip_addr}</code>	From the remote host, enter the telnet command and the name or IP address of the switch you want to access.
Step 2	Password: <code>password</code> Switch#	At the prompt, enter the password for the CLI. If no password has been configured, press Return .
Step 3		Enter the necessary commands to complete your desired tasks.
Step 4	Switch# <code>quit</code>	When finished, exit the Telnet session.

This example shows how to open a Telnet session to the switch:

```

unix_host% telnet Switch_1
Trying 172.20.52.40...
Connected to 172.20.52.40.
Escape character is '^]'.
User Access Verification
Password:< >
Switch_1> enable
Password:
Switch_1#

```

Performing Command-Line Processing

Switch commands are not case sensitive. You can abbreviate commands and parameters if the abbreviations contain enough letters to be different from any other currently available commands or parameters.

You can scroll through the last 20 commands stored in the history buffer and enter or edit a command at the prompt. [Table 2-1](#) lists the keyboard shortcuts for entering and editing switch commands.

Table 2-1 Keyboard Shortcuts

Keystrokes	Result
Press Ctrl-B or press the Left Arrow key ¹	Moves the cursor back one character.
Press Ctrl-F or press the Right Arrow key ¹	Moves the cursor forward one character.
Press Ctrl-A	Moves the cursor to the beginning of the command line.
Press Ctrl-E	Moves the cursor to the end of the command line.
Press Esc-B	Moves the cursor back one word.
Press Esc-F	Moves the cursor forward one word.

1. The Arrow keys function only on ANSI-compatible terminals, such as VT100s.

Performing History Substitution

The history buffer stores the last 20 command lines you entered. History substitution enables you to access these command lines without retyping them. Table 2-2 lists the history substitution commands.

Table 2-2 History Substitution Commands

Command	Purpose
Ctrl-P or the Up Arrow key ¹	Recalls commands in the history buffer, beginning with the most recent command. Repeat the key sequence to recall older commands successively.
Ctrl-N or the Down Arrow key ¹	Returns to more recent commands in the history buffer after commands have been recalled with Ctrl-P or the Up Arrow key. Repeat the key sequence to recall more recent commands.
Switch# show history	Lists the last several commands you have entered in EXEC mode.

1. The Arrow keys function only on ANSI-compatible terminals such as VT100s.

About Cisco IOS Command Modes



Note

For complete information about Cisco IOS command modes, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide* and the *Cisco IOS Configuration Fundamentals Command Reference* at the following URLs:

http://www.cisco.com/en/US/docs/ios/12_2/configfun/configuration/guide/ffun_c.html

http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_book.html

The Cisco IOS user interface has many different modes: user EXEC, privileged EXEC (enable), global configuration, interface, subinterface, and protocol-specific. The commands available to you depend on which mode you are in. To get a list of the commands in a given mode, enter a question mark (?) at the system prompt. See the “Getting a List of Commands and Syntax” section on page 2-5 for more information.

When you start a session on the switch, you begin in user mode, also called user EXEC mode. Only a small subset of commands are available in EXEC mode. To have access to all commands, you must enter privileged EXEC mode, also called enable mode. To access the privileged EXEC mode, you must enter a password. When you are in the privileged EXEC mode, you can enter any EXEC command or access global configuration mode. Most EXEC commands are one-time commands, such as **show** commands, which display the current configuration status, and **clear** commands, which reset counters or interfaces. The EXEC commands are not saved when the switch is rebooted.

The configuration modes allow you to make changes to the running configuration. If you save the configuration, these commands are stored when you reboot the switch. You must start in global configuration mode. From global configuration mode, you can enter interface configuration mode, subinterface configuration mode, and a variety of protocol-specific modes.

You use a separate mode called ROMMON when the switch cannot boot up properly. For example, the switch might enter ROMMON mode if it does not find a valid system image when it is booting, or if its configuration file is corrupted. For more information, see the [“ROMMON Command-Line Interface” section on page 2-7](#).

Table 2-3 lists and describes frequently used Cisco IOS modes.

Table 2-3 Frequently Used Cisco IOS Command Modes

Mode	What You Use It For	How to Access	Prompt
User EXEC	To connect to remote devices, change terminal settings on a temporary basis, perform basic tests, and display system information.	Log in.	Switch>
Privileged EXEC (enable)	To set operating parameters. The privileged command set includes the commands in user EXEC mode, as well as the configure command. Use the configure command to access the other command modes.	From user EXEC mode, enter the enable command and the enable password (if a password has been configured).	Switch#
Global configuration	To configure features that affect the system as a whole, such as the system time or switch name.	From privileged EXEC mode, enter the configure terminal command.	Switch(config)#
Interface configuration	To enable or modify the operation of a 10-Gigabit Ethernet, Gigabit Ethernet, or Fast Ethernet interface with interface commands.	From global configuration mode, enter the interface <i>type location</i> command.	Switch(config-if)#
Console configuration	To configure the console interface; from the directly connected console or the virtual terminal; used with Telnet.	From global configuration mode, enter the line console 0 command.	Switch(config-line)#

The Cisco IOS command interpreter, called the EXEC, interprets and runs the commands you enter. You can abbreviate commands and keywords by entering just enough characters to make the command unique from other commands. For example, you can abbreviate the **show** command to **sh** and the **configure terminal** command to **confi t**.

When you type **exit**, the switch backs out one level. To exit configuration mode completely and return to privileged EXEC mode, press **Ctrl-Z**.

Getting a List of Commands and Syntax

In any command mode, you can get a list of available commands by entering a question mark (?).

```
Switch> ?
```

To obtain a list of commands that begin with a particular character sequence, enter those characters followed by the question mark (?). Do not include a space before the question mark. This form of help is called word help, because it completes a word for you.

To list keywords or arguments, enter a question mark in place of a keyword or argument. Include a space before the question mark. This form of help is called command syntax help, because it reminds you which keywords or arguments are applicable based on the command, keywords, and arguments you have already entered.

```
Switch# configure ?
memory          Configure from NV memory
network         Configure from a TFTP network host
overwrite-network Overwrite NV memory from TFTP network host
terminal        Configure from the terminal
<cr>
```

To redisplay a command you previously entered, press the **Up Arrow** key or **Ctrl-P**. You can continue to press the **Up Arrow** key to see the last 20 commands you entered.

**Tip**

If you are having trouble entering a command, check the system prompt and enter the question mark (?) for a list of available commands. You might be in the wrong command mode or using incorrect syntax.

Type **exit** to return to the previous mode. Press **Ctrl-Z** or enter the **end** command in any mode to immediately return to privileged EXEC mode.

Virtual Console for Standby Supervisor Engine

Catalyst 4500 series switches can be configured with 2 supervisor engines to provide redundancy. When the switch is powered, one of the supervisor engines becomes active and remains active until a switchover occurs. The other supervisor engine remains in standby mode.

Each supervisor engine has its own console port. Access to the standby supervisor engine is possible only through the console port of the standby supervisor engine. You must connect to the standby console to access, monitor or debug the standby supervisor.

Virtual Console for Standby Supervisor Engine enables you to access the standby console from the active supervisor engine without requiring a physical connection to the standby console. It uses IPC over EOBC to communicate with the standby supervisor engine and thus emulate the standby console on the active supervisor engine. Only one active standby console session is active at any time.

The virtual console for standby supervisor engine enables users who are logged onto the active supervisor engine to remotely execute **show** commands on the standby supervisor engine and view the results on the active supervisor engine. Virtual console is available only from the active supervisor engine.

You can access the standby virtual console from the active supervisor engine with the **attach module**, **session module**, or **remote login** commands on the active supervisor engine. You must be in privilege EXEC mode (level 15) to run these commands to access the standby console.

Once you enter the standby virtual console, the terminal prompt automatically changes to *hostname-standby-console#*, where *hostname* is the configured name of the switch. The prompt is restored back to the original prompt when you exit the virtual console.

You exit the virtual console with the **exit** or **quit** commands. When the inactivity period of the terminal on the active supervisor engine where you logged in exceeds the configured idle time, you are automatically logged out of the terminal on the active supervisor engine. In this case, the virtual console session is also terminated. Virtual console session is also automatically terminated when the standby is rebooted. After the standby boots up, you need to create another virtual console session.

To log in to the standby supervisor engine using a virtual console, enter the following command:

```
Switch# session module 2
Connecting to standby virtual console
Type "exit" or "quit" to end this session
Switch-standby-console# exit
```

If the standby console is not enabled, the following message appears:

```
Switch-standby-console#
Standby console disabled.
Valid commands are: exit, logout
```

Virtual session into the standby console is N/A with RPR:

```
Switch# session module 2
IPC server port name IFConsoleServer:2 not registered on standby.
Secondary cannot be accessed by virtual console
```

**Note**

The standby virtual console provides the standard features that are available from the supervisor console such as command history, command completion, command help and partial command keywords.

The following limitations apply to the standby virtual console:

- All commands on the virtual console run to completion. It does not provide the auto-more feature; it behaves as if the **terminal length 0** command has been executed. It is also noninteractive. A executing command cannot be interrupted or aborted by any key sequence on the active supervisor engine. If a command produces considerable output, the virtual console displays it on the supervisor screen.
- The virtual console is noninteractive. Because the virtual console does not detect the interactive nature of a command, any command that requires user interaction causes the virtual console to wait until the RPC timer aborts the command.
- The virtual console timer is set to 60 seconds. The virtual console returns to its prompt after 60 seconds. During this time, you cannot abort the command from the keyboard. You must wait for the timer to expire before you continue.
- You cannot use virtual console to view debug and syslog messages that are being displayed on the standby supervisor engine. The virtual console only displays the output of commands that are executed from the virtual console. Other information that is displayed on the real standby console does not appear on the virtual console.

ROMMON Command-Line Interface

ROMMON is a ROM-based program that is involved at power-up or reset, or when a fatal exception error occurs. The switch enters ROMMON mode if the switch does not find a valid software image, if the NVRAM configuration is corrupted, or if the configuration register is set to enter ROMMON mode. From the ROMMON mode, you can load a software image manually from flash memory, from a network server file, or from bootflash.

You can also enter ROMMON mode by restarting the switch and pressing **Ctrl-C** during the first five seconds of startup.

**Note**

Ctrl-C is always enabled for 60 seconds after you reboot the switch, even if **Ctrl-C** is configured to be off in the configuration register settings.

When you enter ROMMON mode, the prompt changes to **rommon 1>**. Use the **?** command to see the available ROMMON commands.

For more information about the ROMMON commands, refer to the command reference guide.

Archiving Crashfiles Information

This feature allows you to archive crashinfo files (otherwise overwritten if another system reset were to happen first to the bootflash). Having access to archived crashinfo data greatly assists in troubleshooting.

To archive crashinfo files, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# exception crashinfo file bootflash: name	Enables archiving crashinfo files to bootflash. The files are stored in bootflash with the <i>name</i> specified concatenated with the date.
Step 3	Switch(config)# end	Returns to privileged EXEC mode.
Step 4	Switch# show running-config	Verifies your entries.
Step 5	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Displaying a Crash Dump for Supervisor Engine 6-E and 6L-E

A crash dump provides the following information:

- Malloc or free traces
- Chuck alloc/free traces
- Process block dump
- Register memory dump
- Current proc stack partial decode
- Interrupt level stack
- Last 128 memory block dump

To display a crash dump, do the following:

```
Switch# show platform crashdump
```

```
Current Time: 9/6/2010 15:47:21
```

```
Last Power Failure: 09/06/2010 15:03:28
```

```
Last Reload Status: 00002000
```

```
Last Software Reset State: 00000000
```

```
Crashdump version: 1
```

```
Last crash: 09/06/2010 06:21:58
```

```
Build: 12.2(20100723:074204) ENT SERVICES
```

```
buildversion addr: 14847D24
```

```

===== Context =====
pc=10999E70 lr=10999E34 msr=02029230 vector=00000600
cr=20004022 ctr=108EC3EC xer=00000000
r0=10999E34 r1=2421F930 r2=0000001E r3=234BBFD8
r4=0000000A r5=00000000 r6=2421F918 r7=00000000
r8=00000000 r9=00000000 r10=14850000 r11=234BBFD4
r12=EB93A100 r13=B4E9F3F3 r14=10CD0984 r15=00000000
r16=156CA504 r17=156CA504 r18=00000001 r19=00000000
r20=00000000 r21=00000000 r22=00000000 r23=00000000
r24=00000000 r25=00000000 r26=00000000 r27=00000000
r28=15870804 r29=00000000 r30=14850000 r31=00000000
dec=00083695 tbu=00000002 tbl=2A7D42DA
dar=80210020 dsisr=80210020 hid0=80004000

```

```

Traceback: 10999E70 11B430B8 10C84444 10C83338 11BE0C5C 10C93874 10C93D78 10C94140
10C992EC 10CD155C 1099BCFC 10992CEC

```

```

===== Stack frames =====
Frame 1: pc=11B430B8 stack=2421F940
Frame 2: pc=10C84444 stack=2421F948
Frame 3: pc=10C83338 stack=2421F9B0
Frame 4: pc=11BE0C5C stack=2421F9E8
Frame 5: pc=10C93874 stack=2421FA00
Frame 6: pc=10C93D78 stack=2421FA18
Frame 7: pc=10C94140 stack=2421FA48
Frame 8: pc=10C992EC stack=2421FA58
Frame 9: pc=10CD155C stack=2421FA70
Frame 10: pc=1099BCFC stack=2421FB08
Frame 11: pc=10992CEC stack=2421FB10

```

```

===== Pushed stack =====
2421F930: 2421F940 10999E34 2421F940 15868B74
2421F940: 2421F948 11B430B8 2421F9B0 10C84444
2421F950: 2421F978 00000000 00000000 00000000
2421F960: 00000000 2421F9C0 00000000 240CC3C8
2421F970: 2421F990 11AE7394 00000006 FFFFFFFF
2421F980: 00000000 00000000 00000000 14BE0000
2421F990: 00000000 00000000 00000000 00000000
2421F9A0: 00000001 00000000 15868B74 15868B74
2421F9B0: 2421F9E8 10C83338 00000000 00000000
2421F9C0: 00000071 15868B74 156CA328 13794ACD
2421F9D0: 00000000 00000001 00000000 1511A790
2421F9E0: 2366B680 15868B74 2421FA00 11BE0C5C
2421F9F0: 156CA328 156CA328 2366B680 15868B74
2421FA00: 2421FA18 10C93874 2421FA20 00000000
2421FA10: 00000000 2366B628 2421FA48 10C93D78
2421FA20: 2421FA58 10C95370 00000000 11BB0A98
2421FA30: 00000000 00000000 15868B74 00000000
2421FA40: 00000000 15868B74 2421FA58 10C94140
2421FA50: 00000003 15868B74 2421FA70 10C992EC
2421FA60: 00000000 00000000 00000000 156CA328
2421FA70: 2421FB08 10CD155C 0DFFFFFF FFFFFFFF
2421FA80: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
2421FA90: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
2421FAA0: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
2421FAB0: 00000001 FFFFFFFF FFFFFFFF FFFFFFFF
2421FAC0: FFFFFFFF 00000000 00000000 00000000
2421FAD0: 00000000 00000000 00000000 00000000
2421FAE0: 00000000 00000000 00000000 00000000

```

```

2421FAF0: 00000000 00000000 00000000 00000000
2421FB00: 00000000 00000000 2421FB10 1099BCFC
2421FB10: 00000000 10992CEC FFFFFFFF

```

```

===== Popped stack =====

```

```

2421F730: E8000800 151B1AB0 2421F748 132BBFA8
2421F740: 000E8000 151B1AB0 2421F760 132BC0D0
2421F750: 000E8000 00009B0A E8000800 151B1AB0
2421F760: 2421F778 132BC2A0 E8000800 00009B0A
2421F770: 00000800 153B1B7C 2421F790 123FAF28
2421F780: 2421F790 00000000 0000000A 151B17E4
2421F790: 2421F798 123FB2BC 2421F7B0 11C12A90
2421F7A0: 00009B0A 11C12880 0000000A 146C0000
2421F7B0: 2421F7C0 11BA7384 00000000 146B0000
2421F7C0: 2421F7D0 11AD3144 0000000A 0000000A
2421F7D0: 2421F7D8 11C10390 2421F7E0 11BB0424
2421F7E0: 2421F7F0 11BB04E4 2433FCD4 FFFFFFFE
2421F7F0: 2421F800 107CF880 7FFFFFFF FFFFFFFE
2421F800: 2421F8A8 107CCDF0 20637261 73686475
2421F810: 6D700000 00000000 2421F840 00000000
2421F820: 2421F8B0 00000000 0000004A 002E8A00
2421F830: 39760000 0000004A 00000000 2433FCF0
2421F840: 2421F848 2433FCF0 00000000 11A12ACC
2421F850: 13CD617C 10C7DAAC 00000000 2421F8AC
2421F860: 10CD0984 00000000 156CA504 156CA504
2421F870: 00000001 00000000 00000000 00000000
2421F880: 00000000 00000000 00000000 00000000
2421F890: 00000000 00000000 15870804 00000000
2421F8A0: 14850000 FFFFFFFE 2421F930 107CFC7C
2421F8B0: 2421F8C8 14BB1760 00000002 00000000
2421F8C0: 2421F930 14620E40 24330AB4 0000004A
2421F8D0: 00000000 00000000 2421F8E8 10C1FD9C
2421F8E0: 2421F8F8 00000000 00000000 00000000
2421F8F0: 15868B74 15868B74 2421F910 117CF5C0
2421F900: 2421F968 1586A45C 2421F920 15868B74
2421F910: 2421F918 00000000 14850000 00000000
2421F920: 2421F930 10999978 2421F930 00000000

```

```

===== Malloc and Free Traces=====

```

```

MallocFree Trace: ixmmlallocfree=0x2C ptr=0x151A40D8
151A378: 2366B628 11AF1144 2366B628 11AF1348 2366B66C 60000024 2447A940 11AF1350
151A3F98: 2447A940 30000018 2447A940 11AF1110 2366B628 4000000E 2366B628 11AF1144
151A3FB8: 2366B628 11AF1348 2366B66C 60000024 2447A940 11AF1350 2447A940 30000018
151A3FD8: 2447A940 11AF1110 2366B628 4000000E 2366B628 11AF1144 2366B628 11AF1348
151A3FF8: 2366B66C 60000024 2447A940 11AF1350 2447A940 30000018 2447A940 11AF1110
151A4018: 2366B628 4000000E 2366B628 11AF1144 2366B628 11AF1348 2366B66C 60000024
151A4038: 2447A940 11AF1350 22FAC944 119F6CC0 22FACF4C 6000005E 2433FCD4 40000046
151A4058: 2433FCD4 11A31DD4 2433FCD4 11A32370 2433FD88 6000005E 2447A940 30000018
151A4078: 2447A940 107D7294 2366B628 40000018 2366B628 10C9533C 2366B680 3000001A
151A4098: 2366B680 10C9536C 2433FCD4 4000000E 2433FCD4 10C7DAAC 2433FCD4 10C7DB90
151A40B8: 2433FD18 60000096 2433FCD4 4000000E 2433FCD4 10E28604 2433FCD4 10E287BC
151A40D8: 2433FD18 60000096 2366B66C 60000024 2447A940 11AF1350 2447A940 30000018
151A40F8: 2447A940 11AF1110 2366B628 4000000E 2366B628 11AF1144 2366B628 11AF1348
151A4118: 2366B66C 60000024 2447A940 11AF1350 2447A940 30000018 2447A940 11AF1110
151A4138: 2366B628 4000000E 2366B628 11AF1144 2366B628 11AF1348 2366B66C 60000024
151A4158: 2447A940 11AF1350 2447A940 30000018 2447A940 11AF1110 2366B628 4000000E

```

```

===== Chunk Malloc and Chunk Free Traces=====

```

```

151A3B78: 238928B8 11A32D70 11A34618 238928B8 11A3187C 11A34618
151A3B60: 15866F0C 10C7FF20 10C7F104 1586FBF0 10C7FE38 10C7F17C

```

151A3B48: 1586D760 10C7FE38 10C7F17C 1586FF98 10C7FE38 10C7F17C

```

151A3B30: 1586D760 10C84B24 10C7F17C 1586D760 10C7FE38 10C7F17C
151A3B18: 1586FF98 10C84B24 10C7F17C 1586FF98 10C7FE38 10C7F17C
151A3B00: 1586D760 10C84B24 10C7F17C 1586D760 10C7FE38 10C7F17C
151A3AE8: 1586FF98 10C84B24 10C7F17C 1586FF98 10C7FE38 10C7F17C
151A3AD0: 1586D760 10C84B24 10C7F17C 1586FBF0 10C84B24 10C7F17C
151A3AB8: 1586FBF0 10C7FE38 10C7F17C 1586D760 10C7FE38 10C7F17C
151A3AA0: 1586FBF0 10C84B24 10C7F17C 1586FBF0 10C7FE38 10C7F17C
151A3A88: 15870340 10C7FE38 10C7F17C 1586FBF0 10C84B24 10C7F17C
151A3A70: 1586D760 10C84B24 10C7F17C 1586D760 10C7FE38 10C7F17C
151A3A58: 1586FBF0 10C7FE38 10C7F17C 1586D760 10C84B24 10C7F17C
151A3A40: 1586D760 10C7FE38 10C7F17C 1586FBF0 10C84B24 10C7F17C
151A3A28: 1586FBF0 10C7FE38 10C7F17C 1586D760 10C84B24 10C7F17C
151A3A10: 15870340 10C84B24 10C7F17C 15870340 10C7FE38 10C7F17C
151A39F8: 1586D760 10C7FE38 10C7F17C 15870340 10C84B24 10C7F17C
151A39E0: 15870340 10C7FE38 10C7F17C 1586D760 10C84B24 10C7F17C
151A39C8: 1586D760 10C7FE38 10C7F17C 15870340 10C84B24 10C7F17C
151A39B0: 15870340 10C7FE38 10C7F17C 1586D760 10C84B24 10C7F17C
151A3998: 1586D760 10C7FE38 10C7F17C 15870340 10C84B24 10C7F17C
151A3980: 15870340 10C7FE38 10C7F17C 1586D760 10C84B24 10C7F17C
151A3968: 1586D760 10C7FE38 10C7F17C 15870340 10C84B24 10C7F17C
151A3950: 15870340 10C7FE38 10C7F17C 1586D3B8 10C7FE38 10C7F17C
151A3938: 15870340 10C84B24 10C7F17C 1586D760 10C84B24 10C7F17C
151A3920: 1586D760 10C7FE38 10C7F17C
151A3C14: 15870340 10C7FE38 10C7F17C 1586D760 10C84B24 10C7F17C
151A3BFC: 1586D3B8 10C84B24 10C7F17C 1586D3B8 10C7FE38 10C7F17C
151A3BE4: 1586D760 10C7FE38 10C7F17C 1586D3B8 10C84B24 10C7F17C
151A3BCC: 15870340 10C84B24 10C7F17C 15870340 10C7FE38 10C7F17C
151A3BB4: 1586D3B8 10C7FE38 10C7F17C 15870340 10C84B24 10C7F17C
151A3B9C: 1586D760 10C84B24 10C7F17C 1586D760 10C7FE38 10C7F17C
151A3B84: 15870340 10C7FE38 10C7F17C

```

==== Process Level Info =====

---- Current Process Block (at 0x24330AB4) ----

```

24330A8C: AB1234CD 710000 24330AB4 13DF55F0 11A2F280 24330D48 24330A5C 8000014A
24330AAC: 1 10530DC4 242110BC 1582AAAC 156CA328 10CD0984 0 156CA504
24330ACC: 156CA504 6 FFFFFFFF 1 2421FA78 13D2A3E0 FF 0
24330AEC: 1 13D2A3E0 2421FA78 24330AB4 14BE0000 156CA328 107D4240 40004024
24330B0C: 11A3C6A4 2029230 0 0 0 0 0 10100
24330B2C: 0 1000000 0 0 0 71 0 0
24330B4C: 0 25610 2350 320BC 0 0 0 0
24330B6C: 0 2035F 0 156CA328 0 2210B 0 2210B
24330B8C: 0 13 0 13D42FC4 4 1 15E 1
24330BAC: 0 0 0 EA60 EA60 156CA328 0 0
24330BCC: 0 0 0 0 0 24330AB4 151A5708 0
24330BEC: 0 0 149A2408 0 0 0 420A 0
24330C0C: 0 0 24330BEC 0 0 0 4290 24330AB4
24330C2C: 0 0 0 0 24330BEC 24330AB4 0 0
24330C4C: 142D2 0 0 0 24330BEC 24330AB4 0 0
24330C6C: 242D2 0 0 0 0 0 0 24325EB4
24330C8C: 0 0 0 0 0 0 0 0
24330CAC: 0 24325F14 24330C9C 24325EBC 0 151A6450 0 0
24330CCC: 0 0 0 FFFFFFFF FFFFFFFF 0 0 0
24330CEC: 0 0 0 0 0 23EFC15C 0 0
24330D0C: 32 0 0 0 0 0 0 0
24330D2C: 0 0 0 0 BEEFCAFE 0

```

---- Partial decode of process block ----

Pid 113: Process "Exec" stack 0x242110BC savedsp 0x1582AAAC

■ Displaying a Crash Dump for Supervisor Engine 6-E and 6L-E

```

Flags: analyze crashblock on_old_queue
Status      0x00000000 Orig_ra  0x00000000 Routine    0x00000000 Signal  0
Caller_pc   0x00000000 Callee_pc 0x00000000 Dbg_events 0x00000000 State   0
Totmalloc   153104      Totfree   9040      Totgetbuf   0
Totretbuf   0          Edisms    0x0      Eparm       0x156CA328
Elapsed     0x0        Ncalls    0x13     Ngiveups    0x0
Priority_q   4          Ticks_5s   1       Cpu_5sec    0       Cpu_1min    0
Cpu_5min    0          Stacksize 0xEA60    Lowstack    0xEA60
Ttyptr      0x156CA328 Mem_holding 0x320BC   Thrash_count 0
Wakeup_reasons 0x0FFFFFFF Default_wakeup_reasons 0x0FFFFFFF
Direct_wakeup_major 0x00000000 Direct_wakeup_minor 0x00000000

Regs R14-R31, CR, PC, MSR at last suspend; R3 from proc creation, PC unused:
R3 : 156CA328 R14: 10CD0984 R15: 00000000 R16: 156CA504 R17: 156CA504
R18: 00000006 R19: FFFFFFFF R20: 00000001 R21: 2421FA78 R22: 13D2A3E0
R23: 000000FF R24: 00000000 R25: 00000001 R26: 13D2A3E0 R27: 2421FA78
R28: 24330AB4 R29: 14BE0000 R30: 156CA328 R31: 107D4240 CR: 40004024
PC : 11A3C6A4 MSR: 02029230

---- Current Process Stack (0x714 bytes used, out of 0xEA60 available) ----

Current SP = 0x2421F930, saved SP = 0x1582AAAC

2421F71C:      1A 2421F918      0 FFFFFFFF 151B1AB0 E8000800 151B1AB0 2421F748
2421F73C:      132BBFA8      E8000 151B1AB0 2421F760 132BC0D0      E8000      9B0A E8000800
2421F75C:      151B1AB0 2421F778 132BC2A0 E8000800      9B0A      800 153B1B7C 2421F790
2421F77C:      123FAF28 2421F790      0      A 151B17E4 2421F798 123FB2BC 2421F7B0
2421F79C:      11C12A90      9B0A 11C12880      A 146C0000 2421F7C0 11BA7384      0
2421F7BC:      146B0000 2421F7D0 11AD3144      A      A 2421F7D8 11C10390 2421F7E0
2421F7DC:      11BB0424 2421F7F0 11BB04E4 2433FCD4 FFFFFFFF 2421F800 107CF880 7FFFFFFF
2421F7FC:      FFFFFFFF 2421F8A8 107CCDF0 20637261 73686475 6D700000      0 2421F840
2421F81C:      0 2421F8B0      0      4A 2E8A00 39760000      4A      0
2421F83C:      2433FCF0 2421F848 2433FCF0      0 11A12ACC 13CD617C 10C7DAAC      0
2421F85C:      2421F8AC 10CD0984      0 156CA504 156CA504      1      0      0
2421F87C:      0      0      0      0      0      0      0 15870804
2421F89C:      0 14850000 FFFFFFFF 2421F930 107CFC7C 2421F8C8 14BB1760      2
2421F8BC:      0 2421F930 14620E40 24330AB4      4A      0      0 2421F8E8
2421F8DC:      10C1FD9C 2421F8F8      0      0      0 15868B74 15868B74 2421F910
2421F8FC:      117CF5C0 2421F968 1586A45C 2421F920 15868B74 2421F918      0 14850000
2421F91C:      0 2421F930 10999978 2421F930      0 2421F940 10999E34 2421F940
2421F93C:      15868B74 2421F948 11B430B8 2421F9B0 10C84444 2421F978      0      0
2421F95C:      0      0 2421F9C0      0 240CC3C8 2421F990 11AE7394      6
2421F97C:      FFFFFFFF      0      0      0 14BE0000      0      0      0
2421F99C:      0      1      0 15868B74 15868B74 2421F9E8 10C83338      0
2421F9BC:      0      71 15868B74 156CA328 13794ACD      0      1      0
2421F9DC:      1511A790 2366B680 15868B74 2421FA00 11BE0C5C 156CA328 156CA328 2366B680
2421F9FC:      15868B74 2421FA18 10C93874 2421FA20      0      0 2366B628 2421FA48
2421FA1C:      10C93D78 2421FA58 10C95370      0 11BB0A98      0      0 15868B74
2421FA3C:      0      0 15868B74 2421FA58 10C94140      3 15868B74 2421FA70
2421FA5C:      10C992EC      0      0      0 156CA328 2421FB08 10CD155C DFFFFFFF
2421FA7C:      FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
2421FA9C:      FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF      1 FFFFFFFF FFFFFFFF
2421FABC:      FFFFFFFF FFFFFFFF      0      0      0      0      0      0
2421FADC:      0      0      0      0      0      0      0      0
2421FAFC:      0      0      0 2421FB10 1099BCFC      0 10992CEC FFFFFFFF

```

===== Interrupt Level Stack Dump =====

---- Level 1 Interrupt stack (0x0 bytes used, out of 0x2328 available) ----

```

intstacks[1]: base 0x156DB3D8 stack 0x156DD6F8 routine 0x0      count 0x0
               size 0x2328      low  0x2328      desc  0x156BE7D0

```

---- Level 2 Interrupt stack (0x3F8 bytes used, out of 0x2328 available) ----

```
intstacks[2]: base 0x156D90B0 stack 0x156DB3D0 routine 0x0 count 0x2
               size 0x2328 low 0x2328 desc 0x156C0C78

156DAFE0: 156DAFE8 FFFFFFFF 156DB020 119E1374 0 B6B8 0 B6F4
156DB000: 156DB020 16035650 156DB0E0 0 2DAE 4 1 16031964
156DB020: 156DB028 119E15BC 156DB050 119E1670 0 B6B8 0 1E3
156DB040: 2DAE 1603191C 156DB050 1603190C 156DB0D0 11BB458C FFFFFFFF FFFFFFFF
156DB060: 0 1E3 16002438 1603191C 1CCB58E0 64 0 2DAE
156DB080: 0 B6B8 FFFFFFFF FFFFFFFF FFFFFFFF 137B49A8 1603560C 160355D0
156DB0A0: 14BABC00 B9DE8DC0 156DB128 0 1C703D84 17F1C788 0 11
156DB0C0: 0 156DB138 160355D0 156DB128 156DB100 11EBBCDC 0 11
156DB0E0: 1CCB58E0 64 156DB110 2DAE 14BAC400 156DB220 17B7B610 0
156DB100: 90040008 151B1AB0 0 151B1AB0 156DB120 132BBFA8 90040 122C3E40
156DB120: 156DB138 132BC0D0 1CCB58E0 0 0 151B1AB0 156DB150 132BCC08
156DB140: 0 156DB240 156DB158 10 156DB1C0 15 156DB170 129ADCC4
156DB160: 156DB170 11 156DB1E0 16 156DB3B0 122BF51C FFFFFFFF FFFFFFFF
156DB180: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
156DB1A0: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
156DB1C0: 0 2000000 0 7FFFFFFF 0 0 FFFFFFFF FFFFFFFF
156DB1E0: 0 20001FF 0 7FFFFFFF 0 0 FFFFFFFF FFFFFFFF
156DB200: 10100 1F4 1F4 77359400 3 2 16 3D
156DB220: 294 294 294 0 0 2 2 1
156DB240: 80000 0 0 FF 0 0 FFFFFFFF FFFFFFFF
156DB260: 0 1EB 0 1FF 0 0 FFFFFFFF FFFFFFFF
156DB280: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
156DB2A0: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
156DB2C0: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
156DB2E0: 0 FFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
156DB300: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
156DB320: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
156DB340: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
156DB360: 15FFFFFF 10 10 10 10 FFFFFFFF FFFFFFFF FFFFFFFF
156DB380: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF 0 14BE0000 146CF310
156DB3A0: 146D0000 14620E80 4 1C7061FC 156DB3C0 132DE01C FFFFFFFF 2
156DB3C0: 156DB3C8 132DDD84 14BAE470 11C0DCD0 FFFFFFFF FFFFFFFF
```

---- Level 3 Interrupt stack (0x350 bytes used, out of 0x2328 available) ----

```
intstacks[3]: base 0x156D6D88 stack 0x156D90A8 routine 0x0 count 0x995
               size 0x2328 low 0x2328 desc 0x156BE924

156D8D60: 156D8D68 FFFFFFFF 156D8DA0 119E1374 0 21EEB 0 21EE4
156D8D80: 0 1 1 160BA724 0 0 156D8DA0 160BA85C
156D8DA0: 156D8DD8 119E1E40 156D8DB0 1 1603560C 156D8F38 160029D8 16035650
156D8DC0: 1 4 0 22030 156D8DD8 160A5670 156D8DF8 119E1F74
156D8DE0: 156D8DF8 4 156D8E08 160BA85C 156D8E08 4 4 160BA85C
156D8E00: 156D8E08 160BA724 156D8E38 11A312A8 156D8E30 119DF688 156D8E30 16035650
156D8E20: 0 16002CA8 156D8E50 16002F78 156D8E50 119DFBD8 0 8E12
156D8E40: 0 156D8F38 156D8E50 1603148C 156D8ED0 11BB458C 156D8E60 1603148C
156D8E60: 0 8E12 16002CA8 156D8F38 1C6FF080 64 0 0
156D8E80: 156D8E90 156D8EE0 156D8E90 1C6FEF9C 156D8ED0 13B40000 14370000 14BC97D0
156D8EA0: 156D8F28 2980A1B9 156D8ED0 84D7317 0 0 1C6FEFAC 0
156D8EC0: 84D7317 1C6FEED4 0 153B1DA4 156D8ED8 11EBBCDC 156D8EE0 11C0C254
156D8EE0: 156D8EF8 132F05E0 2 2980A460 156D8EF8 84D7317 156D8F00 132EFB10
156D8F00: 156D8F18 132B19FC 84D7317 1C6FEED4 0 84D7317 156D8F90 132B1EEC
156D8F20: 0 84D6D76 1C6FF080 64 0 0 0 8E12
156D8F40: 16002CA8 156D8F38 14BE0000 13FD0000 138A0000 160BA4B0 160BA4B0 160BA4B0
156D8F60: 14BABC00 2980A1B9 137C0000 13F50000 14BAC400 0 14BE0000 0
156D8F80: 84D7317 1C6FEED4 0 84D7317 156D8FA8 132B2448 156D8FA0 156D8FB8
156D8FA0: 156D8FA8 11BBE798 156D9030 132B0C9C 156D8FB8 11BBE798 0 7530
156D8FC0: 0 2EE0 0 1 FFFFFFFF FFFFFFFF 4B354370 754D616E
```

■ Displaying a Crash Dump for Supervisor Engine 6-E and 6L-E

```

156D8FE0: 20526576 69657700      0      0      0      0      0      0
156D9000:      0      0 1ADBEEF 1896AD90 156D9030      0      0 146CF310
156D9020: 146D0000 14620EA0      D 1893E4BC 156D9038 134D23A4 156D9058 12023A6C
156D9040:      0 1B1DDC40 156D9050      40      D 1B1DDC40 156D9080 11ED3534
156D9060:      40 132D6244      0 14620EA0 146D0000 14620EA0      D 22D85610
156D9080: 156D9088 133C43C8 156D9098 132DE01C FFFFFFFF      3 156D90A0 132DDE4C
156D90A0: 14BAE470 11C0DCD0 FFFFFFFF FFFFFFFF

```

---- Level 4 Interrupt stack (0x348 bytes used, out of 0x2328 available) ----

```

intstacks[4]: base 0x156D4A60 stack 0x156D6D80 routine 0x0      count 0x8376
              size 0x2328      low 0x2328      desc 0x156BEA78

156D6A40: 156D6A48 FFFFFFFF 156D6A80 119E1374      0 21E4B      0 21E48
156D6A60: FFFFFFFF      1      1 160BA724      0      0 156D6A80 160BA85C
156D6A80: 156D6AB8 119E1E40 FFFFFFFF      1 1603560C 156D6C18 16002938 16035650
156D6AA0:      1      4      0 21F90 156D6AB8 160A5670 156D6AD8 119E1F74
156D6AC0: 156D6AD8      4      4 160BA85C 156D6AD8 160BA724 156D6B08 11A312A8
156D6AE0: 156D6B00 119DF688 156D6BC0 16035650      0 156C504C      1 160BA724
156D6B00: 156D6B08 156C8B5C 156D6B30 11A31B54 156D6B30 119DFBD8      0 8DCE
156D6B20:      0 1603129C 156D6B30 1603128C 156D6BB0 11BB458C FFFFFFFF FFFFFFFF
156D6B40:      0 8DCE 160028E8 1603129C 1BB124AC      64      0      0
156D6B60:      0 1A68C FFFFFFFF FFFFFFFF FFFFFFFF 13B40000 14370000 14BC97D0
156D6B80: 156D6C08 28E47C74 156D6BB0 84B1B7D      0      0 1BB123D8      0
156D6BA0: 84B1B7D 1BB12300      0 153B1DA4 156D6BB8 11EBBCDC 156D6BC0 11C0C254
156D6BC0: 156D6BD8 132F05E0      2 28E47EC0 156D6BD8 84B1B7D 156D6BE0 132EFB10
156D6BE0: 156D6BF8 132B19FC 84B1B7D 1BB12300      0 84B1B7D 156D6C70 132B1EEC
156D6C00: FFFFFFFF FFFFFFFF 1BB124AC      64      0      0      0 8DCE
156D6C20: 160028E8 1603129C 14BE0000 160BA4D8 13860000 13FA0000 160BA428 160BA4B0
156D6C40: 14BABC00 28E47C74 137C0000 13F50000 14BAC400      0 14BE0000      0
156D6C60: 84B1B7D 1BB12300      0 84B1B7D 156D6C88 132B2448 156D6C80 156D6C98
156D6C80: 156D6C88 11BBE798 156D6D10 132B0C9C A0000 14800000      0 1770
156D6CA0:      0 BB8 156D6CD0      1 14BAC400      0 14BE0000 146CF310
156D6CC0: 146D0000 151B1AB0 156D6CF0 151B1AB0 156D6CE0      0 156D6D10 146CF310
156D6CE0: 2029230 14620EC0 156D6D20 153B1DA4 156D6CF8 151B1AB0 3012000 11C0C254
156D6D00: 156D6D18 132F05E0 3012020 153B1C8C 156D6D30 11C0FE70 156D6D20 153B1C8C
156D6D20: 156D6D40 11C0FE28 156D6D38 153B1C8C 3012040 153B1C8C 156D6D48 11C100CC
156D6D40: 156D6D50 11C10348 14BE0000 146C62B4 156D6D78 11BB0A10 FFFFFF 1CCAFCB8
156D6D60: 156D6D70 146CF310 146D0000 14620EC0      2A      4 14BAE470 11C0DCD0
156D6D80: FFFFFFFF FFFFFFFF

```

---- Level 5 Interrupt stack (0x170 bytes used, out of 0x2328 available) ----

```

intstacks[5]: base 0x156D2738 stack 0x156D4A58 routine 0x0      count 0x8843
              size 0x2328      low 0x2328      desc 0x156BEBCC

156D48F0: 156D4918 FFFFFFFF 14BAC400      0 14BE0000 13B50000      0 151B1AB0
156D4910: A0000060 151B1AB0 156D4928 132BBFA8 A0000 151B1AB0 156D4940 132BC0D0
156D4930: A0000 14C00000 A0000060 151B1AB0 156D4958 132BC42C A0000060 14C00000
156D4950:      60 189A84E0 156D4970 12405FD8 156D4970      1      1 1B1AD9E0
156D4970: 156D4978 124067F8 156D4998 11CB7020 156D49A8 14C00000 FFFFFFFF      0
156D4990: 14BE0000 13B50000      2 1B5A1068 2029230 153B1DA4 156D49D0 1BB11BE0
156D49B0: 156D49B8      0 14BE0000 13B50000      3 14380000 1B1AD9E0 153B1DA4
156D49D0: 156D49D8 132EFB10 156D49E0 11C0C254 156D49F8 132F05E0      2 2A7A9FE0
156D49F0: 2029230 153B1DA4 156D4A00 132EFB10 156D4A18 11CB7200      3 14380000
156D4A10: 14380000 153B1DA4 156D4A20 1338A684 156D4A48 132F0B04      2 2A7A9F6F
156D4A30: FFFFFFFF 146CF310 146D0000 14620EE0      0      5 156D4A50 11BAE8C0
156D4A50: 14BAE470 11C0DCD0 FFFFFFFF FFFFFFFF

```

---- Level 6 Interrupt stack (0x0 bytes used, out of 0x2328 available) ----

```

intstacks[6]: base 0x156D0410 stack 0x156D2730 routine 0x0      count 0x0
              size 0x2328      low 0x2328      desc 0x156BED20

```



```

---- Level 7 Interrupt stack (0x0 bytes used, out of 0x2328 available) ----

intstacks[7]: base 0x156CE0E8 stack 0x156D0408 routine 0x0      count 0x0
              size 0x2328      low  0x2328      desc   0x156BEE74

---- Level 8 Interrupt stack (base 0x0, size 0x0) is invalid ----

---- Level 9 Interrupt stack (base 0x0, size 0x0) is invalid ----

===== Register Memory Dump =====

Reg00(PC ): 10999E70
Reg01(MSR): 2029230 [Not RAM Addr]
Reg02(CR ): 20004022
Reg03(LR ): 10999E34
Reg04(CTR): 108EC3EC
Reg05(XER):      0 [Not RAM Addr]
Reg06(DAR):      0 [Not RAM Addr]
Reg07(DSISR):    0 [Not RAM Addr]
Reg08(DEC):    83695 [Not RAM Addr]
Reg09(TBU):      2 [Not RAM Addr]
Reg10(TBL): 2A7D42DA
Reg11(IMMR):    0 [Not RAM Addr]
Reg12(R0 ): 10999E34
Reg13(R1 ): 2421F930
Reg14(R2 ):     1E [Not RAM Addr]
Reg15(R3 ): 234BBFD8 [In malloc Block 0x234BBB54] [Last malloc Block 0x234BBB10]
Reg16(R4 ):      A [Not RAM Addr]
Reg17(R5 ):      0 [Not RAM Addr]
Reg18(R6 ): 2421F918
Reg19(R7 ):      0 [Not RAM Addr]
Reg20(R8 ):      0 [Not RAM Addr]
Reg21(R9 ):      0 [Not RAM Addr]
Reg22(R10): 14850000
Reg23(R11): 234BBFD4
Reg24(R12): EB93A100 [Not RAM Addr]
Reg25(R13): B4E9F3F3 [Not RAM Addr]
Reg26(R14): 10CD0984
Reg27(R15):      0 [Not RAM Addr]
Reg28(R16): 156CA504 [In malloc Block 0x156CA2F0]
Reg29(R17): 156CA504
Reg30(R18):      1 [Not RAM Addr]
Reg31(R19):      0 [Not RAM Addr]
Reg32(R20):      0 [Not RAM Addr]
Reg33(R21):      0 [Not RAM Addr]
Reg34(R22):      0 [Not RAM Addr]
Reg35(R23):      0 [Not RAM Addr]
Reg36(R24):      0 [Not RAM Addr]
Reg37(R25):      0 [Not RAM Addr]
Reg38(R26):      0 [Not RAM Addr]
Reg39(R27):      0 [Not RAM Addr]
Reg40(R28): 15870804 [In malloc Block 0x158707DC] [Last malloc Block 0x15870790]
Reg41(R29):      0 [Not RAM Addr]
Reg42(R30): 14850000
Reg43(R31):      0 [Not RAM Addr]

buffer check=0 sched_hc=0x0

---- block0 ptr=2421F8D0 is_malloc=0 length=0x260 ----

2421F890:      0      0 15870804      0 14850000 FFFFFFFF 2421F930 107CFC7C
2421F8B0: 2421F8C8 14BB1760      2      0 2421F930 14620E40 24330AB4      4A

```

■ Displaying a Crash Dump for Supervisor Engine 6-E and 6L-E

```

2421F8D0:      0      0 2421F8E8 10C1FD9C 2421F8F8      0      0      0
2421F8F0: 15868B74 15868B74 2421F910 117CF5C0 2421F968 1586A45C 2421F920 15868B74
2421F910: 2421F918      0 14850000      0 2421F930 10999978 2421F930      0
2421F930: 2421F940 10999E34 2421F940 15868B74 2421F948 11B430B8 2421F9B0 10C84444
2421F950: 2421F978      0      0      0      0 2421F9C0      0 240CC3C8
2421F970: 2421F990 11AE7394      6 FFFFFFFF      0      0      0 14BE0000
2421F990:      0      0      0      0      1      0 15868B74 15868B74
2421F9B0: 2421F9E8 10C83338      0      0      71 15868B74 156CA328 13794ACD
2421F9D0:      0      1      0 1511A790 2366B680 15868B74 2421FA00 11BE0C5C
2421F9F0: 156CA328 156CA328 2366B680 15868B74 2421FA18 10C93874 2421FA20      0
2421FA10:      0 2366B628 2421FA48 10C93D78 2421FA58 10C95370      0 11BB0A98
2421FA30:      0      0 15868B74      0      0 15868B74 2421FA58 10C94140
2421FA50:      3 15868B74 2421FA70 10C992EC      0      0      0 156CA328
2421FA70: 2421FB08 10CD155C DFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
2421FA90: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
2421FAB0:      1 FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF      0      0      0
2421FAD0:      0      0      0      0      0      0      0      0
2421FAF0:      0      0      0      0      0      0 2421FB10 1099BCFC
2421FB10:      0 10992CEC FFFFFFFF FD0110DF AB1234CD      0      0 1378AC50

```

```
---- block1 ptr=10999E34 is_malloc=0 length=0x13C ----
```

```

10999DF4: 3D2014BE 8009845C 2F800000 419E0010 3D2014BE 806983C4 4800000C 3D201485
10999E14: 80698C20 4927268D 38000000 3D2014BE 90098484 3D201485 90098CAC 4BFFFAB9
10999E34: 2F9F0000 419E0038 3D2014BE 80098498 2F800000 40BE0028 3860FFFE 3C8013CD
10999E54: 3884617C 4BE35D99 2F9F0003 409D0008 3BE00003 1C7F03E8 4909658D 7FE10808
10999E74: 38000000 3D201445 900900E4 80010014 7C0803A6 83E1000C 38210010 4E800020
10999E94: 9421FFF8 7C0802A6 9001000C 7C862378 38832010 38602000 3CA013CD 38A56198
10999EB4: 482E2201 8001000C 7C0803A6 38210008 4E800020 7C681B78 7C6A1B78 39200000
10999ED4: 89630000 2F8B0000 419E0078 380BFF9F 2B800005 40BD0010 380BFFD0 2B800009
10999EF4: 419D0060 552B2036 880A0000 7C000774 2F800039 419D0014 7D2B0214 3929FFD0
10999F14: 394A0001 48000018 892A0000 7D290774 7D2B4A14 3929FFA9 394A0001 896A0000
10999F34: 2F8B0000 419E001C 380BFF9F 2B800005 40BDFFB4 380BFFD0 2B800009 409DFFA8
10999F54: 38600000 7F8A4000 4D9E0020 91250000 91440000 38600001 4E800020

```

```
---- block2 ptr=20004020 is_malloc=0 length=0x100 ----
```

```

20003FE0:  ADBEEF      0      0      0      BEEF      0 7ADBEEF 1F724E78
20004000:  12800      0 1ADBEEF 12216BE8 1BE7FE60      BEEF 1BE7FE70 1813277C
20004020: 1BE7FE78 1253BA98      FE70 FFFFFFFF      3      0 FFFFFFFF FFFFFFFF
20004040: FFFFFFFF FFFFFFFF 10800      0      0 1FFFF 1BE7FF38 11F86C4C
20004060: 180AB988 1BE7FEF8 2027FEE8 FF84BEEF      3      0      0      0
20004080:      BEEF      0      BEEF      0      0      0      0      0
200040A0:      0  ADBEEF      0      0      0      BEEF      0 7ADBEEF
200040C0: 1F724E78 12800      0 1ADBEEF 12216BE8 1BE7FE60      BEEF 1BE7FE70
200040E0: 1813277C 1BE7FE78 1253BA98      FE70 FFFFFFFF      3      0 FFFFFFFF
20004100: FFFFFFFF FFFFFFFF FFFFFFFF 10800      0      0 1FFFF 1BE7FF38

```

```
---- block3 ptr=108EC3EC is_malloc=0 length=0x100 ----
```

```

108EC3AC: 7C0803A6 83E1000C 38210010 4E800020 9421FFF8 7C0802A6 9001000C 2C030000
108EC3CC: 40A2000C 4BFFF7F9 48000008 4BFFFB79 8001000C 7C0803A6 38210008 4E800020
108EC3EC: 9421FFF8 7C0802A6 9001000C 3D20149A 8009FB08 2F800000 41BE0028 4BFEF791
108EC40C: 3D201485 81299C1C 81290024 7D2903A6 38600000 4E800421 38600000 4BFF0EFD
108EC42C: 8001000C 7C0803A6 38210008 4E800020 9421FFF8 7C0802A6 9001000C 3D20149A
108EC44C: 8009FB08 2F800000 41BE0028 4BFEF741 3D201485 81299C1C 81290024 7D2903A6
108EC46C: 38600000 4E800421 38600000 4BFF0EAD 8001000C 7C0803A6 38210008 4E800020
108EC48C: 9421FFE8 7C0802A6 BF810008 9001001C 7C9C2378 38000000 7C7E1B79 418200B0
108EC4AC: 7FC3F378 492150A5 38000000 7C7D1B79 4182009C 3BE00000 7F9FE800 40BC008C
108EC4CC: 7C9EF8AE 3D201442 8069EDA4 7C840774 49214EFD 2F830000 409E0064 2F9C0000

```

```
---- block4 ptr=234BBB10 is_malloc=1 length=0x100 ----
```

```
234BBAD0: FFFE0000      0 13C9C0B0 107FD290 234BBB10 234BBA24 8000000E      1
```

```

234BBAF0:      0 23056294 23054D90 13597D4C      1      0      0 FD0110DF
234BBB10: AB1234CD FFFE0000      0 13D9A594 10027870 234BBB54 234BBAE0 8000000E
234BBB30:      1      0      1      4      0      0      0      0
234BBB50: FD0110DF AB1234CD FFFE0000      0 156CD7F4 119EB018 234BC350 234BBB24
234BBB70: 800003EA      1 119F6768      0 234466EC 234FFE84      0 156CD7B8
234BBB90:      64      77      C000C      0      0      0      0      0
234BBBB0: 30000 14BB1760 52656720 46756E63 74696F6E 20310000 234BBDB8 234BC34C
234BBBD0:      0      0      0 234BBDB8 234BBDC4 234BBDD0 234BBDDC 234BBDE8
234BBBF0: 234BBDF4 234BBE00 234BBE0C 234BBE18 234BBE24 234BBE30 234BBE3C 234BBE48

```

```

---- block5 ptr=15870790 is_malloc=1 length=0x14C ----

```

```

15870750:      0      0      0      0      0      0      0      0
15870770:      0      0      0      0      0      0      0      0 FD0110DF
15870790: AB1234CD FFFE0000      0 13D9A594 10CA1538 158707DC 1586B958 80000012
158707B0:      1 4928F581      0      1 23C0BED0      0      0      0
158707D0: 1449E540      F FD0110DF AB1234CD FFFE0000      0 13D2B910 10C89680
158707F0: 15870840 158707A4 8000001E      1 38210008 158711F0 13D40DB8      0
15870810: 13D3EC78      0      0      0 10CC65BC      7 144B0254 15870868
15870830: 158708AC      0      0 FD0110DF AB1234CD FFFE0000      0 13D2B91C
15870850: 10C896E8 15870884 158707F0 8000000E      1 7C09002E 158708F0 23F0FB18
15870870:      17      0      0      0 FD0110DF AB1234CD FFFE0000      0
15870890: 13D2B92C 10C8970C 158708C8 15870854 8000000E      1 7D6B4A14 15870980
158708B0: 15871160      8      0      0      0 FD0110DF AB1234CD FFFE0000
158708D0:      0 13D2BA48 10C8BE78

```

```

---- block6 ptr=15870790 is_malloc=1 length=0x100 ----

```

```

15870750:      0      0      0      0      0      0      0      0
15870770:      0      0      0      0      0      0      0      0 FD0110DF
15870790: AB1234CD FFFE0000      0 13D9A594 10CA1538 158707DC 1586B958 80000012
158707B0:      1 4928F581      0      1 23C0BED0      0      0      0
158707D0: 1449E540      F FD0110DF AB1234CD FFFE0000      0 13D2B910 10C89680
158707F0: 15870840 158707A4 8000001E      1 38210008 158711F0 13D40DB8      0
15870810: 13D3EC78      0      0      0 10CC65BC      7 144B0254 15870868
15870830: 158708AC      0      0 FD0110DF AB1234CD FFFE0000      0 13D2B91C
15870850: 10C896E8 15870884 158707F0 8000000E      1 7C09002E 158708F0 23F0FB18
15870870:      17      0      0      0 FD0110DF AB1234CD FFFE0000      0

```

```

---- block7 ptr=240CC354 is_malloc=1 length=0x14C ----

```

```

240CC314:      2 240CC37C      0      0      0      0      2      0
240CC334:      0      0      0      0      0      0      0      0 FD0110DF
240CC354: AB1234CD CD0000 24031228 240CC1F0 1011D6CC 240CC3A0 240CC2F0 80000012
240CC374:      1 D0D0D0D      1      8 10B4FEE8      0      2      4
240CC394:      0 10B50054 FD0110DF AB1234CD FFFE0000      0 13D9A594 10C8D690
240CC3B4: 240CC404 240CC368 8000001E      1 10C7DB80      1      40      0
240CC3D4:      1 23553660 10C8D61C      0      0      0      0      0
240CC3F4:      0      0      0 FD0110DF AB1234CD CD0000 24031228 1362B664
240CC414: 10DF2B24 240CEA20 240CC3B4 800012FA      1 D0D0D0D 2416C8DC ABADCAFE
240CC434:      C      C      0      CD 80000000      0      0      0
240CC454:      0      0      0      0      0      0      0      0
240CC474:      0      0      0      0      0      0      0      0
240CC494:      0      0      0

```

```

---- block8 ptr=13794ACC is_malloc=0 length=0x100 ----

```

```

13794A8C: 51522E2E 2E2E2E2E 5C2E5354 55565758 595A2E2E 2E2E2E2E 30313233 34353637
13794AAC: 38392E2E 2E2E2E2E      0 30313233 34353637 38396162 63646566      0
13794ACC: 202020 20202020 20202828 28282820 20202020 20202020 20202020 20202020
13794AEC: 20881010 10101010 10101010 10101010 10040404 4040404 4040410 10101010
13794B0C: 10104141 41414141 1010101 1010101 1010101 1010101 1010101 10101010
13794B2C: 10104242 42424242 2020202 2020202 2020202 2020202 2020202 10101010
13794B4C: 20000000 436F6D6D 756E6963 6174696F 6E206572 726F7220 6F6E2073 656E6400

```

```

13794B6C: 546F6F20 6D616E79 206C696E 6B730000 426C6F63 6B206465 76696365 20726571
13794B8C: 75697265 64000000 41726720 6C697374 20746F6F 20626967          0 4E6F2073
13794BAC: 75636820 70726F63 65737300 4E6F7420 6F776E65 72000000 4E6F2073 75636820

```

Log buffer:

```

6:21:19 UTC Mon Sep 6 2010
CMD: 'alias exec cas clear auth sess' 06:21:19 UTC Mon Sep 6 2010
CMD: 'alias exec sas show auth sess' 06:21:19 UTC Mon Sep 6 2010
CMD: 'alias exec cpu show proc cpu | inc CPU' 06:21:19 UTC Mon Sep 6 2010
CMD: 'alias exec si show run int gi6/25' 06:21:19 UTC Mon Sep 6 2010
CMD: 'line con 0' 06:21:19 UTC Mon Sep 6 2010
CMD: ' exec-timeout 0 0' 06:21:19 UTC Mon Sep 6 2010
CMD: ' stopbits 1' 06:21:19 UTC Mon Sep 6 2010
CMD: ' speed 38400' 06:21:19 UTC Mon Sep 6 2010
CMD: 'line vty 0 4' 06:21:19 UTC Mon Sep 6 2010
CMD: 'scheduler runtime netinput 100' 06:21:19 UTC Mon Sep 6 2010
CMD: 'mac address-table static 0023.abf8.3303 vlan 1 interface GigabitEthernet6/15'
06:21:19 UTC Mon Sep 6 2010
CMD: 'end' 06:21:19 UTC Mon Sep 6 2010

*Sep 6 06:21:19.103: %SW_VLAN-6-VTP_DOMAIN_NAME_CHG: VTP domain name changed to campus1.
*Sep 6 06:21:21.779: %SYS-5-CONFIG_I: Configured from memory by console
*Sep 6 06:21:21.875: %SYS-5-RESTART: System restarted --
Cisco IOS Software, Catalyst 4500 L3 Switch Software (cat4500e-ENTSERVICES-M),
Experimental Version 12.2(20100723:074204) [/./././././ios/sys 179]
Copyright (c) 1986-2010 by Cisco Systems, Inc.
Compiled Mon 06-Sep-10 22:11 by cisco
*Sep 6 06:21:23.363: Slot 0 : delete
*Sep 6 06:21:23.363: K5SuperportSetConfig:
*Sep 6 06:21:23.363: num of Superports : 4, SuperportIds( 57, 57, 57, 57 )
*Sep 6 06:21:23.363: K5SuperportGroupMode Xauik5PortSpeedType 10G10
*Sep 6 06:21:23.363: K5SuperportConfig:
*Sep 6 06:21:23.363: K5SuperportUsageState Populated, 4K5SuperportManagementProtocol
VsiK5HeaderType K10 SCH Preamble, Max SubportId : 9
*Sep 6 06:21:23.363: num of subports : 1, SubportConfig:
*Sep 6 06:21:23.363: SubportConfig( K5SubportId : 0, PimHwPhyportId : 240 )
*Sep 6 06:21:23.711: %C4K_JOB-4-OVERRUN: (Suppressed 1 times)Job Lj-poll ran 20941
microseconds (its runTimeMax was 2000)
*Sep 6 06:21:23.711: Slot 0 : new
*Sep 6 06:21:23.711: K5SuperportSetConfig:
*Sep 6 06:21:23.711: num of Superports : 4, SuperportIds( 57, 57, 57, 57 )
*Sep 6 06:21:23.711: K5SuperportGroupMode Xauik5PortSpeedType 10G10
*Sep 6 06:21:23.711: K5SuperportConfig:
*Sep 6 06:21:23.711: K5SuperportUsageState Populated, 3K5SuperportManagementProtocol
VsiK5HeaderType K10 SCH Preamble, Max SubportId : 9
*Sep 6 06:21:23.711: num of subports : 2, SubportConfig:
*Sep 6 06:21:23.711: SubportConfig( K5SubportId : 8, PimHwPhyportId : 242 )
*Sep 6 06:21:23.711: SubportConfig( K5SubportId : 9, PimHwPhyportId : 243 )CMD: 'en'
06:21:56 UTC Mon Sep 6 2010
CMD: 'plat' 06:21:57 UTC Mon Sep 6 2010
CMD: 'platform cr' 06:21:57 UTC Mon Sep 6 2010
CMD: 'platform crashdump d' 06:21:58 UTC Mon Sep 6 2010
CMD: 'platform crashdump ' 06:21:58 UTC Mon Sep 6 2010

Supervisor (WS-X45-SUP6-E) Board Specific Crash Data:
MCSR: 0x0
L1CSR0: 0x10001 L1CSR1: 0x10001
SRR0: 0x10999e70 CSRR0: 0x0 MCSRR0: 0x0
MCAR: 0x0
ESR: 0x2000000
CISR0: 0x0 CISR1: 0x0
L2CTL: 0xa0000000
L2CAPDATAHI: 0x0 L2CAPDATALO: 0x0

```

```
L2CAPTECC: 0x0
L2ERRDET: 0x0
L2ERRDIS: 0x0
L2ERRATTR: 0x0
L2ERRADDRH: 0x0L2ERRADDRL: 0x0
L2_ERRCTL: 0x0
DDR_CAPTURE_DATA_HI: 0x0 DDR_CAPTURE_DATA_LO: 0x0
DDR_CAPTURE_ECC: 0x0
DDR_ERR_DETECT: 0x0
DDR_ERR_DISABLE: 0x0
DDR_ERR_INT_EN: 0x9
DDR_CAPTURE_ATTRIBUTES: 0x0
DDR_CAPTURE_ADDRESS: 0x0
DDR_CAPTURE_EXT_ADDRESS: 0x0
DDR_ERR_SBE: 0xff0000
PCI_ERR_DR: 0x0
PCI_ERR_ATTRIB: 0x0
PCI_ERR_ADDR: 0x0
PCI_ERR_EXT_ADDR: 0x0
PCI_ERR_DH: 0x0PCI_ERR_DL: 0x0
Machine Check Interrupt Count: 0
L1 Instruction Cache Parity Errors: 0
L1 Instruction Cache Parity Errors (CPU30): 0
L1 Data Cache Parity Errors: 0
```

Jawa Crash Data:

```
Interrupt Mask: 0xe180
Interrupt: 0x0
```

```
Galk5DriverMan( 0 )
  SlotType( 3 )
  State( Galk5DriverManStateReady )
  SilentRollRegister( 0 )
  GldMajorVersion( 0 )
  CardRevision( 0 )
  GldMinor( 1)
  Load Dynamic Driver( No )
```

```
Galk5DriverMan( 1 )
  SlotType( 1 )
  State( Galk5DriverManStateReady )
  SilentRollRegister( 0 )
  GldMajorVersion( 0 )
  CardRevision( 0 )
  GldMinor( 1)
  Load Dynamic Driver( No )
```

Switch#



Configuring the Switch for the First Time

This chapter describes how to initially configure the switch.

The information presented here supplements the administration information and procedures in the [Configuration Fundamentals Configuration Guide](#) and the [Cisco IOS Configuration Fundamentals Command Reference](#).

This chapter includes the following major sections:

- [Default Switch Configuration, page 3-1](#)
- [Configuring DHCP-Based Autoconfiguration, page 3-2](#)
- [Configuring the Switch, page 3-8](#)
- [Controlling Access to Privileged EXEC Commands, page 3-13](#)
- [Recovering a Lost Enable Password, page 3-25](#)
- [Modifying the Supervisor Engine Startup Configuration, page 3-25](#)
- [Replacing and Rolling-Back Configuration, page 3-33](#)
- [Resetting a Switch to Factory Default Settings, page 3-34](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

Default Switch Configuration

This section describes the default configurations for the switch. [Table 3-1](#) shows the default configuration settings for each feature.

Table 3-1 **Default Switch Configuration**

Feature	Default Settings
Administrative connection	Normal mode
Global switch information	No default value for system name, system contact, and location
System clock	No value for system clock time

Table 3-1 **Default Switch Configuration (continued)**

Feature	Default Settings
Passwords	No passwords are configured for normal mode or enable mode (press the Return key)
Switch prompt	Switch>
Interfaces	Enabled, with speed and flow control autonegotiated, and without IP addresses

Configuring DHCP-Based Autoconfiguration

These sections describe how to configure DHCP-based autoconfiguration:

- [About DHCP-Based Autoconfiguration, page 3-2](#)
- [DHCP Client Request Process, page 3-3](#)
- [Configuring the DHCP Server, page 3-4](#)
- [Configuring the TFTP Server, page 3-4](#)
- [Configuring the DNS Server, page 3-5](#)
- [Configuring the Relay Device, page 3-5](#)
- [Obtaining Configuration Files, page 3-6](#)
- [Example Configuration, page 3-7](#)

If your DHCP server is a Cisco device, or if you are configuring the switch as a DHCP server, refer to the “IP Addressing and Services” section in the *Cisco IOS IP and IP Routing Configuration Guide for Cisco IOS Release 12.1* for additional information about configuring DHCP.

About DHCP-Based Autoconfiguration

**Note**

Beginning with Release 12.2(20)EW, you can enable DHCP AutoConfiguration by entering the **write erase** command. This command clears the startup-config in NVRAM. In images prior to Release 12.2(20)EW, this command does not enable autoconfiguration.

DHCP provides configuration information to Internet hosts and internetworking devices. This protocol consists of two components: one component for delivering configuration parameters from a DHCP server to a device and another component that is a mechanism for allocating network addresses to devices. DHCP is built on a client-server model, in which designated DHCP servers allocate network addresses and deliver configuration parameters to dynamically configured devices. The switch can act as both a DHCP client and a DHCP server.

With DHCP-based autoconfiguration, no DHCP client-side configuration is needed on your switch because your switch (the DHCP client) is automatically configured at startup with IP address information and a configuration file. However, you need to configure the DHCP server or the DHCP server feature on your switch for various lease options associated with IP addresses. If you are using DHCP to relay the configuration file location on the network, you might also need to configure a Trivial File Transfer Protocol (TFTP) server and a Domain Name System (DNS) server.

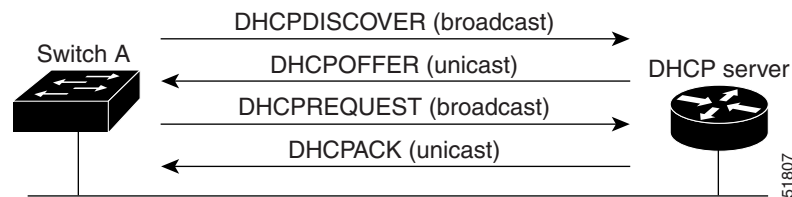
DHCP-based autoconfiguration replaces the BOOTP client functionality on your switch.

DHCP Client Request Process

At startup the switch automatically requests configuration information from a DHCP server if a configuration file is not present on the switch.

Figure 3-1 shows the sequence of messages that are exchanged between the DHCP client and the DHCP server.

Figure 3-1 DHCP Client and Server Message Exchange



The client, Switch A, broadcasts a DHCPDISCOVER message to locate a DHCP server. The DHCP server offers configuration parameters (such as an IP address, subnet mask, gateway IP address, DNS IP address, lease for the IP address, and so forth) to the client in a DHCPOFFER unicast message.

In a DHCPREQUEST broadcast message, the client returns a formal request for the offered configuration information to the DHCP server. The formal request is broadcast so that all other DHCP servers that received the DHCPDISCOVER broadcast message from the client can reclaim the IP addresses that they offered to the client.

The DHCP server confirms that the IP address has been allocated to the client by returning a DHCPACK unicast message to the client. With this message, the client and server are bound, and the client uses the configuration information that it received from the server. The amount of information the switch receives depends on how you configure the DHCP server. For more information, see the [“Configuring the DHCP Server”](#) section on page 3-4.

If the configuration parameters sent to the client in the DHCPOFFER unicast message are invalid (if configuration error exists), the client returns a DHCPDECLINE broadcast message to the DHCP server.

The DHCP server sends the client a DHCPNAK denial broadcast message, which means that the offered configuration parameters have not been assigned, that an error has occurred during the negotiation of the parameters, or that the client has been slow in responding to the DHCPOFFER message. (The DHCP server might have assigned the parameters to another client.)

A DHCP client might receive offers from multiple DHCP servers and can accept any of them; however, the client usually accepts the first offer it receives. The offer from the DHCP server is not a guarantee that the IP address will be allocated to the client; however, the server usually reserves the address until the client has had a chance to formally request the address.

Troubleshooting Tips

Sometimes due to firmware upgrades or client VLAN configuration changes on the device, the DHCP client may send a DHCPREQUEST message with a cached IP address. If the DHCP server does not accept the request, the client may continue to send DHCPREQUEST messages instead of a DHCPDISCOVER message, and fail to get an IP address from the DHCP server.

To remedy this situation, use the **renew deny unknown** command in DHCP pool configuration mode. This command forces the DHCP server to reject renewal requests from clients and the DHCP server sends a DHCPNAK denial message to the client, forcing the client back to its initial state.

Configuring the DHCP Server

A switch can act as both the DHCP client and the DHCP server. By default, the Cisco IOS DHCP server and relay agent features are enabled on your switch.

You should configure the DHCP server, or the DHCP server feature running on your switch, with reserved leases that are bound to each switch by the switch hardware address.

If you want the switch to receive IP address information, you must configure the DHCP server with these lease options:

- IP address of the client (required)
- Subnet mask of the client (required)
- DNS server IP address (optional)
- Router IP address (required)

**Note**

The router IP address is the default gateway address for the switch.

If you want the switch to receive the configuration file from a TFTP server, you must configure the DHCP server with these lease options:

- TFTP server name or IP address (required)
- Boot filename (the name of the configuration file that the client needs) (recommended)
- Host name (optional)

Depending on the settings of the DHCP server or the DHCP server feature running on your switch, the switch can receive IP address information, the configuration file, or both.

If you do not configure the DHCP server, or the DHCP server feature running on your switch, with the lease options described earlier, the switch replies to client requests with only those parameters that are configured. If the IP address and subnet mask are not in the reply, the switch is not configured. If the router IP address or TFTP server name (or IP address) are not found, the switch might send broadcast, instead of unicast, TFTP requests. Unavailability of other lease options does not impact autoconfiguration.

The DHCP server, or the DHCP server feature running on your switch, can be on the same LAN or on a different LAN than the switch. If the DHCP server is running on a different LAN, you should configure a DHCP relay, which forwards broadcast traffic between two directly connected LANs. A router does not forward broadcast packets, but it forwards packets based on the destination IP address in the received packet. For more information on relay devices, see the [“Configuring the Relay Device” section on page 3-5](#).

Configuring the TFTP Server

Based on the DHCP server configuration, the switch attempts to download one or more configuration files from the TFTP server. If you configured the DHCP server to respond to the switch with all the options required for IP connectivity to the TFTP server, and if you configured the DHCP server with a TFTP server name, address, and configuration filename, the switch attempts to download the specified configuration file from the specified TFTP server.

If you did not specify the configuration filename or the TFTP server name, or if the configuration file could not be downloaded, the switch attempts to download a configuration file using various combinations of filenames and TFTP server addresses. The files include the specified configuration

filename (if any) and the following files: `network-config`, `cisconet.cfg`, `hostname.config`, or `hostname.cfg`, where *hostname* is the current hostname of the switch and `router-config` and `ciscotr.cfg`. The TFTP server addresses used include the specified TFTP server address (if any) and the broadcast address (255.255.255.255).

For the switch to successfully download a configuration file, the TFTP server must contain one or more configuration files in its base directory. The files can include the following:

- The configuration file named in the DHCP reply (the actual switch configuration file).
- The `network-config` or the `cisconet.cfg` file (known as the default configuration files).
- The `router-config` or the `ciscotr.cfg` file. (These files contain commands common to all switches. Normally, if the DHCP and TFTP servers are properly configured, these files are not accessed.)

If you specify the TFTP server name in the DHCP server-lease database, you must also configure the TFTP server name-to-IP-address mapping in the DNS-server database.

If the TFTP server you plan to use is on a different LAN from the switch, or if you plan to access it with the switch through the broadcast address (which occurs if the DHCP server response does not contain all the required information described earlier), you must configure a relay to forward the TFTP packets to the TFTP server. For more information, see the [“Configuring the Relay Device” section on page 3-5](#). The preferred solution is to configure either the DHCP server or the DHCP server feature running on your switch with all the required information.

Configuring the DNS Server

The DHCP server, or the DHCP server feature running on your switch, uses the DNS server to resolve the TFTP server name to an IP address. You must configure the TFTP server name-to-IP address map on the DNS server. The TFTP server contains the configuration files for the switch.

You can configure the IP addresses of the DNS servers in the lease database of the DHCP server where the DHCP replies retrieve them. You can enter up to two DNS server IP addresses in the lease database.

The DNS server can be on the same or on a different LAN as the switch. If it is on a different LAN, the switch must be able to access it through a router.

Configuring the Relay Device

You must configure a relay device to forward received broadcast packets to the destination host whenever a switch sends broadcast packets to which a host on a different LAN must respond. Examples of such broadcast packets are DHCP, DNS, and in some cases, TFTP packets.

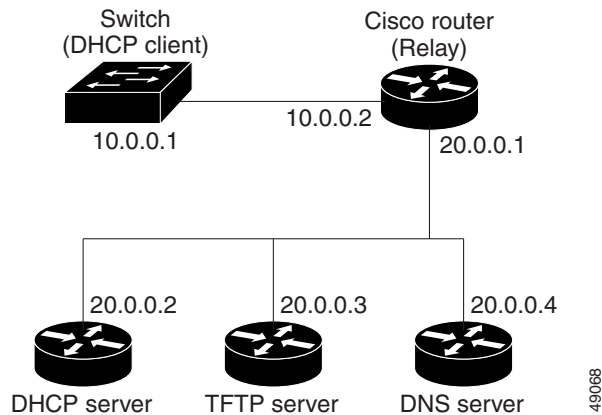
If the relay device is a Cisco router, enable IP routing (**ip routing** global configuration command) and configure helper addresses (**ip helper-address** interface configuration command). For example, in [Figure 3-2](#), configure the router interfaces as follows:

On interface 10.0.0.2:

```
router(config-if)# ip helper-address 20.0.0.2
router(config-if)# ip helper-address 20.0.0.3
router(config-if)# ip helper-address 20.0.0.4
```

On interface 20.0.0.1:

```
router(config-if)# ip helper-address 10.0.0.1
```

Figure 3-2 Relay Device Used in Autoconfiguration

49068

Obtaining Configuration Files

Depending on the availability of the IP address and the configuration filename in the DHCP reserved lease, the switch obtains its configuration information in these ways:

- The IP address and the configuration filename are reserved for the switch and provided in the DHCP reply (one-file read method).

The switch receives its IP address, subnet mask, TFTP server address, and the configuration filename from either the DHCP server or the DHCP server feature running on your switch. The switch sends a unicast message to the TFTP server to retrieve the named configuration file from the base directory of the server, and upon receipt, completes its boot-up process.

- The IP address and the configuration filename is reserved for the switch, but the TFTP server address is not provided in the DHCP reply (one-file read method).

The switch receives its IP address, subnet mask, and the configuration filename from either the DHCP server or the DHCP server feature running on your switch. The switch sends a broadcast message to a TFTP server to retrieve the named configuration file from the base directory of the server, and upon receipt, completes its boot-up process.

- Only the IP address is reserved for the switch and provided in the DHCP reply. The configuration filename is not provided (two-file read method).

The switch receives its IP address, subnet mask, and the TFTP server address from either the DHCP server or the DHCP server feature running on your switch. The switch sends a unicast message to the TFTP server to retrieve the `network-config` or `cisconet.cfg` default configuration file. (If the `network-config` file cannot be read, the switch reads the `cisconet.cfg` file.)

The default configuration file contains the host names-to-IP-address mapping for the switch. The switch fills its host table with the information in the file and obtains its host name. If the host name is not found in the file, the switch uses the host name in the DHCP reply. If the host name is not specified in the DHCP reply, the switch uses the default *Switch* as its host name.

After obtaining its host name from the default configuration file or the DHCP reply, the switch reads the configuration file that has the same name as its host name (`hostname-config` or `hostname.cfg`, depending on whether or not the `network-config` file or the `cisconet.cfg` file was read earlier) from the TFTP server. If the `cisconet.cfg` file is read, the filename of the host is truncated to eight characters.

If the switch cannot read the network-config, cisco.net.cfg, or the hostname file, it reads the router-config file. If the switch cannot read the router-config file, it reads the ciscotr.cfg file.

**Note**

The switch broadcasts TFTP server requests provided that one of these conditions is met: the TFTP server is not obtained from the DHCP replies; all attempts to read the configuration file through unicast transmissions fail; or the TFTP server name cannot be resolved to an IP address.

Example Configuration

Figure 3-3 shows a network example for retrieving IP information using DHCP-based autoconfiguration.

Figure 3-3 DHCP-Based Autoconfiguration Network Example

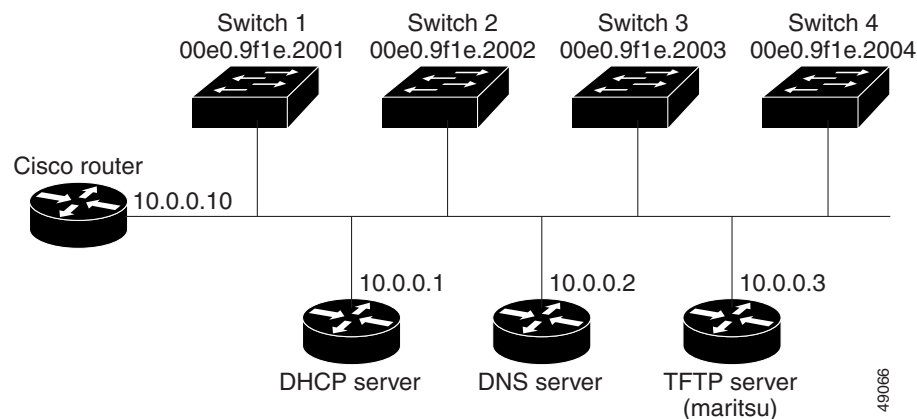


Table 3-2 shows the configuration of the reserved leases on either the DHCP server or the DHCP server feature running on your switch.

Table 3-2 DHCP Server Configuration

	Switch 1	Switch 2	Switch 3	Switch 4
Binding key (hardware address)	00e0.9f1e.2001	00e0.9f1e.2002	00e0.9f1e.2003	00e0.9f1e.2004
IP address	10.0.0.21	10.0.0.22	10.0.0.23	10.0.0.24
Subnet mask	255.255.255.0	255.255.255.0	255.255.255.0	255.255.255.0
Router address	10.0.0.10	10.0.0.10	10.0.0.10	10.0.0.10
DNS server address	10.0.0.2	10.0.0.2	10.0.0.2	10.0.0.2
TFTP server name	maritsu or 10.0.0.3	maritsu or 10.0.0.3	maritsu or 10.0.0.3	maritsu or 10.0.0.3
Boot filename (configuration file) (optional)	switch1-confg	switch2-confg	switch3-confg	switch4-confg
Host name (optional)	switch1	switch2	switch3	switch4

DNS Server Configuration

The DNS server maps the TFTP server name *maritsu* to IP address 10.0.0.3.

TFTP Server Configuration (on UNIX)

The TFTP server base directory is set to `/tftpserver/work/`. This directory contains the `network-config` file used in the two-file read method. This file contains the host name that you plan to assign to the switch based on its IP address. The base directory also contains a configuration file for each switch (*switch1-config*, *switch2-config*, and so forth) as shown in the following display:

```
prompt> cd /tftpserver/work/
prompt> ls
network-config
switch1-config
switch2-config
switch3-config
switch4-config
prompt> cat network-config
ip host switch1 10.0.0.21
ip host switch2 10.0.0.22
ip host switch3 10.0.0.23
ip host switch4 10.0.0.24
```

DHCP Client Configuration

No configuration file is present on Switch 1 through Switch 4.

Configuration Explanation

In [Figure 3-3](#), Switch 1 reads its configuration file as follows:

- Switch 1 obtains its IP address 10.0.0.21 from the DHCP server.
- If no configuration filename is given in the DHCP server reply, Switch 1 reads the `network-config` file from the base directory of the TFTP server.
- Switch 1 adds the contents of the `network-config` file to its host table.
- Switch 1 reads its host table by indexing its IP address 10.0.0.21 to its host name (switch1).
- Switch 1 reads the configuration file that corresponds to its host name; for example, it reads *switch1-config* from the TFTP server.

Switches 2 through 4 retrieve their configuration files and IP addresses in the same way.

Configuring the Switch

The following sections describe how to configure your switch:

- [Using Configuration Mode to Configure Your Switch, page 3-9](#)
- [Verifying the Running Configuration Settings, page 3-9](#)
- [Saving the Running Configuration Settings to Your Start-Up File, page 3-10](#)
- [Reviewing the Configuration in NVRAM, page 3-10](#)
- [Configuring a Default Gateway, page 3-11](#)
- [Configuring a Static Route, page 3-11](#)

Using Configuration Mode to Configure Your Switch

To configure your switch from configuration mode, follow these steps:

- Step 1** Connect a console terminal to the console interface of your supervisor engine.
- Step 2** After a few seconds, you see the user EXEC prompt (**Switch>**). Now, you may want to enter privileged EXEC mode, also known as enable mode. Type **enable** to enter enable mode:

```
Switch> enable
```



Note You must be in enable mode to make configuration changes.

The prompt changes to the enable prompt (**#**):

```
Switch#
```

- Step 3** At the enable prompt (**#**), enter the **configure terminal** command to enter global configuration mode:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#
```

- Step 4** At the global configuration mode prompt, enter the **interface type slot/interface** command to enter interface configuration mode:

```
Switch(config)# interface fastethernet 5/1
Switch(config-if)#
```

- Step 5** In either of these configuration modes, enter changes to the switch configuration.
- Step 6** Enter the **end** command to exit configuration mode.
- Step 7** Save your settings. See the [“Saving the Running Configuration Settings to Your Start-Up File”](#) section on page 3-10.

Your switch is now minimally configured and can boot with the configuration you entered. To see a list of the configuration commands, enter **?** at the prompt or press the **help** key in configuration mode.

Verifying the Running Configuration Settings

To verify the configuration settings you entered or the changes you made, enter the **show running-config** command at the enable prompt (**#**), as shown in this example:

```
Switch# show running-config
Building configuration...

Current configuration:
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Switch
```

```
<...output truncated...>

!
line con 0
  transport input none
line vty 0 4
  exec-timeout 0 0
  password lab
  login
  transport input lat pad dsipcon mop telnet rlogin udptn nasi
!
end
Switch#
```

Saving the Running Configuration Settings to Your Start-Up File



Caution

This command saves the configuration settings that you created in configuration mode. If you fail to do this step, your configuration is lost the next time you reload the system.

To store the configuration, changes to the configuration, or changes to the startup configuration in NVRAM, enter the **copy running-config startup-config** command at the enable prompt (#), as follows:

```
Switch# copy running-config startup-config
```

Reviewing the Configuration in NVRAM

To display information stored in NVRAM, enter the **show startup-config EXEC** command.

The following example shows a typical system configuration:

```
Switch# show startup-config
Using 1579 out of 491500 bytes, uncompressed size = 7372 bytes
Uncompressed configuration from 1579 bytes to 7372 bytes
!
version 12.1
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service compress-config
!
hostname Switch
!
!
ip subnet-zero
!
!
!
interface GigabitEthernet1/1
  no snmp trap link-status
!
interface GigabitEthernet1/2
  no snmp trap link-status
!--More--

<...output truncated...>
```



```

!
line con 0
  exec-timeout 0 0
  transport input none
line vty 0 4
  exec-timeout 0 0
  password lab
  login
  transport input lat pad dsipcon mop telnet rlogin udptn nasi
!
end

Switch#

```

Configuring a Default Gateway



Note

The switch uses the default gateway only when it is not configured with a routing protocol.

Configure a default gateway to send data to subnets other than its own when the switch is not configured with a routing protocol. The default gateway must be the IP address of an interface on a router that is directly connected to the switch.

To configure a default gateway, perform this task:

	Command	Purpose
Step 1	Switch(config)# ip default-gateway <i>IP-address</i>	Configures a default gateway.
Step 2	Switch# show ip route	Verifies that the default gateway is correctly displayed in the IP routing table.

This example shows how to configure a default gateway and how to verify the configuration:

```

Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# ip default-gateway 172.20.52.35
Switch(config)# end
3d17h: %SYS-5-CONFIG_I: Configured from console by console
Switch# show ip route
Default gateway is 172.20.52.35

Host                Gateway                Last Use    Total Uses  Interface
ICMP redirect cache is empty
Switch#

```

Configuring a Static Route

If your Telnet station or SNMP network management workstation is on a different network from your switch and a routing protocol has not been configured, you might need to add a static routing table entry for the network where your end station is located.

To configure a static route, perform this task:

	Command	Purpose
Step 1	Switch(config)# ip route <i>dest_IP_address mask {forwarding_IP vlan vlan_ID}</i>	Configures a static route to the remote network.
Step 2	Switch# show running-config	Verifies that the static route is displayed correctly.

This example shows how to use the **ip route** command to configure a static route to a workstation at IP address 171.10.5.10 on the switch with a subnet mask and IP address 172.20.3.35 of the forwarding router:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# ip route 171.10.5.10 255.255.255.255 172.20.3.35
Switch(config)# end
Switch#
```

This example shows how to use the **show running-config** command to confirm the configuration of the static route:

```
Switch# show running-config
Building configuration...
.
<...output truncated...>
.
ip default-gateway 172.20.52.35
ip classless
ip route 171.10.5.10 255.255.255.255 172.20.3.35
no ip http server
!
line con 0
  transport input none
line vty 0 4
  exec-timeout 0 0
  password lab
  login
  transport input lat pad dsipcon mop telnet rlogin udptn nasi
!
end

Switch#
```

This example shows how to use the **ip route** command to configure the static route IP address 171.20.5.3 with subnet mask and connected over VLAN 1 to a workstation on the switch:

```
Switch# configure terminal
Switch(config)# ip route 171.20.5.3 255.255.255.255 vlan 1
Switch(config)# end
Switch#
```

This example shows how to use the **show running-config** command to confirm the configuration of the static route:

```
Switch# show running-config
Building configuration...
.
<...output truncated...>
```

```
.
ip default-gateway 172.20.52.35
ip classless
ip route 171.20.5.3 255.255.255.255 Vlan1
no ip http server
!
!
x25 host z
!
line con 0
  transport input none
line vty 0 4
  exec-timeout 0 0
  password lab
  login
  transport input lat pad dsipcon mop telnet rlogin udptn nasi
!
end

Switch#
```

Controlling Access to Privileged EXEC Commands

The procedures in these sections let you control access to the system configuration file and privileged EXEC commands:

- [Setting or Changing a Static enable Password, page 3-13](#)
- [Using the enable password and enable secret Commands, page 3-14](#)
- [Setting or Changing a Privileged Password, page 3-14](#)
- [Controlling Switch Access with TACACS+, page 3-15](#)
- [Encrypting Passwords, page 3-22](#)
- [Configuring Multiple Privilege Levels, page 3-23](#)

Setting or Changing a Static enable Password

To set or change a static password that controls access to the enable mode, enter this command:

Command	Purpose
Switch(config)# enable password <i>password</i>	Sets a new password or changes an existing password for the privileged EXEC mode.

This example shows how to configure an enable password as lab:

```
Switch# configure terminal
Switch(config)# enable password lab
Switch(config)#
```

For instructions on how to display the password or access level configuration, see the [“Displaying the Password, Access Level, and Privilege Level Configuration”](#) section on page 3-24.

Using the enable password and enable secret Commands

To provide an additional layer of security, particularly for passwords that cross the network or that are stored on a TFTP server, use either the **enable password** or **enable secret** command. Both commands configure an encrypted password that you must enter to access the enable mode (the default) or any other privilege level that you specify.

We recommend that you use the **enable secret** command.

If you configure the **enable secret** command, it takes precedence over the **enable password** command; the two commands cannot be in effect simultaneously.

To configure the switch to require an enable password, enter one of these commands:

Command	Purpose
Switch(config)# enable password [level <i>level</i>] { <i>password</i> <i>encryption-type</i> <i>encrypted-password</i> }	Establishes a password for the privileged EXEC mode.
Switch(config)# enable secret [level <i>level</i>] { <i>password</i> <i>encryption-type</i> <i>encrypted-password</i> }	Specifies a secret password that is saved using a nonreversible encryption method. (If enable password and enable secret commands are both set, users must enter the enable secret password.)

When you enter either of these password commands with the **level** option, you define a password for a specific privilege level. After you specify the level and set a password, give the password only to users who need to have access at this level. Use the **privilege level** configuration command to specify commands accessible at various levels.

If you enable the **service password-encryption** command, the password you enter is encrypted. When you display the password with the **more system:running-config** command, the password displays the password in encrypted form.

If you specify an encryption type, you must provide an encrypted password—an encrypted password you copy from another switch configuration.



Note

You cannot recover a lost encrypted password. You must clear NVRAM and set a new password. See the [“Recovering a Lost Enable Password”](#) section on page 3-25 for more information.

For information on how to display the password or access level configuration, see the [“Displaying the Password, Access Level, and Privilege Level Configuration”](#) section on page 3-24.

Setting or Changing a Privileged Password

To set or change a privileged password, enter this command:

Command	Purpose
Switch(config-line)# password <i>password</i>	Sets a new password or changes an existing password for the privileged level.

For information on how to display the password or access level configuration, see the [“Displaying the Password, Access Level, and Privilege Level Configuration”](#) section on page 3-24.

Controlling Switch Access with TACACS+

This section describes how to enable and configure TACACS+, which provides detailed accounting information and flexible administrative control over authentication and authorization processes. TACACS+ is facilitated through authentication, authorization, accounting (AAA) and can be enabled only through AAA commands.



Note

For complete syntax and usage information for the commands used in this section, see the *Cisco IOS Security Command Reference, Release 12.2*.

This section contains the following configuration information:

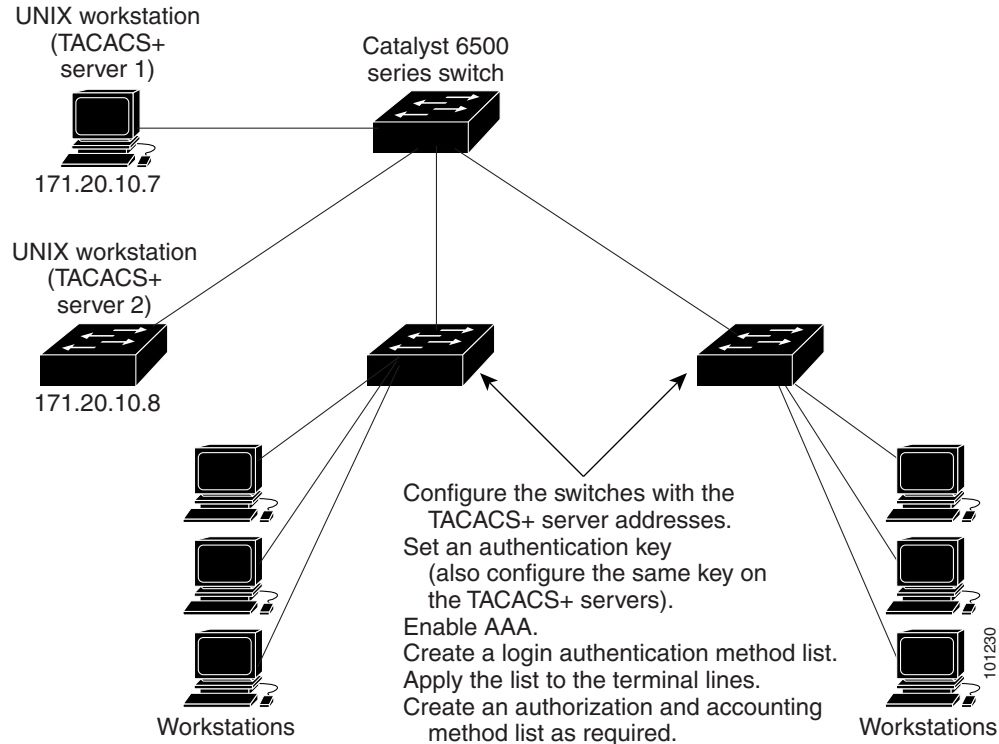
- [Understanding TACACS+, page 3-15](#)
- [TACACS+ Operation, page 3-17](#)
- [Configuring TACACS+, page 3-17](#)
- [Displaying the TACACS+ Configuration, page 3-22](#)

Understanding TACACS+

TACACS+ is a security application that provides centralized validation of users attempting to gain access to your switch. TACACS+ services are maintained in a database on a TACACS+ daemon typically running on a UNIX or Windows NT workstation. You should have access to and should configure a TACACS+ server before configuring TACACS+ features on your switch.

TACACS+ provides for separate and modular AAA facilities. TACACS+ allows for a single access control server (the TACACS+ daemon) to provide each service—authentication, authorization, and accounting—independently. Each service can be locked into its own database to take advantage of other services available on that server or on the network, depending on the capabilities of the daemon.

The goal of TACACS+ is to provide a method for managing multiple network access points from a single management service. Your switch can be a network access server along with other Cisco routers and access servers. A network access server provides connections to a single user, to a network or subnetwork, and to interconnected networks as shown in [Figure 3-4](#).

Figure 3-4 Typical TACACS+ Network Configuration

TACACS+ administered through the AAA security services can provide these services:

- **Authentication**—Provides complete control of authentication through login and password dialog, challenge and response, and messaging support.
The authentication facility can conduct a dialog with the user (such as, after a username and password are provided, to challenge a user with several questions such as home address, mother's maiden name, service type, and social security number). The TACACS+ authentication service can also send messages to user screens. For example, a message could notify users that their passwords must be changed because of the company's password aging policy.
- **Authorization**—Provides strict control over user capabilities for the duration of the user's session, including but not limited to setting autocommands, access control, session duration, or protocol support. You can also enforce restrictions on the commands a user can execute with the TACACS+ authorization feature.
- **Accounting**—Collects and sends information used for billing, auditing, and reporting to the TACACS+ daemon. Network managers can use the accounting facility to track user activity for a security audit or to provide information for user billing. Accounting records include user identities, start and stop times, executed commands (such as PPP), number of packets, and number of bytes.

The TACACS+ protocol provides authentication between the switch and the TACACS+ daemon, and it ensures confidentiality because all protocol exchanges between the switch and the TACACS+ daemon are encrypted.

You need a system running the TACACS+ daemon software to use TACACS+ on your switch.

TACACS+ Operation

When a user attempts a simple ASCII login by authenticating to a switch using TACACS+, this process occurs:

1. When the connection is established, the switch contacts the TACACS+ daemon to obtain a username prompt, which is then displayed to the user. The user enters a username, and the switch then contacts the TACACS+ daemon to obtain a password prompt. The switch displays the password prompt to the user, the user enters a password, and the password is then sent to the TACACS+ daemon.

TACACS+ allows a conversation between the daemon and the user until the daemon receives enough information to authenticate the user. The daemon prompts for a username and password combination, but can include other items such as the user's mother's maiden name.

2. The switch eventually receives one of these responses from the TACACS+ daemon:
 - **ACCEPT**—The user is authenticated and service can begin. If the switch is configured to require authorization, authorization begins at this time.
 - **REJECT**—The user is not authenticated. The user can be denied access or is prompted to retry the login sequence, depending on the TACACS+ daemon.
 - **ERROR**—An error occurred at some time during authentication with the daemon or in the network connection between the daemon and the switch. If an **ERROR** response is received, the switch typically tries to use an alternative method for authenticating the user.
 - **CONTINUE**—The user is prompted for additional authentication information.

After authentication, the user undergoes an additional authorization phase if authorization has been enabled on the switch. Users must first successfully complete TACACS+ authentication before proceeding to TACACS+ authorization.

3. If TACACS+ authorization is required, the TACACS+ daemon is again contacted, and it returns an **ACCEPT** or **REJECT** authorization response. If an **ACCEPT** response is returned, the response contains data in the form of attributes that direct the **EXEC** or **NETWORK** session for that user and the services that the user can access:
 - Telnet, Secure Shell (SSH), rlogin, or privileged **EXEC** services
 - Connection parameters, including the host or client IP address, access list, and user timeouts

Configuring TACACS+

This section describes how to configure your switch to support TACACS+. At a minimum, you must identify the host or hosts maintaining the TACACS+ daemon and define the method lists for TACACS+ authentication. You can optionally define method lists for TACACS+ authorization and accounting. A method list defines the sequence and methods used to authenticate, to authorize, or to keep accounts on a user. Use method lists to designate one or more security protocols, ensuring a backup system if the initial method fails. The software uses the first method listed to authenticate, to authorize, or to keep accounts on users; if that method does not respond, the software selects the next method in the list. This process continues until there is successful communication with a listed method or the method list is exhausted.



Note

Beginning with Cisco IOS XE Release 3.11.3aE, the legacy command **tacacs-server** is deprecated. Use the **tacacs server** command if the software running on your device is Cisco IOS XE Release 3.11.3aE or later releases.

This section contains the following configuration information:

- [Default TACACS+ Configuration, page 3-18](#)
- [Identifying the TACACS+ Server Host and Setting the Authentication Key, page 3-18](#)
- [Configuring TACACS+ Login Authentication, page 3-19](#)
- [Configuring TACACS+ Authorization for Privileged EXEC Access and Network Services, page 3-21](#)
- [Starting TACACS+ Accounting, page 3-21](#)

Default TACACS+ Configuration

TACACS+ and AAA are disabled by default.

To prevent a lapse in security, you cannot configure TACACS+ through a network management application. When enabled, TACACS+ can authenticate users accessing the switch through the CLI.



Note

Although TACACS+ configuration is performed through the CLI, the TACACS+ server authenticates HTTP connections that have been configured with a privilege level of 15.

Identifying the TACACS+ Server Host and Setting the Authentication Key

You can configure the switch to use a single server or AAA server groups in order to group existing server hosts for authentication. You can group servers to select a subset of the configured server hosts and use them for a particular service. The server group is used with a global server-host list and contains the list of IP addresses of the selected server hosts.



Note

Beginning with Cisco IOS XE Release 3.11.3aE, the legacy command **tacacs-server** is deprecated. Use the **tacacs server** command if the software running on your device is Cisco IOS XE Release 3.11.3aE or later releases.

To identify the IP host or host maintaining TACACS+ server and optionally set the encryption key, perform this task, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	configure terminal	Enters global configuration mode.
Step 2	tacacs server <i>servername</i>	Identifies the IP host or hosts maintaining a TACACS+ server. Enter this command multiple times to create a list of preferred hosts. The software searches for hosts in the order in which you specify them. For <i>servername</i> , specify the name of the server.
Step 3	address { <i>ipv4</i> <i>ipv6</i> } <i>ipaaddress</i>	Configures the IP address for the TACACS server.
Step 4	aaa new-model	Enables AAA.
Step 5	aaa group server tacacs+ <i>group-name</i>	(Optional) Defines the AAA server-group with a group name. This command puts the switch in a server group subconfiguration mode.
Step 6	server name <i>servername</i>	(Optional) Associates a particular TACACS+ server with the defined server group. Repeat this step for each TACACS+ server in the AAA server group. Each server in the group must be previously defined in Step 2.

	Command	Purpose
Step 7	<code>end</code>	Returns to privileged EXEC mode.
Step 8	<code>show tacacs</code>	Verifies your entries.
Step 9	<code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

To remove the specified TACACS+ server name or address, use the **no tacacs server *servername*** global configuration command. To remove a server group from the configuration list, use the **no aaa group server tacacs+ *group-name*** global configuration command. To remove the IP address of a TACACS+ server, use the **no server *ip-address*** server group subconfiguration command.

Configuring TACACS+ Login Authentication

To configure AAA authentication, define a named list of authentication methods and then apply that list to various ports. The method list defines the types of authentication you intend to perform and the sequence in which you intend to perform them; you must apply the list to a specific port before you can perform any of the defined authentication methods. The only exception is the default method list (which, by coincidence, is named *default*). The default method list is automatically applied to all ports except those that have a named method list explicitly defined. A defined method list overrides the default method list.

A method list describes the sequence and authentication methods that must be queried to authenticate a user. You can designate one or more security protocols for authentication, ensuring a backup system for authentication in case the initial method fails. The software uses the first method listed to authenticate users; if that method fails to respond, the software selects the next authentication method in the method list. This process continues until there is successful communication with a listed authentication method or until all defined methods are exhausted. If authentication fails at any point in this cycle—meaning that the security server or local username database responds by denying the user access—the authentication process stops, and no other authentication methods are attempted.

To configure login authentication, perform this task, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>aaa new-model</code>	Enables AAA.

	Command	Purpose
Step 3	<code>aaa authentication login {default list-name} method1 [method2...]</code>	<p>Creates a login authentication method list.</p> <ul style="list-style-type: none"> To create a default list that is used when a named list is <i>not</i> specified in the login authentication command, use the default keyword followed by the methods that you plan to use in default situations. The default method list is automatically applied to all ports. For <i>list-name</i>, specify a character string to name the list you are creating. For <i>method1...</i>, specify the actual method the authentication algorithm tries. The additional methods of authentication are used only if the previous method returns an error, not if it fails. <p>Select one of these methods:</p> <ul style="list-style-type: none"> enable—Use the enable password for authentication. Before you can use this authentication method, you must define an enable password by using the enable password global configuration command. group tacacs+—Uses TACACS+ authentication. Before you can use this authentication method, you must configure the TACACS+ server. For more information, see the “Identifying the TACACS+ Server Host and Setting the Authentication Key” section on page 3-18. line—Use the line password for authentication. Before you can use this authentication method, you must define a line password. Use the password password line configuration command. local—Use the local username database for authentication. You must enter username information in the database. Use the username password global configuration command. local-case—Use a case-sensitive local username database for authentication. You must enter username information in the database by using the username name password global configuration command. none—Do not use any authentication for login.
Step 4	<code>line [console tty vty] line-number [ending-line-number]</code>	Enters line configuration mode, and configures the lines to which you want to apply the authentication list.
Step 5	<code>login authentication {default list-name}</code>	<p>Applies the authentication list to a line or set of lines.</p> <ul style="list-style-type: none"> If you specify default, use the default list created with the aaa authentication login command. For <i>list-name</i>, specify the list created with the aaa authentication login command.
Step 6	<code>end</code>	Returns to privileged EXEC mode.
Step 7	<code>show running-config</code>	Verifies your entries.
Step 8	<code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

To disable AAA, use the **no aaa new-model** global configuration command. To disable AAA authentication, use the **no aaa authentication login {default | list-name} method1 [method2...]** global configuration command. To either disable TACACS+ authentication for logins or to return to the default value, use the **no login authentication {default | list-name}** line configuration command.

Configuring TACACS+ Authorization for Privileged EXEC Access and Network Services

AAA authorization limits the services available to a user. When AAA authorization is enabled, the switch uses information retrieved from the user's profile, which is located either in the local user database or on the security server, to configure the user's session. The user is granted access to a requested service only if the information in the user profile allows it.

To set parameters that restrict a user's network access to privileged EXEC mode, use the **aaa authorization** global configuration command with the **tacacs+** keyword.

The **aaa authorization exec tacacs+ local** command sets these authorization parameters:

- Use TACACS+ for privileged EXEC access authorization if authentication was performed by using TACACS+.
- Use the local database if authentication was not performed by using TACACS+.



Note

Authorization is bypassed for authenticated users who log in through the CLI even if authorization has been configured.

To specify TACACS+ authorization for privileged EXEC access and network services, perform this task, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>aaa authorization network tacacs+</code>	Configures the switch for user TACACS+ authorization for all network-related service requests.
Step 3	<code>aaa authorization exec tacacs+</code>	Configures the switch for user TACACS+ authorization if the user has privileged EXEC access. The exec keyword might return user profile information (such as autocommand information).
Step 4	<code>end</code>	Returns to privileged EXEC mode.
Step 5	<code>show running-config</code>	Verifies your entries.
Step 6	<code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

To disable authorization, use the **no aaa authorization {network | exec} method1** global configuration command.

Starting TACACS+ Accounting

The AAA accounting feature tracks the services that users are accessing and the amount of network resources that they are consuming. When AAA accounting is enabled, the switch reports user activity to the TACACS+ security server in the form of accounting records. Each accounting record contains accounting attribute-value (AV) pairs and is stored on the security server. This data can then be analyzed for network management, client billing, or auditing.

To enable TACACS+ accounting for each Cisco IOS privilege level and for network services, perform this task, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>aaa accounting network start-stop tacacs+</code>	Enables TACACS+ accounting for all network-related service requests.
Step 3	<code>aaa accounting exec start-stop tacacs+</code>	Enables TACACS+ accounting to send a start-record accounting notice at the beginning of a privileged EXEC process and a stop-record at the end.
Step 4	<code>end</code>	Returns to privileged EXEC mode.
Step 5	<code>show running-config</code>	Verifies your entries.
Step 6	<code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

To disable accounting, use the **no aaa accounting {network | exec} {start-stop} method1...** global configuration command.

Displaying the TACACS+ Configuration

To display TACACS+ server statistics, use the **show tacacs** privileged EXEC command.

Encrypting Passwords

Because protocol analyzers can examine packets (and read passwords), you can increase access security by configuring the Cisco IOS software to encrypt passwords. Encryption prevents the password from being readable in the configuration file.

To configure the Cisco IOS software to encrypt passwords, enter this command:

Command	Purpose
Switch(config)# service password-encryption	Encrypts a password.

Encryption occurs when the current configuration is written or when a password is configured. Password encryption is applied to all passwords, including authentication key passwords, the privileged command password, console and virtual terminal line access passwords, and Border Gateway Protocol (BGP) neighbor passwords. The **service password-encryption** command keeps unauthorized individuals from viewing your password in your configuration file.



Caution

The **service password-encryption** command does not provide a high-level of network security. If you use this command, you should also take additional network security measures.

Although you cannot recover a lost encrypted password (that is, you cannot get the original password back), you can regain control of the switch after having lost or forgotten the encrypted password. See the [“Recovering a Lost Enable Password”](#) section on page 3-25 for more information.

For information on how to display the password or access level configuration, see the [“Displaying the Password, Access Level, and Privilege Level Configuration”](#) section on page 3-24.

Configuring Multiple Privilege Levels

By default, Cisco IOS software has two modes of password security: user EXEC mode and privileged EXEC mode. You can configure up to 16 hierarchical levels of commands for each mode. By configuring multiple passwords, you can allow different sets of users to have access to specified commands.

For example, if you want many users to have access to the **clear line** command, you can assign it level 2 security and distribute the level 2 password to more users. If you want more restricted access to the **configure** command, you can assign it level 3 security and distribute that password to fewer users.

The procedures in the following sections describe how to configure additional levels of security:

- [Setting the Privilege Level for a Command, page 3-23](#)
- [Changing the Default Privilege Level for Lines, page 3-23](#)
- [Logging In to a Privilege Level, page 3-24](#)
- [Exiting a Privilege Level, page 3-24](#)
- [Displaying the Password, Access Level, and Privilege Level Configuration, page 3-24](#)

Setting the Privilege Level for a Command

To set the privilege level for a command, perform this task:

	Command	Purpose
Step 1	Switch(config)# privilege mode level level <i>command</i>	Sets the privilege level for a command.
Step 2	Switch(config)# enable password level level <i>[encryption-type] password</i>	Specifies the enable password for a privilege level.

For information on how to display the password or access level configuration, see the [“Displaying the Password, Access Level, and Privilege Level Configuration”](#) section on page 3-24.

Changing the Default Privilege Level for Lines

To change the default privilege level for a given line or a group of lines, perform this task:

Command	Purpose
Switch(config-line)# privilege level level	Changes the default privilege level for the line.

For information on how to display the password or access level configuration, see the [“Displaying the Password, Access Level, and Privilege Level Configuration”](#) section on page 3-24.

Logging In to a Privilege Level

To log in at a specified privilege level, enter this command:

Command	Purpose
Switch# enable <i>level</i>	Logs in to a specified privilege level.

Exiting a Privilege Level

To exit to a specified privilege level, enter this command:

Command	Purpose
Switch# disable <i>level</i>	Exits to a specified privilege level.

Displaying the Password, Access Level, and Privilege Level Configuration

To display detailed password information, perform this task:

	Command	Purpose
Step 1	Switch# show running-config	Displays the password and access level configuration.
Step 2	Switch# show privilege	Shows the privilege level configuration.

This example shows how to display the password and access level configuration:

```
Switch# show running-config
Building configuration...

Current configuration:
!
version 12.0
service timestamps debug datetime localtime
service timestamps log datetime localtime
no service password-encryption
!
hostname Switch
!
boot system flash sup-bootflash
enable password lab
!
<...output truncated...>
```

This example shows how to display the privilege level configuration:

```
Switch# show privilege
Current privilege level is 15
Switch#
```

Recovering a Lost Enable Password

**Note**

For more information on the configuration register which is preconfigured in NVRAM, see [“Configuring the Software Configuration Register”](#) section on page 3-26.

To recover a lost enable password, follow these steps:

-
- | | |
|---------------|--|
| Step 1 | Connect to the console interface. |
| Step 2 | Stop the boot sequence and enter ROM monitor by pressing Ctrl-C during the first 5 seconds of bootup. |
| Step 3 | Configure the switch to boot-up without reading the configuration memory (NVRAM). |
| Step 4 | Reboot the system. |
| Step 5 | Access enable mode (this can be done without a password if a password has not been configured). |
| Step 6 | View or change the password, or erase the configuration. |
| Step 7 | Reconfigure the switch to boot-up and read the NVRAM as it normally does. |
| Step 8 | Reboot the system. |
-

Modifying the Supervisor Engine Startup Configuration

These sections describe how the startup configuration on the supervisor engine works and how to modify the BOOT variable and the configuration register:

- [Understanding the Supervisor Engine Boot Configuration, page 3-25](#)
- [Configuring the Software Configuration Register, page 3-26](#)
- [Specifying the Startup System Image, page 3-31](#)
- [Controlling Environment Variables, page 3-32](#)

Understanding the Supervisor Engine Boot Configuration

The supervisor engine boot process involves two software images: ROM monitor and supervisor engine software. When the switch is booted or reset, the ROMMON code is executed. Depending on the NVRAM configuration, the supervisor engine either stays in ROMMON mode or loads the supervisor engine software.

Two user-configurable parameters determine how the switch boots: the configuration register and the BOOT environment variable. The configuration register is described in the [“Modifying the Boot Field and Using the boot Command”](#) section on page 3-28. The BOOT environment variable is described in the [“Specifying the Startup System Image”](#) section on page 3-31.

Understanding the ROM Monitor

The ROM monitor (ROMMON) is invoked at switch bootup, reset, or when a fatal exception occurs. The switch enters ROMMON mode if the switch does not find a valid software image, if the NVRAM configuration is corrupted, or if the configuration register is set to enter ROMMON mode. From ROMMON mode, you can manually load a software image from bootflash or a flash disk, or you can boot up from the management interface. ROMMON mode loads a primary image from which you can configure a secondary image to boot up from a specified source either locally or through the network using the BOOTLDR environment variable. This variable is described in the “[Switch#](#)” section on [page 3-33](#).

You can also enter ROMMON mode by restarting the switch and then pressing **Ctrl-C** during the first five seconds of startup. If you are connected through a terminal server, you can escape to the Telnet prompt and enter the **send break** command to enter ROMMON mode.



Note

Ctrl-C is always enabled for five seconds after you reboot the switch, regardless of whether the configuration-register setting has **Ctrl-C** disabled.

The ROM monitor has these features:

- Power-on confidence test
- Hardware initialization
- Boot capability (manual bootup and autoboot)
- File system (read-only while in ROMMON)

Configuring the Software Configuration Register

The switch uses a 16-bit software configuration register, which allows you to set specific system parameters. Settings for the software configuration register are preconfigured in NVRAM.

Here are some reasons why you might want to change the software configuration register settings:

- To select a boot source and default boot filename
- To control broadcast addresses
- To set the console terminal baud rate
- To load operating software from flash memory
- To recover a lost password
- To manually boot the system using the **boot** command at the bootstrap program prompt
- To force an automatic bootup from the system bootstrap software (boot image) or from a default system image in onboard flash memory, and read any **boot system** commands that are stored in the configuration file in NVRAM



Caution

To avoid possibly halting the switch, remember that valid configuration register settings might be combinations of settings and not just the individual settings listed in [Table 3-3](#). For example, the factory default value of 0x2101 is a combination of settings.

Table 3-3 lists the meaning of each of the software configuration memory bits. Table 3-4 defines the *boot* field.

Table 3-3 Software Configuration Register Bits

Bit Number ¹	Hexadecimal	Meaning
00 to 03	0x0000 to 0x000F	Boot field (see Table 3-4)
04	0x0010	Unused
05	0x0020	Bit two of console line speed
06	0x0040	Causes system software to ignore NVRAM contents
07	0x0080	OEM ² bit enabled
08	0x0100	Unused
09	0x0200	Unused
10	0x0400	IP broadcast with all zeros
11 to 12	0x0800 to 0x1000	Bits one and zero of Console line speed (default is 9600 baud)
13	0x2000	Loads ROM monitor after netboot fails
14	0x4000	IP broadcasts do not have network numbers

1. The factory default value for the configuration register is 0x2101. This value is a combination of the following: binary bit 13, bit 8 = 0x0100 and binary bits 00 through 03 = 0x0001. See Table 3-4.
2. OEM = original equipment manufacturer.

Table 3-4 Explanation of Boot Field (Configuration Register Bits 00 to 03)

Boot Field	Meaning
00	Stays at the system bootstrap prompt (does not autoboot).
01	Boots the first file in onboard flash memory.
02	Auto boots using image(s) specified by the BOOT environment variable. If more than one image is specified, the switch attempts to boot the first image specified in the BOOT variable. As long as the switch can successfully boot from this image, the same image is used on a reboot. If the switch fails to boot from the image specified in the BOOT variable, the switch tries to boot from the next image listed in the BOOT variable. If the end of the BOOT variable is reached without the switch booting successfully, the switch attempts the boot from the beginning of the BOOT variable. The autoboot continues until the switch successfully boots from one of the images specified in the BOOT variable.
03	Unsupported or unused. If the boot field is set to this value, a warning message is displayed, indicating that you must check and reenter the correct value.
04 ¹	Auto boots the image that is retrieved from a network-based source, that is, a DHCP server. Depending on how you configure the DHCP boot file field, the system then retrieves the files from the specified remote (HTTP or TFTP) server.

Table 3-4 Explanation of Boot Field (Configuration Register Bits 00 to 03) (continued)

Boot Field	Meaning
05 ¹	Auto boots the image that is retrieved from a network-based source, that is, a DHCP server. Depending on how you configure the DHCP boot file field, the system then retrieves the files from the specified remote (HTTP or TFTP) server. If this fails, the system resorts to the method specified in boot field 01.
06 ¹	Auto boots the image that is retrieved from a network-based source, that is, a DHCP server. Depending on how you configure the DHCP boot file field, the system then retrieves the files from the specified remote (HTTP or TFTP) server. If this fails, the system resorts to the method specified in boot field 02.

1. Note: Requires ROMMON version 15.0(1r)SG14 on Catalyst 4500-X Series Switches, and ROMMON version 15.1(1r)SG8 on Catalyst 4500-E Series Switches.

Modifying the Boot Field and Using the boot Command

The configuration register boot field determines whether the switch loads an operating system image and, if so, where it obtains this system image. The following sections describe how to use and set the configuration register boot field and the procedures you must perform to modify the configuration register boot field. In ROMMON, to modify the configuration register and change boot settings, use the **confreg** command.

Bits 0 through 3 of the software configuration register contain the boot field.



Note

The factory default configuration register setting for systems and spares is 0x2101. However, the recommended value is 0x0102.

When the boot field is set to either 00 or 01 (0-0-0-0 or 0-0-0-1), the system ignores any boot instructions in the system configuration file and the following occurs:

- When the boot field is set to 00, you must boot up the operating system manually by entering the **boot** command at the system bootstrap or ROMMON prompt.
- When the boot field is set to 01, the system boots the first image in the bootflash single in-line memory module (SIMM).
- When the entire boot field equals a value between 0-0-1-0 and 1-1-1-1, the switch loads the system image specified by **boot system** commands in the startup configuration file.



Caution

If you set bootfield to a value between 0-0-1-0 and 1-1-1-1, you must specify a value in the **boot system** command, else the switch cannot boot up and remains in ROMMON.

You can enter the **boot** command only or enter the command and include additional boot instructions, such as the name of a file stored in flash memory, or a file that you specify for booting from a network server. If you use the **boot** command without specifying a file or any other boot instructions, the system boots from the default flash image (the first image in onboard flash memory). Otherwise, you can instruct the system to boot up from a specific flash image (using the **boot system flash filename** command).

You can also use the **boot** command to boot up images stored in the compact flash cards located in slot 0 on the supervisor engine.

You must **reload** the switch for any changes to take effect.

Modifying the Boot Field

To modify the boot field from the software configuration register, perform this task:

	Command	Purpose
Step 1	Switch# show version	Determines the current configuration register setting.
Step 2	Switch# configure terminal	Enters configuration mode, and specify the terminal option.
Step 3	Switch(config)# config-register <i>value</i>	Modifies the existing configuration register setting to reflect the way you want the switch to load a system image.
Step 4	Switch(config)# end	Exits configuration mode.
Step 5	Switch# reload	Reboots the switch to make your changes take effect.

To modify the configuration register while the switch is running Cisco IOS software, perform this task:

	Command	Purpose
Step 1	Switch# enable	Enters the privileged EXEC mode. Enter the password if required.
Step 2	Switch# configure terminal	Enters configuration mode, and specifies the terminal option.
Step 3	Switch(config)# config-register 0x102	Sets the contents of the configuration register to the specified <i>value</i> , where <i>value</i> is a hexadecimal number preceded by 0x (see Table 3-3 on page 3-27).
Step 4	Switch(config)# end	Exits configuration mode. The new value settings are saved to memory; however, the new settings do not take effect until the system is rebooted.
Step 5	Switch# show version	Displays the configuration register value currently in effect. The value is displayed on the last line of the screen display, as shown in this sample output: Configuration register is 0x141 (will be 0x102 at next reload)
Step 6	Save your settings.	See the “Saving the Running Configuration Settings to Your Start-Up File” section on page 3-10 . Note that configuration register changes take effect only after the system reloads, such as when you enter a reload command from the console.
Step 7	Switch# reload	The new configuration register value takes effect with the next system boot up.

Modifying the Configuration Register Value for Wireless Mode

On Catalyst 4500E Series Switches with Supervisor Engine 9-E and 8-E, to boot the system in wireless mode, perform this task:

	Command	Purpose
Step 1	Switch# enable	Enters the privileged EXEC mode. Enter the password if required.
Step 2	Switch# configure terminal	Enters configuration mode, and specifies the terminal option.
Step 3	Switch(config)# boot system bootflash:packages.conf	Instructs the system to boot up from the specified flash image.
Step 4	Switch(config)# config-register 0x2102	Sets the contents of the configuration register to the specified value.
Step 5	Switch(config)# end	Exits configuration mode. The new value settings are saved to memory; however, the new settings do not take effect until the system is rebooted.

Verifying the Configuration Register Setting

Enter the **show version** EXEC command to verify the current configuration register setting. In ROMMON mode, enter the **show version** command to verify the configuration register setting.

To verify the configuration register setting for the switch, perform this task:

Command	Purpose
Switch# show version	Displays the configuration register setting.

In this example, the **show version** command indicates that the current configuration register is set so that the switch does not automatically load an operating system image. Instead, it enters ROMMON mode and waits for you to enter ROM monitor commands.

Supervisor Engine 6-E and Supervisor Engine 6L-E

```
Switch# show version
Cisco IOS Software, Catalyst 4500 L3 Switch Software (cat4500e-ENTSERVICES-M), Version
15.1(1)SG5.214, CISCO INTERNAL USE ONLY DEVTEST VERSION , synced to END_OF_FLO_ISP
Copyright (c) 1986-2012 by Cisco Systems, Inc.
Compiled Tue 17-Jan-12 23:07 by gsbuprod

ROM: 12.2(44r)SG(0.146)
Switch uptime is 1 minute
System returned to ROM by power-on
System image file is
"tftp://172.25.60.31/auto/gsg-sw/interim/flo_dsgs7/newest_image/ios/dev/cat4500e-entservic
es-mz"
Darkside Revision 4, Jawa Revision 20, Tatooine Revision 141, Forerunner Revision 1.83
```

```
cisco WS-C4503-E (MPC8548) processor (revision 6) with 1048576K bytes of memory.
Processor board ID SPE120301X8
MPC8548 CPU at 1.33GHz, Supervisor 6-E
Last reset from PowerUp
1 Virtual Ethernet interface
52 Gigabit Ethernet interfaces
2 Ten Gigabit Ethernet interfaces
511K bytes of non-volatile configuration memory.
```

Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, and 7-E

```
Switch# show version
Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3 Switch Software
(cat4500e-UNIVERSALK9-M), Version 03.03.00.SG5.
CISCO INTERNAL USE ONLY UNIVERSAL DEVELOPMENT K10 IOSD VERSION , synced to V150_5_20_SID
Copyright (c) 1986-2011 by Cisco Systems, Inc.
Compiled Wed 14-Dec-11 07:59 by gsbuprod

ROM: 15.0(1r)SG(0.326)
Switch uptime is 7 minutes
System returned to ROM by reload
System image file is
"tftp://172.25.60.31/auto/gsg-sw/interim/flo_gsbu8/newest_image/iosxe/dev/cat4500e-univers
alk9.b
Jawa Revision 7, Snowtrooper Revision 0x0.0x1C

Last reload reason: Reload command

...

License Information for 'WS-X45-SUP7-E'
  License Level: entservices   Type: Permanent
  Next reboot license Level: entservices

cisco WS-C4503-E (MPC8572) processor (revision 8) with 2097152K/20480K bytes of memory.
Processor board ID SPE134600QA
MPC8572 CPU at 1.5GHz, Supervisor 7
Last reset from Reload
1 Virtual Ethernet interface
96 Gigabit Ethernet interfaces
4 Ten Gigabit Ethernet interfaces
511K bytes of non-volatile configuration memory.

Configuration register is 0x40
```

Specifying the Startup System Image

You can enter multiple boot commands in the startup configuration file or in the BOOT environment variable to provide backup methods for loading a system image.

The BOOT environment variable is also described in the “Specify the Startup System Image in the Configuration File” section in the “Loading and Maintaining System Images and Microcode” chapter of the *Cisco IOS Configuration Fundamentals Configuration Guide*.

Use the following sections to configure your switch to boot from flash memory. Flash memory can be either single in-line memory modules (SIMMs) or flash disks. Check the appropriate hardware installation and maintenance guide for information about types of flash memory.

Flash Memory Features

Flash memory allows you to do the following:

- Remotely load multiple system software images through TFTP or RCP transfers (one transfer for each file loaded)
- Boot a switch manually or automatically from a system software image stored in flash memory (you can also boot directly from ROM)
- Copy the system image to flash memory using TFTP

- Boot the system from flash memory either automatically or manually
- Copy the flash memory image to a network server using TFTP or RCP

For more information on flash memory, see this URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst4500/hardware/configuration/notes/OL_2788.html

Security Precautions

Note the following security precaution when loading from flash memory:



Caution

You can only change the system image stored in flash memory from privileged EXEC level on the console terminal.

Configuring Flash Memory

To configure your switch to boot from flash memory, perform the following procedure. Refer to the appropriate hardware installation and maintenance publication for complete instructions on installing the hardware.

-
- Step 1** Copy a system image to flash memory using TFTP, FTP, or through a VRF interface.
- Refer to section “Copy Configuration Files from a Network Server to the Router” in chapter “Managing Configuration Files”, of the “Managing Configuration Files Configuration Guide” at the following URL:
- <http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/config-mgmt/configuration/xr-3e/config-mgmt-xr-3e-book/cm-config-files.html>.
- Configure the system to boot automatically from the desired file in flash memory.
- You might need to change the configuration register value. See the “[Modifying the Boot Field and Using the boot Command](#)” section on page 3-28, for more information on modifying the configuration register.
- Step 2** Save your configurations.
- Step 3** Power cycle and reboot your system to verify that all is working as expected.
-

Controlling Environment Variables

Although the ROM monitor controls environment variables, you can create, modify, or view them with certain commands. To create or modify the BOOT and BOOTLDR variables, use the **boot system** and **boot bootldr** global configuration commands, respectively. Refer to the “Specify the Startup System Image in the Configuration File” section in the “Loading and Maintaining System Images and Microcode” chapter of the *Configuration Fundamentals Configuration Guide* for details on setting the BOOT environment variable.

**Note**

When you use the **boot system** and **boot bootldr** global configuration commands, you affect only the running configuration. To save the configuration for future use, you must save the environment variable settings to your startup configuration, which places the information under ROM monitor control. Enter the **copy system:running-config nvram:startup-config** command to save the environment variables from your running configuration to your startup configuration.

You can view the contents of the BOOT and BOOTLDR variables using the **show bootvar** command. This command displays the settings for these variables as they exist in the startup configuration and in the running configuration if a running configuration setting differs from a startup configuration setting. This example shows how to check the BOOT and BOOTLDR variables on the switch:

```
Switch# show bootvar
BOOTLDR variable = bootflash:cat4000-is-mz,1;
Configuration register is 0x0
Switch#
```

Replacing and Rolling-Back Configuration

For detailed information about this feature, see the “Configuration Replace and Configuration Rollback” chapter of the [Managing Configuration Files](#) feature guide on cisco.com.

The following restrictions pertain to the use of this feature on Catalyst 4500 Series Switches:

- You cannot use this feature to convert a VSS system to a non-VSS (standalone) system, or the other way around. The configuration that is replaced and the new configuration that is replacing the existing, must both apply to the same kind of system - whether VSS or non-VSS.
- Do not use the **configure replace** command to replace wireless configuration with non-wireless configuration, or the other way around.
- Catalyst 4500 Series Switches in a high availability (HA) configuration cannot accept any configuration changes when the standby supervisor is booting.

This restriction also applies to the **configure replace** command.

- If the hw-module line in the existing configuration is being replaced by configuration with a different hw-module line, you must manually reload the entire system. To locate the hw-module line see the following sample output (truncated output):

```
Switch# show running-config brief
Building configuration...
Current configuration : 8730 bytes
Last configuration change at 19:40:01 UTC Mon Oct 26 2015 by cisco
version 15.2
-----output truncated-----
hw-module module 5 mode 1
```

Resetting a Switch to Factory Default Settings

Manufacturing and repair centers can use the **erase /all non-default** command to do the following:

- Clear the nonvolatile configurations and states of the local supervisor engine (NVRAM and flashes).
- Set the factory default parameters on the Catalyst 4500 Series Switches before it is ready to ship to a customer.

For example, entering this command can generate the following output:

```
Switch# erase /all non-default
Erase and format operation will destroy all data in non-volatile storage. Continue?
[confirm]
Formatting bootflash: ...

Format of bootflash complete
Erasing nvram:
Erasing cat4000_flash:
Clearing crashinfo:data
Clearing the last power failure timestamp
Clearing all ROMMON variables
Setting default ROMMON variables:
    ConfigReg=0x2101
    PS1=rommon ! >
    EnableAutoConfig=1
Setting vtp mode to transparent
%WARNING! Please reboot the system for the changes to take effect
Switch#
00:01:48: %SYS-7-NV_BLOCK_INIT: Initialized the geometry of nvram
Switch#
```

If the switch is accessible to a TFTP server, you can copy an image to the bootflash memory with the TFTP command:

```
Switch# copy tftp://192.20.3.123/tftpboot/abc/cat4500-entservices-mz.bin bootflash:
```

When the copying is completed, you can reboot the just-copied switch image to the image stored in the bootflash memory with the **reload** command:

```
Switch# reload

System configuration has been modified. Save? [yes/no]: no
Proceed with reload? [confirm]

00:06:17: %SYS-5-RELOAD: Reload requested by console. Reload Reason: Reload Command.
```

To see details about the default parameters set by the **erase /all non-default** command, see the usage guidelines for the **erase** command in the *Catalyst 4500 Series Switch Cisco IOS Command Reference*.



Administering the Switch

This chapter describes how to perform one-time operations to administer the Catalyst 4500 series switch.

This chapter also describes how to install and configure the Embedded CiscoView network management system to provide a graphical representation of a Catalyst 4500 series switch and to provide a GUI-based management and configuration interface.

This chapter includes the following major sections:

- [Managing the System Time and Date, page 4-1](#)
- [Managing Software Licenses Using Right-To-Use Licenses, page 4-15](#)
- [Configuring a System Name and Prompt, page 4-26](#)
- [Understanding the Domain Name System, page 4-26](#)
- [Creating a Banner, page 4-33](#)
- [Managing the MAC Address Table, page 4-37](#)
- [Managing the ARP Table, page 4-53](#)
- [Configuring Embedded CiscoView Support, page 4-53](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

Managing the System Time and Date

You can configure the system time and date on your switch manually or automatically by using Network Time Protocol (NTP).

These sections contain this configuration information:

- [System Clock, page 4-2](#)
- [Understanding Network Time Protocol, page 4-2](#)
- [Configuring NTP, page 4-3](#)
- [Configuring Time and Date Manually, page 4-11](#)

System Clock

The core of the time service is the system clock, which monitors the date and time. This clock starts when the system starts.

The system clock can provide time to these services:

- User **show** commands
- Logging and debugging messages

The system clock keeps track of time internally based on Universal Time Coordinated (UTC), also known as Greenwich Mean Time (GMT). You can configure information about the local time zone and summer time (daylight saving time) so that the time is correct for the local time zone.

The system clock keeps track of whether the time is *authoritative* or not (whether it was set by a time source considered to be authoritative). If it is not authoritative, the time is available only for display purposes and is not redistributed. For configuration information, see the [“Configuring Time and Date Manually” section on page 4-11](#).

Understanding Network Time Protocol

The NTP is designed to synchronize a network of devices. NTP runs over User Datagram Protocol (UDP), which runs over IP. NTP is documented in RFC 1305.

An NTP network usually gets its time from an authoritative time source, such as a radio clock or an atomic clock attached to a time server. NTP then distributes this time across the network. NTP is extremely efficient; no more than one packet per minute is necessary to synchronize two devices to within a millisecond of one another.

NTP uses the concept of a *stratum* to describe how many NTP hops away a device is from an authoritative time source. A stratum 1 time server has a radio or atomic clock directly attached, a stratum 2 time server receives its time through NTP from a stratum 1 time server, and so on. A device running NTP automatically chooses as its time source the device with the lowest stratum number with which it communicates through NTP. This strategy effectively builds a self-organizing tree of NTP speakers.

NTP avoids synchronizing to a device whose time might not have been synchronized. NTP also compares the time reported by several devices and does not synchronize to a device whose time is significantly different than the others, even if its stratum is lower.

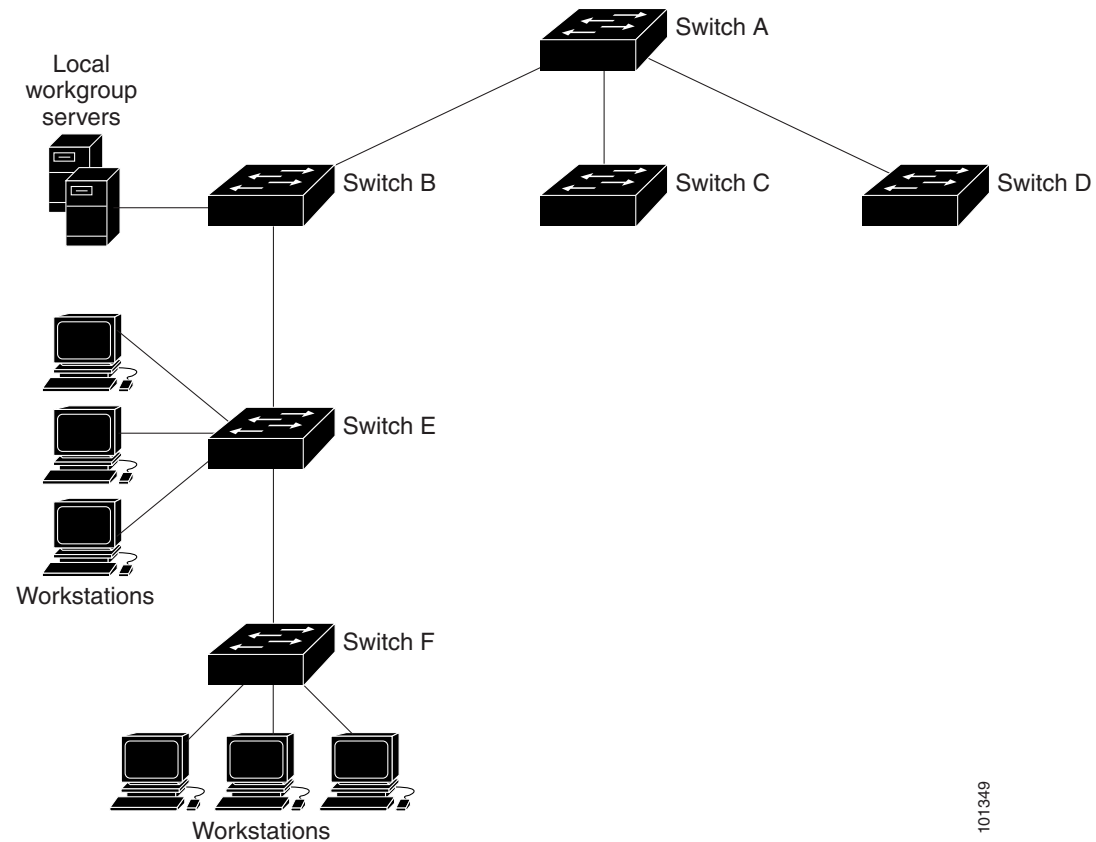
The communications between devices running NTP (known as *associations*) are usually statically configured; each device is given the IP address of all devices with which it should associate. Accurate timekeeping is possible by exchanging NTP messages between each pair of devices with an association. However, in a LAN environment, NTP can be configured to use IP broadcast messages instead. This alternative reduces configuration complexity because each device can be configured to send or receive broadcast messages; however, information flow is one-way only.

The time kept on a device is a critical resource; you should use the security features of NTP to avoid the accidental or malicious setting of an incorrect time. Two mechanisms are available: an access list-based restriction scheme and an encrypted authentication mechanism.

Cisco’s implementation of NTP does not support stratum 1 service; it is not possible to connect to a radio or atomic clock. We recommend that the time service for your network be derived from the public NTP servers available on the IP Internet.

Figure 4-1 shows a typical network example using NTP. Switch A is the NTP master, with Switches B, C, and D configured in NTP server mode, in server association with Switch A. Switch E is configured as an NTP peer to the upstream and downstream switches, Switch B and Switch F, respectively.

Figure 4-1 Typical NTP Network Configuration



If the network is isolated from the Internet, Cisco's implementation of NTP allows a device to act as if it is synchronized through NTP, when it is not. Other devices then synchronize to that device through NTP.

NTP time overrides the time set by any other method.

Several manufacturers include NTP software for their host systems, and a public version for systems running UNIX and its various derivatives is also available. This software allows host systems to be synchronized as well.

Configuring NTP

These sections contain this configuration information:

- [Default NTP Configuration, page 4-4](#)
- [Configuring NTP Authentication, page 4-4](#)
- [Configuring NTP Associations, page 4-6](#)
- [Configuring NTP Broadcast Service, page 4-7](#)
- [Configuring NTP Access Restrictions, page 4-8](#)

- [Configuring the Source IP Address for NTP Packets, page 4-10](#)
- [Displaying the NTP Configuration, page 4-11](#)

Default NTP Configuration

Table 4-1 shows the default NTP configuration.

Table 4-1 Default NTP Configuration

Feature	Default Setting
NTP authentication	Disabled. No authentication key is specified.
NTP peer or server associations	None configured.
NTP broadcast service	Disabled; no interface sends or receives NTP broadcast packets.
NTP access restrictions	No access control is specified.
NTP packet source IP address	The source address is set by the outgoing interface.

NTP is enabled on all interfaces by default. All interfaces receive NTP packets.

Configuring NTP Authentication

This procedure must be coordinated with the administrator of the NTP server; the information you configure in this procedure must be matched by the servers used by the switch to synchronize its time to the NTP server.

To authenticate the associations (communications between devices running NTP that provide for accurate timekeeping) with other devices for security purposes, perform this task:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>ntp authenticate</code>	Enables the NTP authentication feature, which is disabled by default.
Step 3	<code>ntp authentication-key number md5 value</code>	<p>Defines the authentication keys. By default, none are defined.</p> <ul style="list-style-type: none"> • For <i>number</i>, specify a key number. The range is 1 to 4294967295. • md5 specifies that message authentication support is provided by using the message digest algorithm 5 (MD5). • For <i>value</i>, enter an arbitrary string of up to eight characters for the key. <p>The switch does not synchronize to a device unless both have one of these authentication keys, and the key number is specified by the <code>ntp trusted-key key-number</code> command.</p>

	Command	Purpose
Step 4	ntp trusted-key <i>key-number</i>	Specifies one or more key numbers (defined in Step 3) that a peer NTP device must provide in its NTP packets for this switch to synchronize to it. By default, no trusted keys are defined. For <i>key-number</i> , specify the key defined in Step 3. This command provides protection against accidentally synchronizing the switch to a device that is not trusted.
Step 5	end	Returns to privileged EXEC mode.
Step 6	show running-config	Verifies your entries.
Step 7	copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To disable NTP authentication, use the **no ntp authenticate** global configuration command. To remove an authentication key, use the **no ntp authentication-key** *number* global configuration command. To disable authentication of the identity of a device, use the **no ntp trusted-key** *key-number* global configuration command.

This example shows how to configure the switch to synchronize only to devices providing authentication key 42 in the device's NTP packets:

```
Switch# configure terminal
Switch(config)# ntp authenticate
Switch(config)# ntp authentication-key 42 md5 aNiceKey
Switch(config)# ntp trusted-key 42
Switch(config)# end
Switch#
```

Configuring NTP Associations

An NTP association can be a peer association (this switch can either synchronize to the other device or allow the other device to synchronize to it), or it can be a server association (meaning that only this switch synchronizes to the other device, and not the other way around).

To form an NTP association with another device, perform this task:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>ntp peer ip-address [version number]</code> <code>[key keyid] [source interface]</code> <code>[prefer]</code> or <code>ntp server ip-address [version number]</code> <code>[key keyid] [source interface]</code> <code>[prefer]</code>	Configures the switch system clock to synchronize a peer or to be synchronized by a peer (peer association). or Configures the switch system clock to be synchronized by a time server (server association). No peer or server associations are defined by default. <ul style="list-style-type: none"> For <i>ip-address</i> in a peer association, specify either the IP address of the peer providing, or being provided, the clock synchronization. For a server association, specify the IP address of the time server providing the clock synchronization. (Optional) For <i>number</i>, specify the NTP version number. The range is 1 to 3. By default, Version 3 is selected. (Optional) For <i>keyid</i>, enter the authentication key defined by entering the ntp authentication-key global configuration command. (Optional) For <i>interface</i>, specify the interface from which to pick the IP source address. By default, the source IP address is taken from the outgoing interface. (Optional) Enter the prefer keyword to make this peer or server the preferred one that provides synchronization. This keyword reduces switching back and forth between peers and servers.
Step 3	<code>end</code>	Returns to privileged EXEC mode.
Step 4	<code>show running-config</code>	Verifies your entries.
Step 5	<code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

You need to configure only one end of an association; the other device can automatically establish the association. If you are using the default NTP version (Version 3) and NTP synchronization does not occur, try using NTP Version 2. Many NTP servers on the Internet run Version 2.

To remove a peer or server association, use the **no ntp peer ip-address** or the **no ntp server ip-address** global configuration command.

This example shows how to configure the switch to synchronize its system clock with the clock of the peer at IP address 172.16.22.44 using NTP Version 2:

```
Switch# configure terminal
Switch(config)# ntp server 172.16.22.44 version 2
Switch(config)# end
Switch#
```

Configuring NTP Broadcast Service

The communications between devices running NTP (known as *associations*) are usually statically configured; each device is given the IP addresses of all devices with which it should form associations. Accurate timekeeping is possible by exchanging NTP messages between each pair of devices with an association. However, in a LAN environment, NTP can be configured to use IP broadcast messages instead. This alternative reduces configuration complexity because each device can be configured to send or receive broadcast messages. However, the information flow is one-way only.

The switch can send or receive NTP broadcast packets on an interface-by-interface basis if there is an NTP broadcast server, such as a router, broadcasting time information on the network. The switch can send NTP broadcast packets to a peer so that the peer can synchronize to it. The switch can also receive NTP broadcast packets to synchronize its own clock. This section provides procedures for both sending and receiving NTP broadcast packets.

To configure the switch to send NTP broadcast packets to peers so that they can synchronize their clock to the switch, perform this task:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>interface interface-id</code>	Specifies the interface to send NTP broadcast packets, and enter interface configuration mode.
Step 3	<code>ntp broadcast [version number] [key keyid] [destination-address]</code>	Enables the interface to send NTP broadcast packets to a peer. By default, this feature is disabled on all interfaces. <ul style="list-style-type: none"> • (Optional) For <i>number</i>, specify the NTP version number. The range is 1 to 3. If you do not specify a version, Version 3 is used. • (Optional) For <i>keyid</i>, specify the authentication key to use when sending packets to the peer. • (Optional) For <i>destination-address</i>, specify the IP address of the peer that is synchronizing its clock to this switch.
Step 4	<code>end</code>	Returns to privileged EXEC mode.
Step 5	<code>show running-config</code>	Verifies your entries.
Step 6	<code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

To disable the interface from sending NTP broadcast packets, use the **no ntp broadcast** interface configuration command.

This example shows how to configure a port to send NTP Version 2 packets:

```
Switch# configure terminal
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# ntp broadcast version 2
Switch(config-if)# end
Switch#
```

To configure the switch to receive NTP broadcast packets from connected peers, perform this task:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>interface interface-id</code>	Specifies the interface to receive NTP broadcast packets, and enter interface configuration mode.
Step 3	<code>ntp broadcast client</code>	Enables the interface to receive NTP broadcast packets. By default, no interfaces receive NTP broadcast packets.
Step 4	<code>exit</code>	Returns to global configuration mode.
Step 5	<code>ntp broadcastdelay microseconds</code>	(Optional) Changes the estimated round-trip delay between the switch and the NTP broadcast server. The default is 3000 microseconds; the range is 1 to 999999.
Step 6	<code>end</code>	Returns to privileged EXEC mode.
Step 7	<code>show running-config</code>	Verifies your entries.
Step 8	<code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

To disable an interface from receiving NTP broadcast packets, use the **no ntp broadcast client** interface configuration command. To change the estimated round-trip delay to the default, use the **no ntp broadcastdelay** global configuration command.

This example shows how to configure a port to receive NTP broadcast packets:

```
Switch# configure terminal
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# ntp broadcast client
Switch(config-if)# end
Switch#
```

Configuring NTP Access Restrictions

You can control NTP access on two levels as described in these sections:

- [Creating an Access Group and Assigning a Basic IP Access List, page 4-9](#)
- [Disabling NTP Services on a Specific Interface, page 4-10](#)

Creating an Access Group and Assigning a Basic IP Access List

To control access to NTP services by using access lists, perform this task:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>ntp access-group {query-only serve-only serve peer} access-list-number</code>	Creates an access group, and apply a basic IP access list. The keywords have these meanings: <ul style="list-style-type: none"> • query-only—Allows only NTP control queries. • serve-only—Allows only time requests. • serve—Allows time requests and NTP control queries, but does not allow the switch to synchronize to the remote device. • peer—Allows time requests and NTP control queries and allows the switch to synchronize to the remote device. For <i>access-list-number</i> , enter a standard IP access list number from 1 to 99.
Step 3	<code>access-list access-list-number permit source [source-wildcard]</code>	Creates the access list. <ul style="list-style-type: none"> • For <i>access-list-number</i>, enter the number specified in Step 2. • Enter the permit keyword to permit access if the conditions are matched. • For <i>source</i>, enter the IP address of the device that is permitted access to the switch. • (Optional) For <i>source-wildcard</i>, enter the wildcard bits to be applied to the source. Note When creating an access list, remember that, by default, the end of the access list contains an implicit deny statement for everything if it did not find a match before reaching the end.
Step 4	<code>end</code>	Returns to privileged EXEC mode.
Step 5	<code>show running-config</code>	Verifies your entries.
Step 6	<code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

The access group keywords are scanned in this order, from least restrictive to most restrictive:

1. **peer**—Allows time requests and NTP control queries and allows the switch to synchronize itself to a device whose address passes the access list criteria.
2. **serve**—Allows time requests and NTP control queries, but does not allow the switch to synchronize itself to a device whose address passes the access list criteria.
3. **serve-only**—Allows only time requests from a device whose address passes the access list criteria.
4. **query-only**—Allows only NTP control queries from a device whose address passes the access list criteria.

If the source IP address matches the access lists for more than one access type, the first type is granted. If no access groups are specified, all access types are granted to all devices. If any access groups are specified, only the specified access types are granted.

To remove access control to the switch NTP services, use the **no ntp access-group {query-only | serve-only | serve | peer}** global configuration command.

This example shows how to configure the switch to allow itself to synchronize to a peer from access list 99. However, the switch restricts access to allow only time requests from access list 42:

```
Switch# configure terminal
Switch(config)# ntp access-group peer 99
Switch(config)# ntp access-group serve-only 42
Switch(config)# access-list 99 permit 172.20.130.5
Switch(config)# access list 42 permit 172.20.130.6
Switch(config)# end
Switch#
```

Disabling NTP Services on a Specific Interface

NTP services are enabled on all interfaces by default.

To disable NTP packets from being received on an interface, perform this task:

	Command	Purpose
Step 1	configure terminal	Enters global configuration mode.
Step 2	interface <i>interface-id</i>	Enters interface configuration mode, and specify the interface to disable.
Step 3	ntp disable	Disables NTP packets from being received on the interface. By default, all interfaces receive NTP packets. To reenabling receipt of NTP packets on an interface, use the no ntp disable interface configuration command.
Step 4	end	Returns to privileged EXEC mode.
Step 5	show running-config	Verifies your entries.
Step 6	copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuring the Source IP Address for NTP Packets

When the switch sends an NTP packet, the source IP address is normally set to the address of the interface through which the NTP packet is sent. To use a particular source IP address for all NTP packets, use the **ntp source** global configuration command. The address is taken from the specified interface. This command is useful if the address on an interface cannot be used as the destination for reply packets.

To configure a specific interface from which the IP source address is to be taken, perform this task:

	Command	Purpose
Step 1	configure terminal	Enters global configuration mode.
Step 2	ntp source <i>type number</i>	Specifies the interface type and number from which the IP source address is taken. By default, the source address is set by the outgoing interface.
Step 3	end	Returns to privileged EXEC mode.
Step 4	show running-config	Verifies your entries.
Step 5	copy running-config startup-config	(Optional) Saves your entries in the configuration file.

The specified interface is used for the source address for all packets sent to all destinations. If a source address is to be used for a specific association, use the **source** keyword in the **ntp peer** or **ntp server** global configuration command as described in the [“Configuring NTP Associations”](#) section on page 4-6.

Displaying the NTP Configuration

Use the following privileged EXEC commands to display NTP information:

- **show ntp associations [detail]**
- **show ntp status**

For detailed information about the fields in these displays, see the *Cisco IOS Configuration Fundamentals Command Reference, Release 12.3*.

Configuring Time and Date Manually

If no other source of time is available, you can manually configure the time and date after the system is restarted. The time remains accurate until the next system restart. We recommend that you use manual configuration only as a last resort. If you have an outside source to which the switch can synchronize, you do not need to manually set the system clock.

These sections contain this configuration information:

- [Setting the System Clock, page 4-11](#)
- [Displaying the Time and Date Configuration, page 4-12](#)
- [Configuring the Time Zone, page 4-12](#)
- [Configuring Summer Time \(Daylight Saving Time\), page 4-13](#)

Setting the System Clock

If you have an outside source on the network that provides time services, such as an NTP server, you do not need to manually set the system clock.

To set the system clock, perform this task:

	Command	Purpose
Step 1	clock set <i>hh:mm:ss day month year</i> or clock set <i>hh:mm:ss month day year</i>	Manually sets the system clock using one of these formats. <ul style="list-style-type: none"> • For <i>hh:mm:ss</i>, specify the time in hours (24-hour format), minutes, and seconds. The time specified is relative to the configured time zone. • For <i>day</i>, specify the day by date in the month. • For <i>month</i>, specify the month by name. • For <i>year</i>, specify the year (no abbreviation).

This example shows how to manually set the system clock to 1:32 p.m. on July 23, 2001:

```
Switch# clock set 13:32:00 23 July 2001
```

Displaying the Time and Date Configuration

To display the time and date configuration, use the **show clock [detail]** privileged EXEC command.

The system clock keeps an *authoritative* flag that shows whether the time is authoritative (believed to be accurate). If the system clock was set by a timing source such as NTP, the flag is set. If the time is not authoritative, it is used only for display purposes. Until the clock is authoritative and the *authoritative* flag is set, the flag prevents peers from synchronizing to the clock when the peers' time is invalid.

The symbol that precedes the **show clock** display has this meaning:

- *—Time is not authoritative.
- (blank)—Time is authoritative.
- .—Time is authoritative, but NTP is not synchronized.

Configuring the Time Zone

To manually configure the time zone, perform this task:

	Command	Purpose
Step 1	configure terminal	Enters global configuration mode.
Step 2	clock timezone <i>zone</i> <i>hours-offset</i> [<i>minutes-offset</i>]	<p>Sets the time zone.</p> <p>To set the time to UTC, use the no clock timezone global configuration command.</p> <p>The switch keeps internal time in universal time coordinated (UTC), so this command is used only for display purposes and when the time is manually set.</p> <ul style="list-style-type: none"> • For <i>zone</i>, enter the name of the time zone to be displayed when standard time is in effect. The default is UTC. • For <i>hours-offset</i>, enter the hours offset from UTC. • (Optional) For <i>minutes-offset</i>, enter the minutes offset from UTC.
Step 3	end	Returns to privileged EXEC mode.
Step 4	show running-config	Verifies your entries.
Step 5	copy running-config startup-config	(Optional) Saves your entries in the configuration file.

The *minutes-offset* variable in the **clock timezone** global configuration command is available for those cases where a local time zone is a percentage of an hour different from UTC. For example, the time zone for some sections of Atlantic Canada (AST) is UTC-3.5, where the 3 means 3 hours and .5 means 50 percent. The necessary command is **clock timezone AST -3 30**.

Configuring Summer Time (Daylight Saving Time)

To configure summer time (daylight saving time) in areas where it starts and ends on a particular day of the week each year, perform this task:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>clock summer-time zone recurring</code> [<i>week day month hh:mm week day</i> <i>month hh:mm [offset]</i>]	Configures summer time to start and end on the specified days every year. Summer time is disabled by default. If you specify clock summer-time zone recurring without parameters, the summer time rules default to the United States rules. <ul style="list-style-type: none"> For <i>zone</i>, specify the name of the time zone (for example, PDT) to be displayed when summer time is in effect. (Optional) For <i>week</i>, specify the week of the month (1 to 5 or last). (Optional) For <i>day</i>, specify the day of the week (Sunday, Monday...). (Optional) For <i>month</i>, specify the month (January, February...). (Optional) For <i>hh:mm</i>, specify the time (24-hour format) in hours and minutes. (Optional) For <i>offset</i>, specify the number of minutes to add during summer time. The default is 60.
Step 3	<code>end</code>	Returns to privileged EXEC mode.
Step 4	<code>show running-config</code>	Verifies your entries.
Step 5	<code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

The first part of the **clock summer-time** global configuration command specifies when summer time begins, and the second part specifies when it ends. All times are relative to the local time zone. The start time is relative to standard time. The end time is relative to summer time. If the starting month is after the ending month, the system assumes that you are in the southern hemisphere.

This example shows how to specify that summer time starts on the first Sunday in April at 02:00 and ends on the last Sunday in October at 02:00:

```
Switch# configure terminal
Switch(config)# clock summer-time PDT recurring 1 Sunday April 2:00 last Sunday October 2:00
Switch(config)# end
Switch#
```

If summer time in your area does not follow a recurring pattern (configure the exact date and time of the next summer time events), perform this task:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>clock summer-time zone date [month date year hh:mm month date year hh:mm [offset]]</code> or <code>clock summer-time zone date [date month year hh:mm date month year hh:mm [offset]]</code>	<p>Configures summer time to start on the first date and end on the second date.</p> <p>To disable summer time, use the no clock summer-time global configuration command.</p> <p>Summer time is disabled by default.</p> <ul style="list-style-type: none"> For <i>zone</i>, specify the name of the time zone (for example, PDT) to be displayed when summer time is in effect. (Optional) For <i>week</i>, specify the week of the month (1 to 5 or last). (Optional) For <i>day</i>, specify the day of the week (Sunday, Monday...). (Optional) For <i>month</i>, specify the month (January, February...). (Optional) For <i>hh:mm</i>, specify the time (24-hour format) in hours and minutes. (Optional) For <i>offset</i>, specify the number of minutes to add during summer time. The default is 60.
Step 3	<code>end</code>	Returns to privileged EXEC mode.
Step 4	<code>show running-config</code>	Verifies your entries.
Step 5	<code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

The first part of the **clock summer-time** global configuration command specifies when summer time begins, and the second part specifies when it ends. All times are relative to the local time zone. The start time is relative to standard time. The end time is relative to summer time. If the starting month is after the ending month, the system assumes that you are in the southern hemisphere.

To disable summer time, use the **no clock summer-time** global configuration command.

This example shows how to set summer time to start on October 12, 2000, at 02:00, and end on April 26, 2001, at 02:00:

```
Switch# configure terminal
Switch(config)# clock summer-time pdt date 12 October 2000 2:00 26 April 2001 2:00
Switch#
```

Managing Software Licenses Using Right-To-Use Licenses

Starting with Cisco IOS XE Release 3.4.2SG, the RTU licensing feature (**license right-to-use** command) enabled you to order and activate a specific license type and level, and then to manage license usage on your switch. You can also use RTU licenses to

- Upgrade from a lower license to a higher license.
- Downgrade from a higher license to a lower license.

You cannot relocate an RTU license to another device because the license is bundled with the image. So, by upgrading the IOS image, you obtain the RTU license.

The Right-to-Use (RTU) license is not installable and it cannot be cleared; it is available by default.

Starting with Cisco IOS XE Release 3.10.0E, on Cisco Catalyst 4500E Series Switches and Cisco Catalyst 4500-X Series Switches, you can configure add-on licenses. This is in addition to the base license you have configured (LAN Base, IP Base, Enterprise Services). The features available with add-on license levels provide Cisco innovations on the switch, as well as on the Cisco Digital Network Architecture Center (Cisco DNA Center).

Node-Locked Licenses

Prior to IOS Release XE 3.4.2SG, when you upgraded from one license level to another (e.g., IP Base to Entservices), you visited <http://www.cisco.com/go/license>, obtained the desired license using your device's PAK, and then applied the license on your device.

If you want to upgrade or downgrade from one license level to another, we recommend that you use the right-to-use (RTU) license instead of the node-locked license.

For details of this older activation process, see the *Configuring the Cisco IOS Software Activation Feature* guide at this URL:

http://www.cisco.com/en/US/docs/ios/csa/configuration/guide/csa_commands.html

RTU License Levels

Options for Base License Levels

- LAN Base (Only on Cisco Catalyst 4500E Series Switches)
- IP Base
- Enterprise Services

Options for Add-On License Levels

Add-On licenses require a base license as a pre-requisite.

- DNA Essentials
- DNA Advantage

When ordering an add-on license with a particular base license, note the combinations that are permitted and those that are not permitted:

Base License + Add-on License	Cisco Catalyst 4500E Series Switches	Cisco Catalyst 4500-X Series Switches
LAN Base + DNA Essentials	Yes Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, and 7-E	Not applicable
LAN Base + DNA Advantage	No	Not applicable
IP Base + DNA Essentials	Yes Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, and 7-E	Yes
IP Base + DNA Advantage	Yes Supervisor Engine 9-E and Sup8-E	No
Enterprise Services + DNA Essentials	Yes Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, and 7-E	Yes
Enterprise Services + DNA Advantage	Yes Supervisor Engine 9-E and Sup8-E	No

RTU License Types

The types of licenses available to order by duration are:

- Permanent—for base license levels only, and without an expiration date.
- Term—for add-on license levels only.
- Evaluation—for a base or add-on license level, preinstalled on the device, and for a 90-day trial period only.

Ordering with Smart Accounts

We recommend that you use Smart Accounts to order devices as well as licenses. Smart Accounts enable you to manage all of your software licenses for switches, routers, firewalls, access-points or tools from one centralized website. To create Smart Accounts, use the Cisco Smart Software Manager (Cisco SSM).



Note

This is especially relevant to the term licenses that you order, because information about the expiry of term licences is available only through the Cisco SSM website.

For more information about Cisco SSM, see:

<http://www.cisco.com/c/en/us/buy/smart-accounts/software-licensing.html>

Benefits of an RTU License

- They are not associated with a specific switch.

With the *node-locked* license model, in a release prior to IOS Cisco XE 3.4.2SG, a license was applicable to a specific switch UID. Therefore, to activate a license on a new switch, you had to obtain a new license for the new UID. With PRTU licenses, logging on the Cisco server is unnecessary to download and install the license. The license is available with the image.

- They can be instantly activated on any supported switch.

With the node-locked license model, you open the Cisco Product License Registration Portal to obtain a license for a new switch that you purchase or an RMA switch that you need to replace. This process is often cumbersome and lengthy, and applying the license on the new switch is an error-prone activity. With RTU licenses, you can apply a license on a switch and activate it immediately.

- They can be applied without requiring an Internet connection.

With the node-locked license model, you need to access an Internet connection to obtain a license for your device's UID. This may be difficult in some deployment scenarios where an Internet connection is unavailable. With RTU licenses, you can apply a supported license on any switch at any time without requiring an Internet connection to interact with the Cisco Product License Registration Portal.

Guidelines for the RTU Licensing Model

- The RTU license model is based on mutual trust between you and Cisco. When you apply a RTU license, it is implied that you have first purchased the license from Cisco. This agreement is explained in detail in the EULA, which is displayed when you activate the license.
- The RTU license model does not replace the node-locked license model. Instead, it simplifies upgrading or moving your switch's license. The node-locked license model is still available.
- Base licenses (LAN Base, IP Base, Enterprise Services) may be ordered only with a permanent license type.
- Add-on licenses (DNA Essentials and DNA Advantage) may be ordered only with a term license type.

You can set up Cisco SSM to receive daily e-mail alerts, to be notified of expiring add-on licenses that you want to renew.

Applying an RTU License

To apply an RTU license on a switch, follow these steps:

-
- | | |
|---------------|---|
| Step 1 | Upgrade from one license level to another by using the Cisco sales ordering tool to purchase the license. You will receive an e-mail or paper confirmation that grants you permission to activate the license on your switch. |
| Step 2 | Apply the license by entering the appropriate commands on your switch. If you are upgrading a license on a switch, enter the activation command to activate the higher license. If you are moving a license from one switch to another, enter the deactivation command on the first switch and the activation command on the second switch. |

**Note**

Prior to IOS Release XE 3.4.2SG, you provided the license file to a Cisco server, then obtained the new license file. With IOS Release XE 3.4.2SG, you do not require those operations. Once the IOS image is upgraded, you receive the license on the switch and activate it with the **license right-to-use activate** *feature name* command.

Step 3 Read and accept the EULA.

Step 4 If you change the license boot level, reboot.

**Note**

Reboot is not necessary when:
 Activating RTU for the same license level as the existing one.
 Activating or deactivating an add-on license level.

Activating an RTU License

To activate a PRTU license on a switch, use either of the following commands in privileged EXEC mode:

	Command	Purpose
Step 1	license right-to-use [activate deactivate] [ipbase entservices lanbase] [acceptEULA] OR license right-to-use [activate deactivate] [addon { dna-essentials dna-advantage }] { subscription evaluation } [acceptEULA]	Activates the specified license level on a switch and enables acceptance of the end-user license agreement (EULA).
	Example Switch# license right-to-use activate ipbase acceptEULA Switch# license right-to-use activate addon dna-essentials subscription acceptEULA	

	Command	Purpose
Step 2	reload Example Switch# reload	Reboots the switch. The reminder to accept the EULA is displayed after reload if it was not accepted earlier. When changing license level, you are not required to save the configuration. But, it is a good practice to ensure all the configuration is stored properly before reload. Changing from a higher license level to a lower license level on reboot will remove CLIs that are not applicable. Ensure that all features in the lower license level that are actively used are not removed.
Step 3	show license right-to-use usage	Displays detailed usage information. If the command output displays the following values for In-Use and EULA parameters, disregard the In-Use status. In-Use: no EULA: yes The license will be in use if it is activated and EULA is accepted.

Deactivating an RTU License

To deactivate a PRTU license on a switch, enter the following command in privileged EXEC mode:

	Command	Purpose
Step 1	license right-to-use deactivate [ipbase entservices lanbase addon {dna-essentials dna-advantage }]	Deactivates a license on a switch.
Step 2	reload Example Switch# reload	Reboots the switch. The reminder to accept the EULA is displayed after reload if it was not accepted earlier. When changing license level, you are not required to save the configuration. But, it is a good practice to ensure all the configuration is stored properly before reload. Changing from a higher license level to a lower license level on reboot will remove CLIs that are not applicable. Ensure that all features in the lower license level that are actively used are not removed.

Displaying Software License Information

To display information about the software licenses on your switch, use one of these methods:

- Use Cisco License Manager to view license and device information. In the GUI, the discovery and polling features collect all the license and device information that appears in the Properties window. For detailed instructions, see the Cisco License Manager online help.

- Use the Cisco IOS privileged EXEC commands in [Table 2](#).

Table 2 **Commands for Displaying Software License Information**

Command	Description
show license agent {counters session}	Displays the information about the software license agent. For information about the show license agent privileged EXEC command, see the <i>Cisco Software Activation Tasks and Commands</i> feature module.
show license [agent all detail evaluation expiring feature file image in-use permanent right-to-use statistics status summary udi]	Displays information about the software license. For information about the show license privileged EXEC command, see the <i>Cisco IOS Software Licensing</i> feature module at this URL: http://www.cisco.com/en/US/docs/ios/12_2/12_2se/feature/guide/se_cisl.html
show license right-to-use [eula default detail mismatch summary usage]	Displays information related to the right-to-use licenses on the device. If the command output displays the following values for In-Use and EULA parameters, disregard the In-Use status. In-Use: no EULA: yes The license will be in use if it is activated and EULA is accepted.
show version	Displays the software licenses installed on the switch.

This is an example of output from the **show license all** command:

```
Switch# show license all
License Store: Primary License Storage
License Store: Dynamic License Storage
StoreIndex: 0 Feature: entservices Version: 1.0
  License Type: Evaluation
    Evaluation total period: 8 weeks 4 days
    Evaluation period left: 7 weeks
  License State: Active, Not in Use, EULA accepted
  License Count: Non-Counted
  License Priority: Low
StoreIndex: 1 Feature: entservices Version: 1.0
  License Type: PermanentRightToUse
  License State: Inactive
  License Count: Non-Counted
StoreIndex: 2 Feature: ipbase Version: 1.0
  License Type: Evaluation
    Evaluation total period: 8 weeks 4 days
    Evaluation period left: 8 weeks 4 days
  License State: Inactive
  License Count: Non-Counted
  License Priority: None
StoreIndex: 3 Feature: ipbase Version: 1.0
  License Type: PermanentRightToUse
  License State: Active, In Use
  License Count: Non-Counted
StoreIndex: 4 Feature: lanbase Version: 1.0
```

```

License Type: PermanentRightToUse
License State: Active, Not in Use, EULA accepted
License Count: Non-Counted
StoreIndex: 5 Feature: apcount Version: 1.0
License Type: Evaluation
Evaluation total period: 8 weeks 4 days
Evaluation period left: 8 weeks 4 days
License State: Inactive
License Count: 0/0
License Priority: Low
StoreIndex: 6 Feature: apcount Version: 1.0
License Type: PermanentRightToUse
License State: Active, Not in Use, EULA accepted
License Count: 0/0
StoreIndex: 7 Feature: dna-advantage Version: 1.0
License Type: Subscription
License State: Not Activated
License Count: Non-Counted
StoreIndex: 8 Feature: dna-essentials Version: 1.0
License Type: Subscription
License State: Not Activated

```

This is an example of output from the **show license feature** command:

```

Switch# show license feature
Feature name Enforcement Evaluation Clear Allowed Enabled Right...
-----
entservices true true true false true
ibase true true true true true
lanbase false false true false false
internal_service true false true false false

```

This is an example of output from the **show license file** command:

```

Switch# show license file
License Store: Primary License Storage
License Index: 1
License: 11 ibase 1.0 LONG NORMAL STANDALONE EXCL INFINITE_KEYS INFINITE_KEYS NEVER NEVER
NiL SLM_CODE CL_ND_LCK NiL *1DELA9XDSFSJXAH400 NiL NiL NiL 5_MINS WS-C4507R+EFOX1327G52D
xLt5Q1e2VJi03pzp3GSE3Prvxwyf0,SLjP0SXuZ0q0f4QTXyc1pSQY51xj31fh7ZfTD6AskNyeUYT8sCUesi9IVKB8
5wsZSX1HZiXwOd9RHp3mjmnxhFDnS0e6UxjgXgqvV:$AQEBIf8B//kh4dluXv+U+xjUPlzoc3++jpV9d8He4jOuba
fbkmmOtaOYAoB3inJLnlLyv50VCuRqwinXo3s+nsLU7rOtdOxoIXYZAo3LYmUJ+MFzsq1hKoJVlPyEvQ8H21MNUjVb
hoN0gyIWsyiJaM8AQIkVBQFzhr10GYolVzdzfJfEPQIx6tZ++/Vtc/q3SF/5Ko8XCY=
Comment:
Hash: Z+EY3ce1csQlVpRGc5NNy5ypmds=
License Store: Dynamic License Storage
License Store: Primary License Storage
License Store: Dynamic License Storage
License Index: 0
License: 11 entservices 1.0 LONG TRIAL DISABLED 1440 DISABLED STANDALONE ADD INFINITE_KEYS
INFINITE_KEYS NEVER NEVER NiL SLM_CODE DEMO NiL NiL Ni NiL NiL 5_MINS NiL
BGf3gQnLuroDmnMjMwWVa2ukr8kP2JZyinKpmOXpa32jwPuSBmHvcSRiSSaqBngV8$AQEBIQAB//FTlc+Qu1Xlg2
Z+yB2StUHHymf2w5PEw+cYg/hTOKYCI+oXi0jwBZ2iLrYTKYwxSSRqwinXo3s+nsLU7rOtdOxoIXYZAo3LYmUJ+MFz
sq1hKoJVlPyEvQ8H21MNUjVbhoN0gyIWsyiJaM8AQIkVBQFzhr10GYolVzdzfJfEPQIx6tZ++/Vtc/q3SF/5Ko8XCY
=
Comment:
Hash: Rm09Kumi8BFKq0wCAx2CcUDE6rg=
License Index: 1
License: 12 entservices 1.0 LONG TRIAL DISABLED DISABLED DISABLED STANDALONE ADD
INFINITE_KEYS INFINITE_KEYS 1 JAN 2006 1 JAN 2035 NiL NiL SLM_CODE DEMO NiL NiL Ni NiL NiL
5_MINS NOTLOCKEDNOTLOCKEDHBL
1lnG2zXePlBt,ifk7ZReL80LqzvzgrUCelWrBp41FC3jOKer6ZMT7XC4834W3Ev7fmleXoWaK58t:oDeH5RI1V3dVE

```

```

2VpAnYb7WiKDz9En8PfrI7vewhayNbschEXBD9:tfPfir6GaALUFwsLxcqYzHuL2$AQEBIf8B//mCS09+7kn+8zTC
3WX1YS9if+g0e8AjRRu1Jq3Kye4y8wv4c+Y9FHJ7Ro/mw7ERwqRqWInXo3s+nsLU7rOtdOxoIXYZAo3LYmUJ+MFzsq
lhKoJV1PyEvQ8H21MNUjVbhoN0gyIWsyiJaM8AQIkVBQFzhr10GYolVzdzfJfEPQIx6tZ++/Vtc/q3SF/5Ko8XCXY=
Comment:
Hash: 9w09jAFGBzi2w6XQC1jL0Be2p+Y=
License Index: 2
License: 11 ipbase 1.0 LONG TRIAL DISABLED 1440 DISABLED STANDALONE ADD INFINITE_KEYS
INFINITE_KEYS NEVER NEVER NiL SLM_CODE DEMO NiL NiL Ni NiL NiL 5_MINS NiL
YXNJUtpFJiC2Rpdt1SJNVQBCpQUBNt59tdkJJTgKwmLTKj:vmp,sVkMiIRYLfMHQfj$AQEBIf8B//kagzg0R7bT5rn
6dVYVPUFmxBlUsblGgbkInHYo55DJzHE/Bqnlf9keNdSyZPbUhSRqWInXo3snsLU7rOtdOxoIXYZAo3LYmUJ+MFzsq
lhKoJV1PyEvQ8H21MNUjVbhoN0gyIWsyiJaM8AQIkVBQFzhr10GYolVzdzfJfEPQIx6tZ++/Vtc/q3SF/5Ko8XCXY=
Comment:
Hash: H6zsXVLv9TFImTfFGm0tK4VHJ2Q=
License Index: 3
License: 12 ipbase 1.0 LONG TRIAL DISABLED DISABLED DISABLED STANDALONE ADD INFINITE_KEYS
INFINITE_KEYS 1 JAN 2006 1 JAN 2035 NiL NiL SLM_CODE DEMO NiL NiL Ni NiL NiL 5_MINS
NOTLOCKEDNOTLOCKEDHBL
Zh0GdIANTlXwW6LJgQ95LB0aCazzbsjSOL4HUaqcySLcOvcLq,d04oTgS8pJbHIO3BaD0tgELHog9egQWj9bCJ3,sm
2jRaJkgkhYK09BrbWYLOA,m03Qe2E,TPJou8fms:LtvrfctzLbuJmB0Xcb68MPLm$AQEBIf8B//+08JwRWipzfjtwl
AItclx+D6NLhKMyqS1hJoxCM1Txgw8BpmG5QQY5nCiE14CPvVKRqWInXo3s+nsLU7rOtdOxoIXYZAo3LYmUJ+MFzsq
lhKoJV1PyEvQ8H21MNUjVbhoN0gyIWsyiJaM8AQIkVBQFzhr10GYolVzdzfJfEPQIx6tZ++/Vtc/q3SF/5Ko8XCXY=
Comment:
Hash: S3Ks+G07ueugA9hMFpKXGTF12So=

```

This is an example of output from the **show license statistics** command:

```

Switch# show license statistics
Administrative statistics
Install success count: 4
Install failure count: 1
Install duplicate count: 0
Comment add count: 0
Comment delete count: 0
Clear count: 0
Save count: 0
Save cred count: 0
Client status Request success count 1 Request failure count 0 Release count 0 Global
Notify count 1

```

This is an example of output from the **show license status** command:

```

Switch# show license status
License Type Supported
permanent Non-expiring node locked license
extension Expiring node locked license
evaluation Expiring non node locked license
License Operation Supported install Install license clear Clear license annotate Comment
license save Save license revoke Revoke license call-home License call-home Call-home
Operation Supported show pak Display license pak via call-home install Install license via
call-home revoke Revoke license via call-home resend Fetch license via call-home Device
status Device Credential type: IMAGE Device Credential Verification: PASS Rehost Type:
DC_OR_IC

```

When you enter the **show license udi** command on WS-C4507R+E, this output appears:

```

Switch# show license udi
Device# PID SN UDI
-----
*0 WS-C4507R+E FOX1327G52D WS-C4507R+E:FOX1327G52D

```



Note The **show license udi** command output shows details on the current switch.

This is an example of the **show license right-to-use** command:

```
Switch# show license right-to-use
License Store: Primary License Storage
License Store: Dynamic License Storage
StoreIndex: 1 Feature: entservices Version: 1.0
    License Type: PermanentRightToUse
    License State: Inactive
    License Count: Non-Counted
StoreIndex: 3 Feature: ipbase Version: 1.0
    License Type: PermanentRightToUse
    License State: Active, In Use
    License Count: Non-Counted
StoreIndex: 4 Feature: lanbase Version: 1.0
    License Type: PermanentRightToUse
    License State: Active, Not in Use, EULA accepted
    License Count: Non-Counted
StoreIndex: 6 Feature: apcount Version: 1.0
    License Type: PermanentRightToUse
    License State: Active, Not in Use, EULA accepted
    License Count: 0/0
StoreIndex: 7 Feature: dna-advantage Version: 1.1
    License Type: Subscription
    License State: Not Activated
    License Count: Non-Counted
StoreIndex: 8 Feature: dna-essentials Version: 1.1
    License Type: Subscription
    License State: Not Activated
    License Count: Non-Counted
```

This is an example of the **show license right-to-use usage** command:

```
Switch# show license right-to-use usage
Index: 1
License Name: entservices
    License Type: PermanentRightToUse
    usage-duration: Life Time
    In-Use: no
    EULA: no
Index: 3
License Name: ipbase
    License Type: PermanentRightToUse
    usage-duration: Life Time
    In-Use: yes
    EULA: no
Index: 4
License Name: lanbase
    License Type: PermanentRightToUse
    usage-duration: Life Time
    In-Use: no
    EULA: yes
Index: 6
License Name: apcount
    License Type: PermanentRightToUse
    usage-duration: Life Time
    In-Use: no
    EULA: yes
Index: 7
License Name: dna-advantage
    License Type: Subscription
    Usage-duration: CSSM Managed
    In-Use: no
    EULA: yes
Index: 8
License Name: dna-essentials
    License Type: Subscription
```

```
Usage-duration: CSSM Managed
In-Use: no
EULA: yes
```

This is an example of the **show license summary** command:

```
Switch# show license summary
Index 0 Feature: entservices
Period left: 8 weeks 3 days
License Type: Evaluation
License State: Active, Not in Use, EULA accepted
License Count: Non-Counted
License Priority: Low
Index 1 Feature: ipbase
Period left: Life time
License Type: Permanent
License State: Active, In Use
License Count: Non-Counted
License Priority: Medium
Index 2 Feature: lanbase
Period left: 0 seconds
Index 3 Feature: internal_service
Period left: 0 seconds
```

This is an example of the **show license evaluation** command:

```
Switch# show license evaluation
License Store: Primary License Storage
License Store: Dynamic License Storage
StoreIndex: 0 Feature: entservices Version: 1.0
License Type: Evaluation
Evaluation total period: 8 weeks 4 days
Evaluation period left: 8 weeks 3 days
License State: Active, Not in Use, EULA accepted
License Count: Non-Counted
License Priority: Low
StoreIndex: 2 Feature: ipbase Version: 1.0
License Type: Evaluation
Evaluation total period: 8 weeks 4 days
Evaluation period left: 8 weeks 4 days
License State: Inactive
License Count: Non-Counted
License Priority: None
```

This is an example of the **show license image levels** command:

```
Switch# show license image levels
Module name Image level Priority Configured Valid license
-----
WS-X45-SUP7-E entservices 1 NO entservices
ipbase 2 NO ipbase
lanbase 3 NO lanbase

Module Name Role Current Level Reboot Level
-----
WS-X45-SUP7-E Active ipbase ipbase
```

This is an example of the **show license expiring** command

```
Switch# show license expiring
License Store: Primary License Storage
License Store: Dynamic License Storage
StoreIndex: 0 Feature: entservices Version: 1.0
License Type: Evaluation
Evaluation total period: 8 weeks 4 days
```



```
Evaluation period left: 8 weeks 3 days
License State: Active, Not in Use, EULA accepted
License Count: Non-Counted
License Priority: Low
StoreIndex: 2 Feature: ipbase Version: 1.0
License Type: Evaluation
Evaluation total period: 8 weeks 4 days
Evaluation period left: 8 weeks 4 days
License State: Inactive
License Count: Non-Counted
License Priority: None
Switch#
```

This is an example of the **show license in-use** command

```
Switch# show license in-use
License Store: Primary License Storage
StoreIndex: 1 Feature: ipbase Version: 1.0
License Type: Permanent
License State: Active, In Use
License Count: Non-Counted
License Priority: Medium
License Store: Dynamic License Storage
```

Configuring a System Name and Prompt

You configure the system name on the switch to identify it. By default, the system name and prompt are *Switch*.

If you have not configured a system prompt, the first 20 characters of the system name are used as the system prompt. A greater-than symbol [>] is appended. The prompt is updated whenever the system name changes.

For complete syntax and usage information for the commands used in this section, see the *Cisco IOS Configuration Fundamentals Command Reference, Release 12.3* and the *Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols, Release 12.3*.

To manually configure a system name, perform this task:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>hostname name</code>	Manually configures a system name. When you set the system name, it is also used as the system prompt. The default setting is <i>switch</i> . The name must follow the rules for ARPANET hostnames. They must start with a letter, end with a letter or digit, and have as interior characters only letters, digits, and hyphens. Names can be up to 63 characters. To return to the default hostname, use the no hostname global configuration command.
Step 3	<code>end</code>	Returns to privileged EXEC mode.
Step 4	<code>show running-config</code>	Verifies your entries.
Step 5	<code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

Understanding the Domain Name System

The DNS protocol controls the Domain Name System (DNS), a distributed database with which you can map hostnames to IP addresses. When you configure DNS on your switch, you can substitute the hostname for the IP address with all IP commands, such as **ping**, **telnet**, **connect**, and related Telnet support operations.

IP defines a hierarchical naming scheme that allows a device to be identified by its location or domain. Domain names are pieced together with periods (.) as the delimiting characters. For example, Cisco Systems is a commercial organization that IP identifies by a *com* domain name, so its domain name is *cisco.com*. A specific device in this domain, for example, the File Transfer Protocol (FTP) system is identified as *ftp.cisco.com*.

To keep track of domain names, IP has defined the concept of a domain name server, which holds a cache (or database) of names mapped to IP addresses. To map domain names to IP addresses, you must first identify the hostnames, specify the name server that is present on your network, and enable the DNS.

These sections contain this configuration information:

- [Default DNS Configuration, page 4-27](#)
- [Setting Up DNS, page 4-27](#)

- [Displaying the DNS Configuration, page 4-28](#)
- [DNS for IPv6, page 4-28](#)
- [IPv6 Router Advertisement Options for DNS Configuration, page 4-29](#)
- [Configuring DNS Server Using IPv6 Router Advertisement Options, page 4-30](#)
- [Configuring DNS Search List Using IPv6 Router Advertisement Options, page 4-31](#)
- [Troubleshooting DNS Servers and DNS Search Lists, page 4-32](#)

Default DNS Configuration

Table 4-3 shows the default DNS configuration.

Table 4-3 *Default DNS Configuration*

Feature	Default Setting
DNS enable state	Enabled.
DNS default domain name	None configured.
DNS servers	No name server addresses are configured.

Setting Up DNS

To set up your switch to use the DNS, perform this task:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>ip domain-name name</code>	<p>Defines a default domain name that the software uses to complete unqualified hostnames (names without a dotted-decimal domain name).</p> <p>To remove a domain name, use the no ip domain-name name global configuration command.</p> <p>Do not include the initial period that separates an unqualified name from the domain name.</p> <p>At boot time, no domain name is configured; however, if the switch configuration comes from a BOOTP or Dynamic Host Configuration Protocol (DHCP) server, then the default domain name might be set by the BOOTP or DHCP server (if the servers were configured with this information).</p>
Step 3	<code>ip name-server server-address1 [server-address2 ... server-address6]</code>	<p>Specifies the address of one or more name servers to use for name and address resolution.</p> <p>To remove a name server address, use the no ip name-server server-address global configuration command.</p> <p>You can specify up to six name servers. Separate each server address with a space. The first server specified is the primary server. The switch sends DNS queries to the primary server first. If that query fails, the backup servers are queried.</p>

	Command	Purpose
Step 4	<code>ip domain-lookup</code>	(Optional) Enables DNS-based hostname-to-address translation on your switch. This feature is enabled by default. To disable DNS on the switch, use the no ip domain-lookup global configuration command. If your network devices require connectivity with devices in networks for which you do not control name assignment, you can dynamically assign device names that uniquely identify your devices by using the global Internet naming scheme (DNS).
Step 5	<code>end</code>	Returns to privileged EXEC mode.
Step 6	<code>show running-config</code>	Verifies your entries.
Step 7	<code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

If you use the switch IP address as its hostname, the IP address is used and no DNS query occurs. If you configure a hostname that contains no periods (.), a period followed by the default domain name is appended to the hostname before the DNS query is made to map the name to an IP address. The default domain name is the value set by the **ip domain-name** global configuration command. If there is a period (.) in the hostname, the Cisco IOS software looks up the IP address without appending any default domain name to the hostname.

Displaying the DNS Configuration


To display the DNS configuration information, use the **show running-config** privileged EXEC command.

DNS for IPv6

IPv6 supports DNS record types that are supported in the DNS name-to-address and address-to-name lookup processes. The DNS record types support IPv6 addresses. IPv6 also supports the reverse mapping of IPv6 addresses to DNS names.

The table below lists the IPv6 DNS record types.

Table 4-1 IPv6 DNS Record Types

Record Type	Description	Format
AAAA	Maps a hostname to an IPv6 address. (Equivalent to an A record in IPv4.)	www.abc.test AAAA 3FFE:YYYY:C18:1::2
PTR	Maps an IPv6 address to a hostname. (Equivalent to a PTR record in IPv4.)	2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.8.1 .c.0.y.y.y.e.f.f.3.ip6.int PTR www.abc.test
	 Note Cisco software supports resolution of PTR records for the IP6.INT domain.	

IPv6 Router Advertisement Options for DNS Configuration

Starting with Cisco IOS XE Release 3.10.0E, IPv6 Router Advertisement Options for DNS Configuration feature is supported on the Cisco Catalyst 4500E Series Switches and Cisco Catalyst 4500-X Series Switches, at all available license levels for these platforms.

DNS configuration based on IPv6 Router Advertisement (RA) options ([RFC 6106](#)) is useful in networks where:

- An IPv6 host's address is autoconfigured through IPv6 stateless address autoconfiguration.
- There is either no DHCP IPv6 infrastructure at all or some hosts do not have a DHCPv6 client.

In these scenarios, it is not feasible to manually configure hosts each time they connect to a different network. While a one-time static configuration is possible, it is generally not desirable on general-purpose hosts such as laptops and mobile phones.

Neighbor Discovery (ND) for IPv6 and IPv6 stateless address autoconfiguration provide ways to configure either fixed or mobile nodes with one or more IPv6 addresses, default routers, and some other parameters [RFC4861][RFC4862]. Most Internet services are identified by using a DNS name. The following RA options defined in RFC 6106 provide the DNS information needed for an IPv6 host to reach Internet services:

Recursive DNS Server (RDNSS) Option

RDNSS provides a recursive DNS resolution service for translating domain names into IP addresses as defined in [RFC1034] and [RFC1035]. The RDNSS option contains one or more IPv6 addresses of recursive DNS servers. All addresses share the same Lifetime value, which is the maximum time in seconds over which this RDNSS address may be used for name resolution. If different Lifetime values are required, multiple RDNSS options can be used. There can be up to 5 DNS servers.

DNS Search List (DNSSL) Option

DNSSL is a list of DNS suffix domain names used by IPv6 hosts when they perform DNS query searches for short, unqualified domain names. The DNSSL option contains one or more domain names. All domain names share the same Lifetime value, which is the maximum time in seconds over which this DNSSL may be used. If different Lifetime values are required, multiple DNSSL options can be used. There can be up to 5 DNSSLs.

**Note**

If DNS information is available from multiple RAs and/or from DHCP, the host must maintain an ordered list of this DNS information.

Configuring DNS Server Using IPv6 Router Advertisement Options

RFC 6106 specifies IPv6 Router Advertisement (RA) options to allow IPv6 routers to advertise a list of recursive DNS server (RDNSS) addresses used for the DNS name resolution in IPv6 hosts.

The DNS lifetime range should be between maximum RA interval and twice the maximum RA interval:

```
(max ra interval) <= dns lifetime <= (2*(max ra interval))
```

The maximum RA interval can have a value between 4 and 1800 seconds (default is 240 seconds). Therefore, the following configuration will result in a lifetime out of range:

```
Switch(config-if)# ipv6 nd ra dns server 4::4 3600 <<< Lifetime configured out of range
for the interface having default maximum RA interval
```

The following is a sample configuration of recursive DNS server:

```
Switch# configure terminal
Switch(config)# interface GigabitEthernet 0/2/0/0
Switch(config-if)# ipv6 nd prefix 2002:4898:e8:1011::/64 1111 222
Switch(config-if)# ipv6 nd prefix 2002:4899:e8:1011::/64 1111 222
Switch(config-if)# ipv6 nd ra-lifetime 9000
Switch(config-if)# ipv6 nd ra dns server 4::4 infinite-lifetime
Switch(config-if)# ipv6 nd router-preference high
Switch(config-if)# ipv6 nd ra specific route 3::3/116 Lifetime 1112 preference low
```

Use the **show running-config** command to display the DNS server configuration for a specific interface:

```
Switch# show running-config interface gigabitEthernet 0/2/0/0
interface GigabitEthernet0/2/0/0
ipv6 nd prefix 2002:4898:e8:1011::/64 1111 222
ipv6 nd prefix 2002:4899:e8:1011::/64 1111 222
ipv6 nd ra-lifetime 9000
/* RDNSS configuration */
ipv6 nd ra dns server 4::4 infinite-lifetime
ipv6 nd router-preference high
ipv6 nd ra specific route 3::3/116 Lifetime 1112 preference low !
```

Use the **show ipv6 nd idb interface** command to verify DNS server configuration based on IPv6 RA options:

```
Switch# show ipv6 nd idb interface gigabitEthernet 0/2/0/0 detail location 0/2/CPU0
Mon Jul 4 14:28:53.422 IST
ifname: Gi0/2/0/0, ifh: 0x01000300, iftype: 15, VI-type: 0, Pseudo IDB: FALSE
vrf-id: 0x60000000, table-id: 0xe0800000
Mac Addr: 02d1.1e2b.0baf, size: 6, Vlan tag set: FALSE

Media Name: ether, Media Encap: 0x1 (ARPA)
Mac Length: 6, Media Header Len: 14, Media Proto: 0xdd86
Current Encap: 0x1 (ARPA), Mcast Encap : 0x1 (ARPA)

IPv6 Interface: Enabled, IPV6: Enabled, MPLS: Disabled
Link local address: fe80::d1:1eff:fe2b:baf, Global Addr count: 1
Global Addresses:1::1(0x2),
Default Prefix Address: ::, Prefix Addr Count: 3,
Prefix addresses: 1::(0x401), 2002:4898:e8:1011::(0x4), 2002:4899:e8:1011::(0x4)
```

```

RA Specific Route Count: 1,
RA Specific Route : Address 3:: Prefix Length 116 Lifetime 1112 Preference Low

RA DNS Servers Addr Count: 3,
RA DNS Server adresse: Address 5::6 Lifetime 240
RA DNS Server adresse: Address 5::5 Lifetime 240
RA DNS Server adresse: Address 4::4 Lifetime 4294967295

```

Use the **no ipv6 nd ra dns server ipv6 address** command to delete a single DNS server under an interface. Use the **no ipv6 nd ra dns server** command to delete all DNS servers under an interface.

Configuring DNS Search List Using IPv6 Router Advertisement Options

RFC 6106 specifies IPv6 Router Advertisement (RA) options to allow IPv6 routers to advertise a DNS Search List (DNSSL) to IPv6 hosts for an enhanced DNS configuration.

The DNS lifetime range should be between maximum RA interval and twice the maximum RA interval:

```
(max ra interval) <= dns lifetime <= (2*(max ra interval))
```

The maximum RA interval can have a value between 4 and 1800 seconds (default is 240 seconds).

Therefore, the following configuration will result in a lifetime out of range:

```
Switch(config-if)#ipv6 nd ra dns search list sss.com 3600 <<< Lifetime configured out of
range for the interface having default maximum RA interval
```

The following is a sample configuration of a DNS search list:

```

Switch# configure terminal
Switch(config)# interface GigabitEthernet 0/2/0/0
Switch(config-if)# ipv6 nd prefix 2002:4898:e8:1011::/64 1111 222
Switch(config-if)# ipv6 nd prefix 2002:4899:e8:1011::/64 1111 222
Switch(config-if)# ipv6 nd ra-lifetime 9000
Switch(config-if)# ipv6 nd ra dns search list aaa.cc.com infinite-lifetime
Switch(config-if)# ipv6 address 1::1/64

```



Note The domain name configuration should follow RFC 1035. If not, the configuration will be rejected. For example, the following domain name configuration will result in an error:

```
Switch(config-if)# ipv6 nd ra dns search list .aaa.cc.com infinite-lifetime
```

Use the **no ipv6 nd ra dns search list name** command to delete a single DNS search list under an interface. Use the **no ipv6 nd ra dns search list** command to delete all DNS search lists under an interface.

Use the **show running-config** command to display the configuration for a specific interface:

```

Switch# show running-config interface gigabitEthernet 0/2/0/0
interface GigabitEthernet0/2/0/0
  ipv6 nd prefix 2002:4898:e8:1011::/64 1111 222
  ipv6 nd prefix 2002:4899:e8:1011::/64 1111 222
  ipv6 nd ra-lifetime 9000
  ipv6 nd router-preference high
  ipv6 nd ra specific route 3::3/116 Lifetime 1112 preference low
/* DNSSL configuration */?ipv6 nd ra dns search list aaa.cc.com infinite-lifetime ipv6
address 1::1/64?!

```

Use the **show ipv6 nd idb interface** command to verify DNS search list configuration based on IPv6 RA options:

```
Switch# show ipv6 nd idb interface gigabitEthernet 0/2/0/0 detail location 0/2/CPU0
Mon Jul  4 14:28:53.422 IST

ifname: Gi0/2/0/0, ifh: 0x01000300, iftype: 15, VI-type: 0, Pseudo IDB: FALSE
vrf-id: 0x60000000, table-id: 0xe0800000
Mac Addr: 02d1.1e2b.0baf, size: 6, Vlan tag set: FALSE

Media Name: ether, Media Encap: 0x1 (ARPA)
Mac Length: 6, Media Header Len: 14, Media Proto: 0xdd86
Current Encap: 0x1 (ARPA), Mcast Encap : 0x1 (ARPA)

IPv6 Interface: Enabled, IPv6: Enabled, MPLS: Disabled
Link local address: fe80::d1:1eff:fe2b:baf, Global Addr count: 1
Global Addresses: 1::1(0x2),
Default Prefix Address: ::, Prefix Addr Count: 3,
Prefix addresses: 1::(0x401), 2002:4898:e8:1011::(0x4), 2002:4899:e8:1011::(0x4)

RA Specific Route Count: 1,
RA Specific Route : Address 3:: Prefix Length 116 Lifetime 1112 Preference Low

RA DNS Search List Count: 3,
RA DNS Search List : Name aaa.bbb.ccc.com Lifetime 240
RA DNS Search List : Name aaa.bbb.com Lifetime 240
RA DNS Search List : Name aaa.cc.com Lifetime 4294967295
```

Troubleshooting DNS Servers and DNS Search Lists

Recursive DNS servers and DNS search lists are sent as part of RA messages.

The interface for which the DNS servers and DNS search lists are configured, should be IPv6 enabled and the IPv6 protocol should be up. Verify using the following command:

```
Switch# show ipv6 interface gigabitEthernet 0/2/0/0
Wed Jun 29 18:52:31.936 IST
GigabitEthernet0/2/0/0 is Up, ipv6 protocol is Up, Vrfid is default (0x60000000)
IPv6 is enabled, link-local address is fe80::d1:1eff:fe2b:baf
Global unicast address(es):
  1::1, subnet is 1::/64
Joined group address(es): ff02::1:ff00:1 ff02::1:ff2b:baf ff02::2
  ff02::1
```

If the interface has IPv6 protocol up and the RAs are not being sent, then verify whether RA is suppressed:

```
Switch# show running-config interface gigabitEthernet 0/2/0/0
interface GigabitEthernet0/2/0/0
ipv6 nd suppress-ra <<< configuration of RA suppression under interface
ipv6 nd router-preference high
ipv6 nd ra specific route 3::3/116 Lifetime 1112 preference low
ipv6 address 1::1/64
!
```

If the interface has IPv6 protocol up and the RAs are not being sent, then verify whether RA is enabled in Interface Descriptor Block (IDB). The RA flag is 1.

```
Switch# show ipv6 nd idb interface gigabitEthernet 0/2/0/0 detail location 0/2/CPU0
Thu Jun 30 20:02:20.211 IST
ifname: Gi0/2/0/0, ifh: 0x01000300, iftype: 15, VI-type: 0, Pseudo IDB: FALSE
vrf-id: 0x60000000, table-id: 0xe0800000
Mac Addr: 02d1.1e2b.0baf, size: 6, Vlan tag set: FALSE
RA Specific Route Count: 1,
RA Specific Route : Address 3:: Prefix Length 116 Lifetime 1112 Preference Low
RA DNS Servers Addr Count: 1,
```



```

RA DNS Server address: Address 4::4 Lifetime 4294967295 <<< RDNSS for interface
RA DNS Search List Count: 1,
RA DNS Search List : Name aaa.cc.com Lifetime 4294967295 <<< DNSSL for interface
RA flag: 0x1, Unicast RA send: FALSE, Initial RA count: 3, RA pkts sent count: 442 <<<
RA flag should be 1

```

Run the IPv6 ND traces to debug any particular issue related to a DNS servers and DNS search lists:

```

Switch# show ipv6 nd trace location 0/2/CPU0
Jun 30 20:07:03.508 nd/fevent 0/2/CPU0 t26702 Sending RA to ff02::1 on
GigabitEthernet0/2/0/0 (0x1000300)
Jun 30 20:07:03.508 nd/fevent 0/2/CPU0 t26702 hoplimit 64 lifetime 9000 reachable 0
retrans 0
Jun 30 20:07:03.509 nd/fevent 0/2/CPU0 t26702 1::/64 Onlink Auto
Jun 30 20:07:03.509 nd/fevent 0/2/CPU0 t26702 valid 2592000 pref 604800
Jun 30 20:07:03.509 nd/fevent 0/2/CPU0 t26702 2002:4898:e8:1011::/64 Onlink Auto
Jun 30 20:07:03.509 nd/fevent 0/2/CPU0 t26702 valid 1111 pref 222
Jun 30 20:07:03.509 nd/fevent 0/2/CPU0 t26702 2002:4899:e8:1011::/64 Onlink Auto
Jun 30 20:07:03.509 nd/fevent 0/2/CPU0 t26702 valid 1111 pref 222
Jun 30 20:07:03.509 nd/fevent 0/2/CPU0 t26702 ra specific route address 3:: lifetime
1112 preference Low
Jun 30 20:07:03.509 nd/fevent 0/2/CPU0 t26702 ra dns server address 5::6 lifetime 240
first
Jun 30 20:07:03.509 nd/fevent 0/2/CPU0 t26702 ra dns server address 5::5 lifetime 240
part of same ra dns server option
Jun 30 20:07:03.509 nd/fevent 0/2/CPU0 t26702 ra dns server address 4::4 lifetime
4294967295 first
Jun 30 20:07:03.509 nd/fevent 0/2/CPU0 t26702 ra dns search list name aaa.bbb.ccc.com
lifetime 240 first
Jun 30 20:07:03.509 nd/fevent 0/2/CPU0 t26702 ra dns search list name aaa.bbb.com
lifetime 240 part of same ra dns search list option
Jun 30 20:07:03.509 nd/fevent 0/2/CPU0 t26702 ra dns search list name aaa.cc.com
lifetime 4294967295 first
Jun 30 20:07:03.509 nd/fevent 0/2/CPU0 t26702 nd_send_ra: sending RA pakesize=320,
plen=280
Jun 30 20:07:03.509 nd/fevent 0/2/CPU0 t26702 nd_pak_send: size=320, ifh
GigabitEthernet0/2/0/0 (0x1000300) , priority=2 to ipv6-io
Jun 30 20:07:03.509 nd/fevent 0/2/CPU0 t26702 nd_pak_send: sending pak=0x60c07d8b with
NO FVS set, size=320, ifh GigabitEthernet0/2/0/0 (0x1000300) to ipv6-io

```

Creating a Banner

You can configure a message-of-the-day (MOTD) and a login banner. The MOTD banner displays on all connected terminals at login and is useful for sending messages that affect all network users (such as impending system shutdowns).

The login banner also displays on all connected terminals. It appears after the MOTD banner and before the login prompts.



Note

For complete syntax and usage information for the commands used in this section, see the *Cisco IOS Configuration Fundamentals Command Reference, Release 12.3*.

The contain this configuration information:

- [Default Banner Configuration, page 4-34](#)
- [Configuring a Message-of-the-Day Login Banner, page 4-34](#)

- [Configuring a Login Banner, page 4-36](#)

Default Banner Configuration

The MOTD and login banners are not configured.

Configuring a Message-of-the-Day Login Banner

You can create a single or multiline message banner that appears on the screen when someone logs in to the switch.

To configure a MOTD login banner, perform this task:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>banner motd c message c</code>	<p>Specifies the message of the day.</p> <p>To delete the MOTD banner, use the no banner motd global configuration command.</p> <p>For <i>c</i>, enter the delimiting character of your choice, for example, a pound sign (#), and press the Return key. The delimiting character signifies the beginning and end of the banner text. Characters after the ending delimiter are discarded.</p> <p>Note When configuring a banner using the "#" sign as a delimiter on Supervisor Engine 7-E and Supervisor Engine 7L-E, you must first turn off shell processing with the no shell processing command. Else, you can not exit from the banner configuration.</p> <pre>### With shell processing enabled ### Sup7# conf t Enter configuration commands, one per line. End with CNTL/Z. Sup7(config)# ban Sup7(config)# banner lo Sup7(config)# banner login # Enter TEXT message. End with the character '#' test login banner # ## e# Sup7(config)# ### With shell processing disabled ### Sup7(config)# banner login # Enter TEXT message. End with the character '#' test login banner # Sup7(config)#</pre> <p>For <i>message</i>, enter a banner message up to 255 characters. You cannot use the delimiting character in the message.</p>

	Command	Purpose
Step 3	<code>end</code>	Returns to privileged EXEC mode.
Step 4	<code>show running-config</code>	Verifies your entries.
Step 5	<code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

This example shows how to configure a MOTD banner for the switch by using the pound sign (#) symbol as the beginning and ending delimiter:

```
Switch(config)# banner motd #
it is a secure site. Only authorized users are allowed.
For access, contact technical support.
#
Switch(config)#
```

This example shows the banner that appears from the previous configuration:

```
Unix> telnet 172.2.5.4
Trying 172.2.5.4...
Connected to 172.2.5.4.
Escape character is '^]'.

it is a secure site. Only authorized users are allowed.
For access, contact technical support.

User Access Verification

Password:
```

Configuring a Login Banner

You can configure a login banner to be displayed on all connected terminals. This banner appears after the MOTD banner and before the login prompt.

To configure a login banner, perform this task:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>banner login c message c</code>	<p>Specifies the login message.</p> <p>To delete the login banner, use the no banner login global configuration command.</p> <p>For <i>c</i>, enter the delimiting character of your choice, for example, a pound sign (#), and press the Return key. The delimiting character signifies the beginning and end of the banner text. Characters after the ending delimiter are discarded.</p> <p>Note When configuring a banner using the "#" sign as a delimiter on Supervisor Engine 7-E and Supervisor Engine 7L-E, you must first turn off shell processing with the no shell processing command. Else, you can not exit from the banner configuration.</p> <pre> ### With shell processing enabled ### Sup7# conf t Enter configuration commands, one per line. End with CNTL/Z. Sup7(config)# ban Sup7(config)# banner lo Sup7(config)# banner login # Enter TEXT message. End with the character '#' test login banner # ## e# Sup7(config)# ### With shell processing disabled ### Sup7(config)# banner login # Enter TEXT message. End with the character '#' test login banner # Sup7(config)# </pre> <p>For <i>message</i>, enter a login message up to 255 characters. You cannot use the delimiting character in the message.</p>
Step 3	<code>end</code>	Returns to privileged EXEC mode.
Step 4	<code>show running-config</code>	Verifies your entries.
Step 5	<code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

This example shows how to configure a login banner for the switch by using the dollar sign (\$) symbol as the beginning and ending delimiter:

```
Switch# configuration terminal
Switch(config)# banner login $
Access for authorized users only. Please enter your username and password.
$
Switch(config)# end
Switch#
```

Managing the MAC Address Table

The MAC address table contains address information that the switch uses to forward traffic between ports. All MAC addresses in the address table are associated with one or more ports. The address table includes these types of addresses:

- Dynamic address—A source MAC address that the switch learns and then ages when it is not in use.
- Static address—A manually entered unicast address that does not age and that is not lost when the switch resets.

The address table lists the destination MAC address, the associated VLAN ID, and port number associated with the address and the type (static or dynamic).



Note

For complete syntax and usage information for the commands used in this section, see the command reference for this release.

These sections contain this configuration information:

- [Building the Address Table, page 4-37](#)
- [MAC Addresses and VLANs, page 4-38](#)
- [Default MAC Address Table Configuration, page 4-39](#)
- [Changing the Address Aging Time, page 4-39](#)
- [Removing Dynamic Address Entries, page 4-40](#)
- [Configuring MAC Change Notification Traps, page 4-40](#)
- [Configuring MAC Move Notification Traps, page 4-42](#)
- [Configuring MAC Threshold Notification Traps, page 4-44](#)
- [Adding and Removing Static Address Entries, page 4-45](#)
- [Configuring Unicast MAC Address Filtering, page 4-46](#)
- [Disabling MAC Address Learning on a VLAN, page 4-48](#)
- [Displaying Address Table Entries, page 4-53](#)

Building the Address Table

With multiple MAC addresses supported on all ports, you can connect any port on the switch to individual workstations, repeaters, switches, routers, or other network devices. The switch provides dynamic addressing by learning the source address of packets it receives on each port and adding the

address and its associated port number to the address table. As stations are added or removed from the network, the switch updates the address table, adding new dynamic addresses and aging out those that are not in use.

The aging interval is globally configured. However, the switch maintains an address table for each VLAN, and STP can accelerate the aging interval on a per-VLAN basis.

The switch sends packets between any combination of ports, based on the destination address of the received packet. Using the MAC address table, the switch forwards the packet only to the port associated with the destination address. If the destination address is on the port that sent the packet, the packet is filtered and not forwarded. The switch always uses the store-and-forward method: complete packets are stored and checked for errors before transmission.

MAC Addresses and VLANs

All addresses are associated with a VLAN. An address can exist in more than one VLAN and have different destinations in each. Unicast addresses, for example, could be forwarded to port 1 in VLAN 1 and ports 9, 10, and 1 in VLAN 5.

Each VLAN maintains its own logical address table. A known address in one VLAN is unknown in another until it is learned or statically associated with a port in the other VLAN.

When PVLANs are configured, address learning depends on the type of MAC address:

- Dynamic MAC addresses learned in one VLAN of a PVLAN are replicated in the associated VLANs. For example, a MAC address learned in a private-VLAN secondary VLAN is replicated in the primary VLAN.
- Static MAC addresses configured in a primary or secondary VLAN are not replicated in the associated VLANs. When you configure a static MAC address in a PVLAN primary or secondary VLAN, you should also configure the same static MAC address in all associated VLANs.

For more information about PVLANs, see [Chapter 47, “Configuring Private VLANs.”](#)

Default MAC Address Table Configuration

[Table 4-4](#) shows the default MAC address table configuration.

Table 4-4 Default MAC Address Table Configuration

Feature	Default Setting
Aging time	300 seconds
Dynamic addresses	Automatically learned
Static addresses	None configured

Changing the Address Aging Time

Dynamic addresses are source MAC addresses that the switch learns and then ages when they are not in use. You can change the aging time setting for all VLANs or for a specified VLAN.

Setting too short an aging time can cause addresses to be prematurely removed from the table. When the switch receives a packet for an unknown destination, it floods the packet to all ports in the same VLAN as the receiving port. This unnecessary flooding can impact performance. Setting too long an aging time can cause the address table to be filled with unused addresses, which prevents new addresses from being learned. Flooding results, which can impact switch performance.

To configure the dynamic address table aging time, perform this task:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>mac address-table aging-time [0 10-1000000] [vlan vlan-id]</code>	<p>Sets the length of time that a dynamic entry remains in the MAC address table after the entry is used or updated.</p> <p>To return to the default value, use the no mac address-table aging-time global configuration command.</p> <p>The range is 10 to 1000000 seconds. The default is 300. You can also enter 0, which disables aging. Static address entries are never aged or removed from the table.</p> <p>For <i>vlan-id</i>, valid IDs are 1 to 4094.</p>
Step 3	<code>end</code>	Returns to privileged EXEC mode.

	Command	Purpose
Step 4	<code>show mac address-table aging-time</code>	Verifies your entries.
Step 5	<code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

Removing Dynamic Address Entries

To remove all dynamic entries, use the **clear mac address-table dynamic** command in EXEC mode. You can also remove a specific MAC address (**clear mac address-table dynamic address** *mac-address*), remove all addresses on the specified physical port or port channel (**clear mac address-table dynamic interface** *interface-id*), or remove all addresses on a specified VLAN (**clear mac address-table dynamic vlan** *vlan-id*).

To verify that dynamic entries have been removed, use the **show mac address-table dynamic** privileged EXEC command.

Configuring MAC Change Notification Traps

MAC change notification allows you to track users on a network by storing the MAC change activity on the switch. Whenever the switch learns or removes a MAC address, an SNMP notification can be generated and sent to the network management system. If you have many users entering and exiting the network, you can set a trap interval time to bundle the notification traps and reduce network traffic. The MAC notification history table stores the MAC address activity for each hardware port for which the trap is enabled. MAC address notifications are generated for dynamic and static MAC addresses; events are not generated for self addresses or multicast addresses.

To send MAC change notification traps to an NMS host, perform this task:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>snmp-server host host-addr [traps informs] {version {1/2c/3}} [auth noauth priv] community-string [udp-port port] [notification-type]</code>	Specifies the recipient of the trap message. <ul style="list-style-type: none"> For <i>host-addr</i>, specify the name or address of the NMS. Specify traps (the default) to send SNMP traps to the host. Specify informs to send SNMP informs to the host. Specify the SNMP version to support. Version 1, the default, is not available with informs. For <i>community-string</i>, specify the string to send with the notification operation. Though you can set this string by using the snmp-server host command, we recommend that you define this string by using the snmp-server community command before using the snmp-server host command. For <i>notification-type</i>, use the mac-notification keyword.

	Command	Purpose
Step 3	<code>snmp-server enable traps mac-notification change</code>	Enables the switch to send MAC change traps to the NMS. To disable the switch from sending MAC change notification traps, use the no snmp-server enable traps mac-notification change global configuration command.
Step 4	<code>mac address-table notification change</code>	Enables the MAC address change notification feature.
Step 5	<code>mac address-table notification change [interval value] [history-size value]</code>	Enters the trap interval time and the history table size. <ul style="list-style-type: none"> • (Optional) For interval value, specify the notification trap interval in seconds between each set of traps that are generated to the NMS. The range is 0 to 2147483647 seconds; the default is 1 second. • (Optional) For history-size value, specify the maximum number of entries in the MAC notification history table. The range is 0 to 500; the default is 1. To disable the MAC change notification feature, use the no mac address-table notification change global configuration command.
Step 6	<code>interface interface-id</code>	Enters interface configuration mode, and specifies the interface on which to enable the SNMP MAC change notification trap.
Step 7	<code>snmp trap mac-notification change {added removed}</code>	Enables the MAC change notification trap. <ul style="list-style-type: none"> • Enable the MAC change notification trap whenever a MAC address is added on this interface. • Enable the MAC change notification trap whenever a MAC address is removed from this interface. To disable the MAC change notification traps on a specific interface, use the no snmp trap mac-notification change {added removed} interface configuration command.
Step 8	<code>end</code>	Returns to privileged EXEC mode.
Step 9	<code>show mac address-table notification change interface</code> <code>show running-config</code>	Verifies your entries.
Step 10	<code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

This example shows how to specify 172.69.59.93 as the network management system, enable the switch to send MAC change notification traps to the network management system, enable the MAC change notification feature, set the interval time to 60 seconds, set the history-size to 100 entries, and enable traps whenever a MAC address is added on the specified port:

```
Switch# configure terminal
Switch(config)# snmp-server host 172.69.59.93 private mac-notification
Switch(config)# snmp-server enable traps mac-notification change
Switch(config)# mac address-table notification change
Switch(config)# mac address-table notification change interval 60
Switch(config)# mac address-table notification change history-size 100
Switch(config)# interface fastethernet0/2
Switch(config-if)# snmp trap mac-notification change added
Switch(config-if)# end
Switch# show mac address-table notification change interface
MAC Notification Feature is Enabled on the switch
MAC Notification Flags For All Ethernet Interfaces :
-----
Interface          MAC Added Trap  MAC Removed Trap
-----
GigabitEthernet1/1  Enabled         Enabled
GigabitEthernet1/2  Enabled         Enabled
GigabitEthernet1/3  Enabled         Enabled
GigabitEthernet1/4  Enabled         Enabled
GigabitEthernet1/5  Enabled         Enabled
GigabitEthernet1/6  Enabled         Enabled
GigabitEthernet1/7  Enabled         Enabled
GigabitEthernet1/8  Enabled         Enabled
GigabitEthernet1/9  Enabled         Enabled
GigabitEthernet1/10 Enabled         Enabled
GigabitEthernet1/11 Enabled         Enabled
GigabitEthernet1/12 Enabled         Enabled

Switch#
```

Configuring MAC Move Notification Traps

When you configure MAC move notification, an SNMP notification is generated and sent to the network management system whenever a MAC address moves from one port to another within the same VLAN.

To configure MAC move notification, perform this task:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>snmp-server host host-addr [traps informs] {version {1/2c/3}} [auth noauth priv] community-string [udp-port port] [notification-type]</code>	Specifies the recipient of the trap message. <ul style="list-style-type: none"> For <i>host-addr</i>, specify the name or address of the NMS. Specify traps (the default) to send SNMP traps to the host. Specify informs to send SNMP informs to the host. Specify the SNMP version to support. Version 1, the default, is not available with informs. For <i>community-string</i>, specify the string to send with the notification operation. Though you can set this string by using the snmp-server host command, we recommend that you define this string by using the snmp-server community command before using the snmp-server host command. For <i>notification-type</i>, use the mac-notification keyword.
Step 3	<code>snmp-server enable traps mac-notification move</code>	Enables the switch to send MAC move notification traps to the NMS. To disable the switch from sending MAC notification traps, use the no snmp-server enable traps mac-notification move global configuration command.
Step 4	<code>mac address-table notification mac-move</code>	Enables the MAC-move notification feature. To disable this feature, use the no mac-address-table notification mac-move global configuration command.
Step 5	<code>end</code>	Returns to privileged EXEC mode.
Step 6	<code>show mac address-table notification mac-move</code> <code>show running-config</code>	Displays the MAC-move notification status.
Step 7	<code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

This example shows how to specify 172.69.59.93 as the network management system, enable the switch to send MAC move notification traps to the NMS, enable the MAC move notification feature, and enable traps whenever a MAC address moves from one port to another:

```
Switch# configure terminal
Switch(config)# snmp-server host 171.69.59.93 private mac-notification
Switch(config)# snmp-server enable traps mac-notification move
Switch(config)# mac address-table notification mac-move
Switch(config)# end
Switch# show mac address-table notification mac-move
MAC Move Notification: Enabled
```

Configuring MAC Threshold Notification Traps

When you configure MAC threshold notification, an SNMP notification is generated and sent to the network management system when a MAC address table (MAT) threshold limit is reached or exceeded.

To configure MAC address threshold notification, perform this task:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>snmp-server host host-addr [traps/informs] {version {1/2c/3}} [auth noauth priv] community-string [udp-port port] [notification-type]</code>	Specifies the recipient of the trap message. <ul style="list-style-type: none"> For <i>host-addr</i>, specify the name or address of the NMS. Specify traps (the default) to send SNMP traps to the host. Specify informs to send SNMP informs to the host. Specify the SNMP version to support. Version 1, the default, is not available with informs. For <i>community-string</i>, specify the string to send with the notification operation. Though you can set this string by using the snmp-server host command, we recommend that you define this string by using the snmp-server community command before using the snmp-server host command. For <i>notification-type</i>, use the mac-notification keyword.
Step 3	<code>snmp-server enable traps mac-notification threshold</code>	Enables the switch to send MAC threshold notification traps to the NMS. To disable the switch from sending MAC threshold notification traps, use the no snmp-server enable traps mac-notification threshold global configuration command.
Step 4	<code>mac address-table notification threshold</code>	Enables the MAC address threshold notification feature. To disable this feature, use the no address-table notification threshold global configuration command.
Step 5	<code>mac address-table notification threshold [limit percentage] [interval time]</code>	Enters the threshold value for the MAT usage monitoring. <ul style="list-style-type: none"> (Optional) For limit percentage, specify the percentage of the MAT utilization; valid values are from 1 to 100 percent. Default is 50 percent. (Optional) For interval time, specify the time between notifications; valid values are greater than or equal to 120 seconds. Default is 120 seconds.

	Command	Purpose
Step 6	<code>end</code>	Returns to privileged EXEC mode.
Step 7	<code>show mac address-table notification threshold</code> <code>show running-config</code>	Displays the MAC utilization threshold notification status.
Step 8	<code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

This example shows how to specify 172.69.59.93 as the network management system, enable the MAC threshold notification feature, enable the switch to send MAC threshold notification traps to the NMS, set the interval to 123 seconds, and set the limit to 78 percent:

```
Switch# configure terminal
Switch(config)# snmp-server host 171.69.59.93 private mac-notification
Switch(config)# snmp-server enable traps mac-notification threshold
Switch(config)# mac address-table notification threshold
Switch(config)# mac address-table notification threshold interval 123
Switch(config)# mac address-table notification threshold limit 78
Switch(config)# end
Switch# show mac-address-table notification threshold
      Status      limit      Interval
-----+-----+-----
      enabled      78          123
Switch#
```

Adding and Removing Static Address Entries

A static address has these characteristics:

- It is manually entered in the address table and must be manually removed.
- It can be a unicast or multicast address.
- It does not age and is retained when the switch restarts.

You can add and remove static addresses and define the forwarding behavior for them. The forwarding behavior defines how a port that receives a packet forwards it to another port for transmission. Because all ports are associated with at least one VLAN, the switch acquires the VLAN ID for the address from the ports that you specify. You can specify a different list of destination ports for each source port.

A packet with a static address that arrives on a VLAN where it has not been statically entered is flooded to all ports and not learned.

You add a static address to the address table by specifying the destination MAC unicast address and the VLAN from which it is received. Packets received with this destination address are forwarded to the interface specified with the *interface-id* option.

When you configure a static MAC address in a private-VLAN primary or secondary VLAN, you should also configure the same static MAC address in all associated VLANs. Static MAC addresses configured in a private-VLAN primary or secondary VLAN are not replicated in the associated VLAN. For more information about PVLANS, see [Chapter 47, “Configuring Private VLANs.”](#)

To add a static address, perform this task:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>mac address-table static mac-addr vlan vlan-id interface interface-id</code>	<p>Adds a static address to the MAC address table.</p> <ul style="list-style-type: none"> For <i>mac-addr</i>, specify the destination MAC unicast address to add to the address table. Packets with this destination address received in the specified VLAN are forwarded to the specified interface. For <i>vlan-id</i>, specify the VLAN for which the packet with the specified MAC address is received. Valid VLAN IDs are 1 to 4094. For <i>interface-id</i>, specify the interface to which the received packet is forwarded. Valid interfaces include physical ports or port channels. <p>You can specify static multicast addresses for multiple interface IDs. However, you cannot assign static unicast MAC address to multiple interfaces with the same MAC address and VLAN ID.</p> <p>To remove static entries from the address table, use the no mac address-table static mac-addr vlan vlan-id [interface interface-id] global configuration command.</p>
Step 3	<code>end</code>	Returns to privileged EXEC mode.
Step 4	<code>show mac address-table static</code>	Verifies your entries.
Step 5	<code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

This example shows how to add the static address c2f3.220a.12f4 to the MAC address table. When a packet is received in VLAN 4 with this MAC address as its destination address, the packet is forwarded to the specified port:

```
Switch# configure terminal
Switch(config)# mac address-table static c2f3.220a.12f4 vlan 4 interface
gigabitethernet0/1
Switch(config)# end
Switch#
```

Configuring Unicast MAC Address Filtering

When unicast MAC address filtering is enabled, the switch drops packets with specific source or destination MAC addresses. This feature is disabled by default and only supports unicast static addresses.

When using unicast address filtering, consider these guidelines:

- Multicast MAC addresses, broadcast MAC addresses, and router MAC addresses are not supported. If you specify one of these addresses when entering the **mac address-table static vlan drop** global configuration command, one of these messages appears:
 - % Only unicast addresses can be configured to be dropped
 - % CPU destined address cannot be configured as drop address
- Packets that are forwarded to the CPU are also not supported.

- If you add a unicast MAC address as a static address and configure unicast MAC address filtering, the switch either adds the MAC address as a static address or drops packets with that MAC address, depending on which command was entered last. The second command that you entered overrides the first command.

For example, if you enter the **mac address-table static vlan interface** global configuration command followed by the **mac address-table static vlan drop** command, the switch drops packets with the specified MAC address as a source or destination.

If you enter the **mac address-table static vlan drop** global configuration command followed by the **mac address-table static vlan interface** command, the switch adds the MAC address as a static address.

You enable unicast MAC address filtering and configure the switch to drop packets with a specific address by specifying the source or destination unicast MAC address and the VLAN from which it is received.

To configure the switch to drop a source or destination unicast static address, perform this task:

	Command	Purpose
Step 1	configure terminal	Enters global configuration mode.
Step 2	mac address-table static <i>mac-addr</i> vlan <i>vlan-id</i> drop	Enables unicast MAC address filtering and configure the switch to drop a packet with the specified source or destination unicast static address. <ul style="list-style-type: none"> • For <i>mac-addr</i>, specify a source or destination unicast MAC address. Packets with this MAC address are dropped. • For <i>vlan-id</i>, specify the VLAN for which the packet with the specified MAC address is received. Valid VLAN IDs are 1 to 4094. To disable unicast MAC address filtering, use the no mac address-table static vlan global configuration command.
Step 3	end	Returns to privileged EXEC mode.
Step 4	show mac address-table static	Verifies your entries.
Step 5	copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to enable unicast MAC address filtering and to configure the switch to drop packets that have a source or destination address of c2f3.220a.12f4. When a packet is received in VLAN 4 with this MAC address as its source or destination, the packet is dropped:

```
Switch# configure terminal
Switch(config)# mac address-table static c2f3.220a.12f4 vlan 4 drop
Switch(config)# end
Switch#
```



Note

To filter MAC addresses on a secondary VLAN, specify the corresponding primary VLAN in the above configuration. If the specified VLAN is a primary VLAN, all matching packets received in this primary VLAN and associated secondary VLANs are dropped.

Disabling MAC Address Learning on a VLAN

By default, MAC address learning is enabled on all VLANs on the switch. By controlling which VLANs can learn MAC addresses, you can manage the available MAC address table space. By disabling learning on a VLAN, you can conserve the MAC address table space because all the MAC addresses seen on this VLAN are not learned.

Before disabling MAC address learning, you should understand the network topology and features deployed. Many Layer 2 features use MAC addresses and may not work properly if learning is disabled. Because disabling learning causes flooding of packets, you need to understand the impact of flooding on the network.

These sections contain this information:

- [Deployment Scenarios, page 4-49](#)
- [Configuring Disable MAC Address Learning, page 4-48](#)
- [Usage Guidelines, page 4-49](#)
- [Deployment Scenarios, page 4-49](#)
- [Feature Compatibility, page 4-51](#)
- [Feature Incompatibility, page 4-52](#)

Configuring Disable MAC Address Learning

To disable MAC address learning on a VLAN, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# no mac address-table learning vlan <i>vlan-id range</i>	Disables MAC address learning on the specified VLAN or VLANs. You can specify a single VLAN ID or a range of VLAN IDs separated by a hyphen or comma. Valid VLAN IDs are 1 to 4094. You can reenable MAC address learning on a VLAN by entering the mac address-table learning vlan global configuration command.
Step 3	Switch(config)# end	Returns to privileged EXEC mode.
Step 4	Switch# show mac address-table learning [vlan <i>vlan-id range</i>]	Displays the MAC address learning status of all VLANs or a specified VLAN.
Step 5	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to disable learning on any VLAN or range of VLANs:

```
Switch# configure terminal
Switch(config)# no mac address-table learning vlan 9-16
Switch(config)# end
Switch#

Switch# show mac address-table learning
Learning disabled on vlans: 9-11,13-16

Switch# show mac address-table learning vlan 10-15
Learning disabled on vlans: 10-11,13-15
```


Usage Guidelines

**Note**

These guidelines are advisory only. Contact the Cisco solution provider team for specific solution implementations.

When disabling MAC address learning on a VLAN, consider these guidelines:

- If learning is disabled on a VLAN with an SVI interface, it floods every IP packet in the Layer 2 domain. Because this flooding may be undesirable, you should disable MAC address learning on a SVI VLAN carefully.
- If you provide a VLAN range that includes reserved VLAN (such as 1000-1006), the command is accepted and disable learning is enabled for all VLANs except for 1002-005 (that is, 1000-1001,1006). However, if you specify an invalid range (such as 1-5000), the command fails and disable learning is not enabled on any of the VLANs.
- With PVLANS, you need to disable learning on the primary VLAN and all secondary VLANs associated with that primary VLANs. Otherwise, you encounter traffic flooding in one direction and unicast flooding in the other direction.
- To disable MAC address learning on a VLAN, consider the flooding implications.

Deployment Scenarios

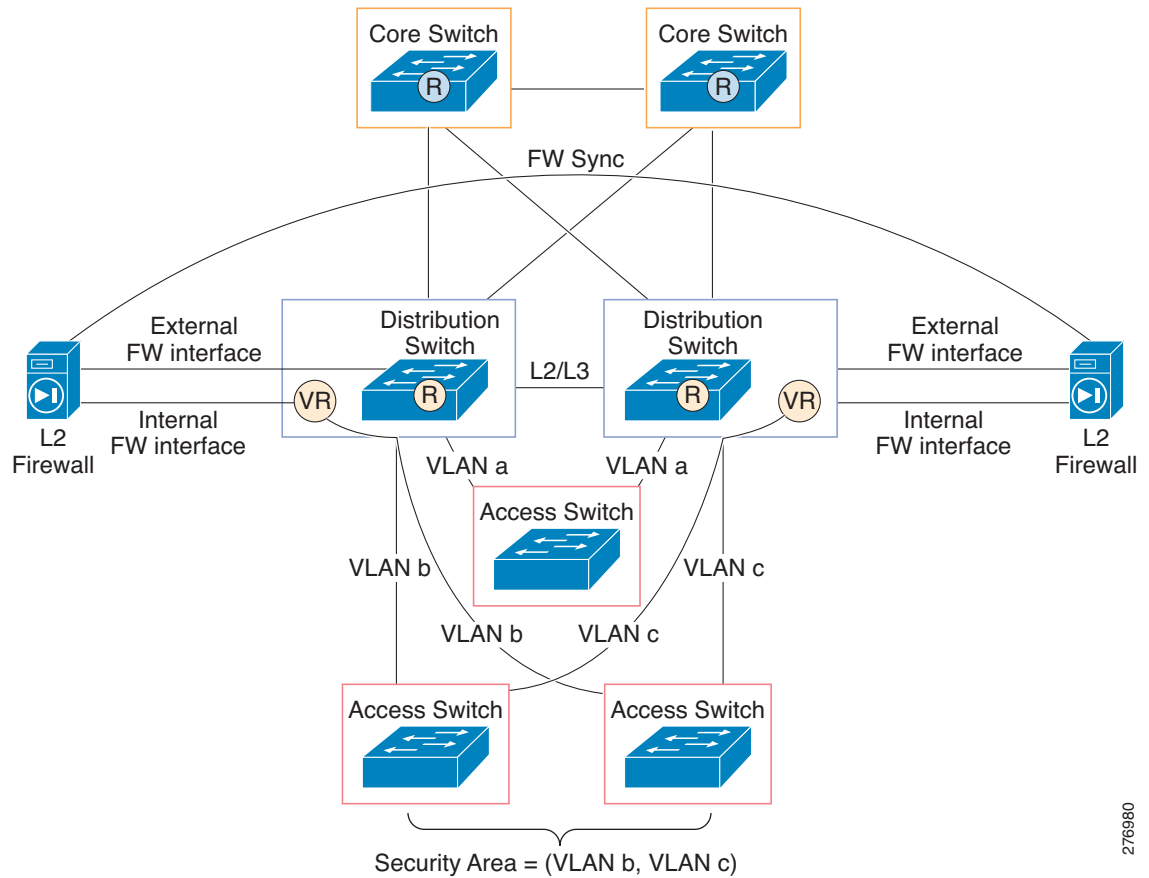
This section includes these deployment scenarios:

- [Metro \(Point to Point Links\), page 4-49](#)
- [Network Load Balancers, page 4-50](#)
- [Layer 2 Firewall or Cache, page 4-51](#)

Metro (Point to Point Links)

In this topology, you have two ports on a VLAN; traffic enters one and must exit the other. On a point-to-point link in metro networks, numerous MAC addresses are on these types of ports by disabling learning on the VLAN to which these two ports belong, many entries in the MAC address table space can be saved. Because there is only one egress port for the traffic, you can flood the packet and avoid having to learn all the MAC addresses seen on this port. This process saves considerable space in the MAC address table.

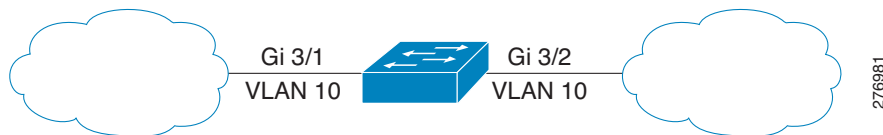
To obtain source learning, packets are bridged as Layer 2 flood packets. Replicated packets use a distinct dedicated bandwidth. Regardless of the number of ports in a flood set, a flood packet always consumes replication packet bandwidth, which consumes some multicast and broadcast packet-processing bandwidth ([Figure 4-2](#)).

Figure 4-2 Disabling MAC Address Learning: Point-to-Point Links

276980

Network Load Balancers

In this topology, you have two devices, one active and one standby. To perform load balancing, both devices must receive all packets. You could place both devices on the same VLAN. If learning can be disabled on this VLAN, the packet is flooded and both devices receive all traffic destined to any MAC address on the VLAN. You also can assign a multicast MAC address to both load balancers to ensure that all packets reach them. (Figure 4-3).

Figure 4-3 Disabling MAC Address Learning: Network Load Balancers

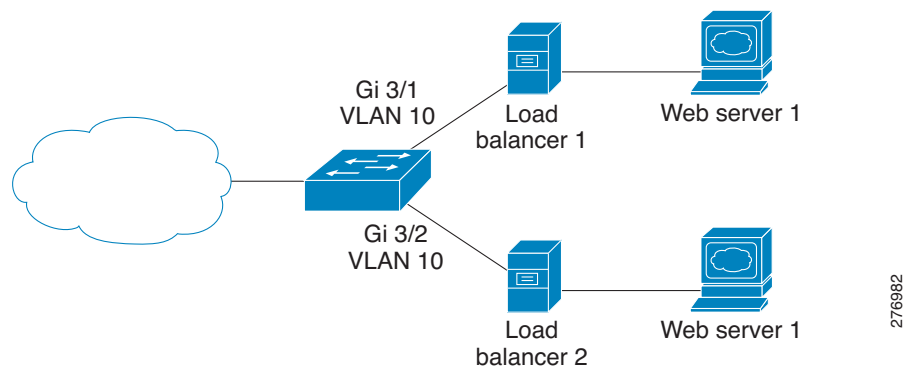
276981

Layer 2 Firewall or Cache

In this topology, a rewritten Layer 3 packet is routed back to a Layer 2 firewall (or cache) before exiting. When the packet reenters the switch from the firewall, it possesses the switch's MAC address because the packet was previously routed. If the ingress port is a switch port, the switch learns the router's MAC address. For a routed port or SVI, however, the switch does not learn the address. Source misses are generated continuously for all arriving data packets and the switch shows a very high CPU utilization.

By disabling learning on the VLAN that the firewall or cache egress is connected to, you will routinely suppress the source miss and do not observe high CPU utilization (Figure 4-4).

Figure 4-4 Disabling MAC Address Learning: Layer 2 Firewall/Cache



Feature Compatibility

The following features are compatible with disabling MAC address learning on a VLAN:

- **EtherChannel**—The learning disable feature has no impact on EtherChannel provided that the MAC learning state is either disabled or enabled for a VLAN on EtherChannel ports.
- **Switch Virtual Interface (SVI, Layer 3 on a VLAN)**— The learning disable feature has no impact on SVI. Although disabling MAC address learning on a SVI VLAN causes flooding, it does not impact any Layer 3 feature.
- **REP**—The learning disable feature has no impact on REP provided that the MAC learning state is either disabled or enabled for an active VLAN on a port where REP is running.
- **Unicast, Multicast, and Broadcast**—When you enable learning on a VLAN, learning is disabled on all types of traffic.
- **DAI, ESMP, and IGMP snooping**— These features do not interact with the learning disable feature.
- **Control packets**— Control packets arrive at the CPU even if learning is disabled.
- **RSPAN**— Learning on a VLAN and on an RSPAN are compatible.
- **VLAN translation**—To disable learning on a VLAN that is being translated, you must disable learning on the translated VLAN.

Feature Incompatibility

The following features are incompatible with disabling MAC address learning and do not work properly when the feature is enabled:

- 802.1X—The 802.1X class of features does not work when learning is disabled because some of these features require source miss, which is ignored.
- Port security—Port security VLANs requires learning to be enabled. To secure MAC addresses, packets must first arrive at the CPU. However, if you disable learning on a VLAN, SA suppression ensures that packets do not operate this way.
- Unicast flood blocking—When unicast flood blocking is enabled on a port, it is removed from the VLAN flood set. If learning is disabled on the same VLAN, the host connected to that port do not receive traffic.
- DHCP snooping—To send the packet out the correct port once a DHCP request has been resolved, DHCP snooping must learn the MAC address. If you disable learning, the switch do not know on which port to exit the packet; the two features are incompatible.
- Broadcast storm control—This feature does not interact with the learning disable feature.
- Flooding of packets in a VLAN domain in which learning is disabled through PVL.

Partial Feature Incompatibility

Although the following features are partially incompatible with disabling MAC address learning, they still retain a large portion of their functionality:

- FlexLink—FlexLink functions and upstream convergence is not impacted. However, downstream fast convergence uses a MAC table to send dummy multicast packets for each learned MAC address upstream to expedite downstream convergence. This situation does not happen if you enabled learning disable. FlexLink downstream convergence occurs naturally, but it is slower if learning is enabled on that VLAN.
- PVLAN—To observe correct behavior, you must disable learning on the primary VLAN and all secondary VLANs associated with the primary VLAN.

**Note**

To avoid confusion, configure PVLAN similarly on both the primary and secondary VLANs in the PVLAN space.

- Spanning Tree (STP)—Except for the UplinkFast feature, per-VLAN spanning tree functionality is not impacted. To achieve faster downstream convergence, UplinkFast forwards dummy multicast packets using learned MAC addresses. This action is not possible unless MAC learning is enabled.

Displaying Address Table Entries

You can display the MAC address table by using one or more of the privileged EXEC commands described in [Table 4-5](#).

Table 4-5 *Commands for Displaying the MAC Address Table*

Command	Description
<code>show ip igmp snooping groups</code>	Displays the Layer 2 multicast entries for all VLANs or the specified VLAN.
<code>show mac address-table address</code>	Displays MAC address table information for the specified MAC address.
<code>show mac address-table aging-time</code>	Displays the aging time in all VLANs or the specified VLAN.
<code>show mac address-table count</code>	Displays the number of addresses present in all VLANs or the specified VLAN.
<code>show mac address-table dynamic</code>	Displays only dynamic MAC address table entries.
<code>show mac address-table interface</code>	Displays the MAC address table information for the specified interface.
<code>show mac address-table notification</code>	Displays the MAC notification parameters and history table.
<code>show mac address-table static</code>	Displays only static MAC address table entries.
<code>show mac address-table vlan</code>	Displays the MAC address table information for the specified VLAN.

Managing the ARP Table

To communicate with a device (over Ethernet, for example), the software first must learn the 48-bit MAC address or the local data link address of that device. The process of learning the local data link address from an IP address is called *address resolution*.

The Address Resolution Protocol (ARP) associates a host IP address with the corresponding media or MAC addresses and the VLAN ID. Using an IP address, ARP finds the associated MAC address. When a MAC address is found, the IP-MAC address association is stored in an ARP cache for rapid retrieval and the IP datagram is encapsulated in a link-layer frame and sent over the network. Encapsulation of IP datagrams and ARP requests and replies on IEEE 802 networks other than Ethernet is specified by the Subnetwork Access Protocol (SNAP). By default, standard Ethernet-style ARP encapsulation (represented by the **arpa** keyword) is enabled on the IP interface.

ARP entries added manually to the table do not age and must be manually removed.

For CLI procedures, see the Cisco IOS Release 12.3 documentation on Cisco.com.

Configuring Embedded CiscoView Support

The Catalyst 4500 series switch supports CiscoView web-based administration using the Catalyst Web Interface (CWI) tool. CiscoView is a device management application that can be embedded on the switch flash and provides dynamic status, monitoring, and configuration information for your switch.

CiscoView displays a physical view of your switch chassis with color-coded modules and ports and monitoring capabilities that display the switch status, performance, and other statistics. Configuration capabilities allow comprehensive changes to devices, if the required security privileges have been granted. The configuration and monitoring capabilities for the Catalyst 4500 series of switches mirror those available in CiscoView in all server-based CiscoWorks solutions, including CiscoWorks LAN Management Solution (LMS) and CiscoWorks Routed WAN Management Solution (RWAN).

These sections describe the Embedded CiscoView support available with Cisco IOS Release 12.1(20)EW and later releases:

- [Understanding Embedded CiscoView, page 4-54](#)
- [Installing and Configuring Embedded CiscoView, page 4-54](#)
- [Displaying Embedded CiscoView Information, page 4-57](#)

Understanding Embedded CiscoView

The Embedded CiscoView network management system is a web-based interface that uses HTTP and SNMP to provide a graphical representation of the switch and to provide a GUI-based management and configuration interface.

Installing and Configuring Embedded CiscoView

To install and configure Embedded CiscoView, perform this task:

	Command	Purpose
Step 1	Switch# dir <i>device_name</i>	Displays the contents of the device. If you are installing Embedded CiscoView for the first time, or if the CiscoView directory is empty, skip to Step 5 .
Step 2	Switch# delete <i>device_name:cv/*</i>	Removes existing files from the CiscoView directory.
Step 3	Switch# squeeze <i>device_name:</i>	Recovers the space in the file system.
Step 4	Switch# copy tftp bootflash	Copies the tar file to bootflash.
Step 5	Switch# archive tar /xtract tftp:// ip address of tftp server/ciscoview.tar device_name:cv	Extracts the CiscoView files from the tar file on the TFTP server to the CiscoView directory.
Step 6	Switch# dir <i>device_name:</i>	Displays the contents of the device. In a redundant configuration, repeat Step 1 through Step 6 for the file system on the redundant supervisor engine.
Step 7	Switch# configure terminal	Enters global configuration mode.
Step 8	Switch(config)# ip http server	Enables the HTTP web server.
Step 9	Switch(config)# snmp-server community string ro	Configures the SNMP password for read-only operation.
Step 10	Switch(config)# snmp-server community string rw	Configures the SNMP password for read/write operation.



Note

The default password for accessing the switch web page is the enable-level password of the switch.

The following example shows how to install and configure Embedded CiscoView on your switch:

```
Switch# dir
Directory of bootflash:/
Directory of bootflash:/
  1  -rw-     9572396  Dec 30 2002 01:05:01 +00:00  cat4000-i9k2s-mz.121-19.EW
  2  -rw-     9604192   Jan 3 2003 07:46:49 +00:00  cat4000-i5k2s-mz.121-19.EW
  3  -rw-     1985024  Jan 21 2003 03:31:20 +00:00  Cat4000IOS.v4-0.tar
  4  -rw-     1910127  Jan 23 2003 04:23:39 +00:00  cv/Cat4000IOS-4.0.sgz
  5  -rw-        7258  Jan 23 2003 04:23:46 +00:00  cv/Cat4000IOS-4.0_ace.html
  6  -rw-        405   Jan 23 2003 04:23:46 +00:00  cv/Cat4000IOS-4.0_error.html
  7  -rw-       2738   Jan 23 2003 04:23:46 +00:00  cv/Cat4000IOS-4.0_install.html
  8  -rw-     20450   Jan 23 2003 04:23:46 +00:00  cv/Cat4000IOS-4.0_jks.jar
  9  -rw-     20743   Jan 23 2003 04:23:46 +00:00  cv/Cat4000IOS-4.0_nos.jar
 10  -rw-     12383   Jan 23 2003 04:23:46 +00:00  cv/applet.html
 11  -rw-        529   Jan 23 2003 04:23:46 +00:00  cv/cisco.x509
 12  -rw-     2523   Jan 23 2003 04:23:46 +00:00  cv/identitydb.obj
 13  -rw-       1173  Mar 19 2003 05:50:26 +00:00  post-2003.03.19.05.50.07-passed.txt

32578556 bytes total (38199688 bytes free)
Switch#
Switch# del cv/*
Delete filename [cv/*]?
Delete bootflash:cv/Cat4000IOS-4.0.sgz? [confirm]y
Delete bootflash:cv/Cat4000IOS-4.0_ace.html? [confirm]y
Delete bootflash:cv/Cat4000IOS-4.0_error.html? [confirm]y
Delete bootflash:cv/Cat4000IOS-4.0_install.html? [confirm]y
Delete bootflash:cv/Cat4000IOS-4.0_jks.jar? [confirm]y
Delete bootflash:cv/Cat4000IOS-4.0_nos.jar? [confirm]y
Delete bootflash:cv/applet.html? [confirm]y
Delete bootflash:cv/cisco.x509? [confirm]y
Delete bootflash:cv/identitydb.obj? [confirm]y
Switch#

Switch# squeeze bootflash:
All deleted files will be removed. Continue? [confirm]y
Squeeze operation may take a while. Continue? [confirm]y
Squeeze of bootflash complete
Switch#
Switch# copy tftp bootflash
Address or name of remote host []? 10.5.5.5
Source filename []? Cat4000IOS.v5-1.tar
Destination filename [Cat4000IOS.v5-1.tar]?
Accessing tftp://10.5.5.5/Cat4000IOS.v5-1.tar...
Loading Cat4000IOS.v5-1.tar from 10.5.5.5 (via FastEthernet2/1):
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 2031616 bytes]

2031616 bytes copied in 11.388 secs (178400 bytes/sec)
Switch#
Switch# dir
Directory of bootflash:/

Directory of bootflash:/
  1  -rw-     9572396  Dec 30 2002 01:05:01 +00:00  cat4000-i9k2s-mz.121-19.EW
  2  -rw-     9604192   Jan 3 2003 07:46:49 +00:00  cat4000-i5k2s-mz.121-19.EW
  3  -rw-     1985024  Jan 21 2003 03:31:20 +00:00  Cat4000IOS.v4-0.tar
  4  -rw-        1173  Mar 19 2003 05:50:26 +00:00  post-2003.03.19.05.50.07-passed.txt
  5  -rw-     2031616  Mar 26 2003 05:33:12 +00:00  Cat4000IOS.v5-1.tar

32578556 bytes total (38199688 bytes free)
```

```

Switch#
Switch# archive tar /xtract Cat4000IOS.v5-1.tar /cv
extracting Cat4000IOS-5.1.sgz (1956591 bytes)
extracting Cat4000IOS-5.1_ace.html (7263 bytes)
extracting Cat4000IOS-5.1_error.html (410 bytes)
extracting Cat4000IOS-5.1_install.html (2743 bytes)
extracting Cat4000IOS-5.1_jks.jar (20450 bytes)
extracting Cat4000IOS-5.1_nos.jar (20782 bytes)
extracting applet.html (12388 bytes)
extracting cisco.x509 (529 bytes)
extracting identitydb.obj (2523 bytes)
Switch#
Switch# dir

Directory of bootflash:/
 1  -rw-      9572396  Dec 30 2002 01:05:01 +00:00  cat4000-i9k2s-mz.121-19.EW
 2  -rw-      9604192   Jan 3 2003 07:46:49 +00:00  cat4000-i5k2s-mz.121-19.EW
 3  -rw-      1985024   Jan 21 2003 03:31:20 +00:00  Cat4000IOS.v4-0.tar
 4  -rw-         1173   Mar 19 2003 05:50:26 +00:00  post-2003.03.19.05.50.07-passed.txt
 5  -rw-      2031616   Mar 26 2003 05:33:12 +00:00  Cat4000IOS.v5-1.tar
 6  -rw-      1956591   Mar 26 2003 05:36:11 +00:00  cv/Cat4000IOS-5.1.sgz
 7  -rw-         7263   Mar 26 2003 05:36:19 +00:00  cv/Cat4000IOS-5.1_ace.html
 8  -rw-         410   Mar 26 2003 05:36:19 +00:00  cv/Cat4000IOS-5.1_error.html
 9  -rw-         2743   Mar 26 2003 05:36:19 +00:00  cv/Cat4000IOS-5.1_install.html
10  -rw-         20450   Mar 26 2003 05:36:19 +00:00  cv/Cat4000IOS-5.1_jks.jar
11  -rw-         20782   Mar 26 2003 05:36:19 +00:00  cv/Cat4000IOS-5.1_nos.jar
12  -rw-         12388   Mar 26 2003 05:36:19 +00:00  cv/applet.html
13  -rw-          529   Mar 26 2003 05:36:19 +00:00  cv/cisco.x509
14  -rw-         2523   Mar 26 2003 05:36:19 +00:00  cv/identitydb.obj

32578556 bytes total (7358284 bytes free)

Switch#
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# ip http server
Switch(config)# snmp-server community public ro
Switch(config)# snmp-server community public rw
Switch(config)# exit
Switch# wr
Building configuration...
Compressed configuration from 2735 bytes to 1169 bytes[OK]
Switch# show ciscoview ?
  package  ADP Package Details
  version  ADP version
  |         Output modifiers
  <

```

For more information about web access to the switch, refer to the “Using the Cisco Web Browser” chapter in the *Cisco IOS Configuration Fundamentals Configuration Guide* at this URL:

http://www.cisco.com/en/US/docs/ios/fundamentals/configuration/guide/12_4t/cf_12_4t_book.html

Displaying Embedded CiscoView Information

To display the Embedded CiscoView information, enter the following commands:

Command	Purpose
Switch# show ciscoview package	Displays information about the Embedded CiscoView files.
Switch# show ciscoview version	Displays the Embedded CiscoView version.

The following example shows how to display the Embedded CiscoView file and version information:

```
Switch# show ciscoview package
File source:
CVFILE                                SIZE(in bytes)
-----
Cat4000IOS-5.1.sgz                    1956591
Cat4000IOS-5.1_ace.html                7263
Cat4000IOS-5.1_error.html              410
Cat4000IOS-5.1_install.html            2743
Cat4000IOS-5.1_jks.jar                 20450
Cat4000IOS-5.1_nos.jar                 20782
applet.html                           12388
cisco.x509                             529
identitydb.obj                         2523

Switch# show ciscoview version
Engine Version: 5.3.4 ADP Device: Cat4000IOS ADP Version: 5.1 ADK: 49
Switch#
```




Configuring Virtual Switching Systems

This chapter describes how to configure a virtual switching system (VSS) on Catalyst 4500 series switches with Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, and 7-E, and Catalyst 4500-X). VSS is supported on Cisco Release IOS XE 3.4.0SG and later.



Note

For complete syntax and usage information for virtual switch commands used in this chapter, see the publication at this location:

[Cisco IOS Virtual Switch Command Reference](#)

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

This chapter consists of these sections:

- [Restrictions for Virtual Switching Systems, page 5-1](#)
- [Understanding Virtual Switching Systems, page 5-2](#)
- [VSS Configuration Guidelines and Restrictions, page 5-28](#)
- [Configuring a VSS, page 5-30](#)
- [In-Service Software Upgrade \(ISSU\) on a VSS, page 5-56](#)
- [License Upgrade on a VSS, page 5-85](#)

Restrictions for Virtual Switching Systems

Virtual Switching System (VSS) requires all the supervisors to have the same license level. For the In-chassis Standby supervisors to be capable of Route Processor Redundancy (RPR) their licenses should match those on the In-chassis Active supervisors. For VSS to function, install the same license levels on the In-chassis Standby supervisors as the In-chassis Active supervisors.

Understanding Virtual Switching Systems

These sections describe a VSS:

- [VSS Overview, page 5-2](#)
- [VSS Redundancy, page 5-10](#)
- [Multichassis EtherChannels, page 5-13](#)
- [Packet Handling, page 5-15](#)
- [System Monitoring, page 5-19](#)
- [Dual-Active Detection, page 5-23](#)
- [Configuring a Recovery IP Address, page 5-25](#)
- [VSS Initialization, page 5-26](#)

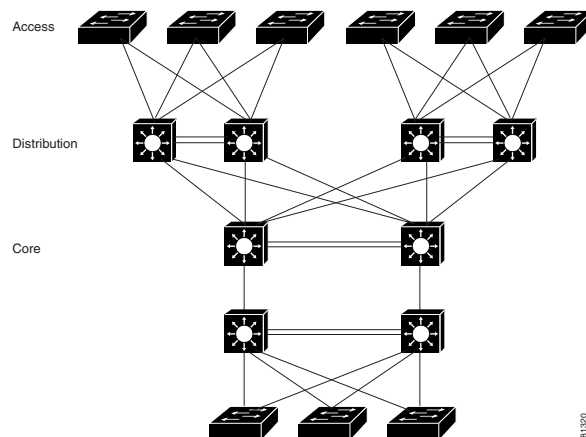
VSS Overview

Network operators increase network reliability by configuring switches and by provisioning links to the redundant pairs. [Figure 5-1](#) shows a typical switch network configuration. Redundant network elements and redundant links can add complexity to network design and operation. Virtual switching simplifies the network by reducing the number of network elements and hiding the complexity of managing redundant switches and links.

A VSS combines a pair of Catalyst 4500 or 4500-X series switches into a single network element. The VSS manages the redundant links, which externally act as a single port channel.

The VSS simplifies network configuration and operation by reducing the number of Layer 3 routing neighbors and by providing a loop-free Layer 2 topology.

Figure 5-1 Typical Switch Network Design



The following sections present an overview of the VSS. These topics are covered in detail in subsequent chapters:

- [Key Concepts, page 5-3](#)
- [VSS Functionality, page 5-5](#)

- [Hardware Requirements, page 5-8](#)
- [Understanding VSL Topology, page 5-10](#)

Key Concepts

The VSS incorporates the following key concepts:

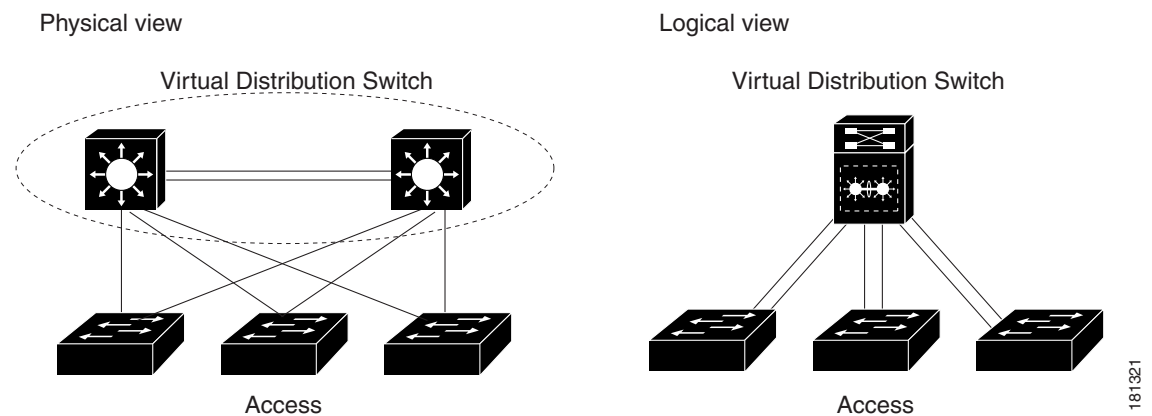
- [Virtual Switching System, page 5-3](#)
- [VSS Active and VSS Standby Switch, page 5-3](#)
- [Virtual Switch Link, page 5-4](#)
- [Multichassis EtherChannel, page 5-5](#)

Virtual Switching System

A VSS combines a pair of switches into a single network element. For example, a VSS in the distribution layer of the network interacts with the access and core networks as if it were a single switch. See [Figure 5-2](#).

An access switch connects to both switches of the VSS using one logical port channel. The VSS manages redundancy and load balancing on the port channel. This capability enables a loop-free Layer 2 network topology. The VSS also simplifies the Layer 3 network topology by reducing the number of routing peers in the network.

Figure 5-2 VSS in the Distribution Network



VSS Active and VSS Standby Switch

When you create or restart a VSS, the peer switches negotiate their roles. One switch becomes the VSS Active switch, and the other switch becomes the VSS Standby switch.

The VSS Active controls the VSS, running the Layer 2 and Layer 3 control protocols for the switching modules on both switches. The VSS Active switch also provides management functions for the VSS, such as module online insertion and removal (OIR) and the console interface.

The VSS Active and VSS Standby switches perform packet forwarding for ingress data traffic on their locally hosted interfaces. However, the VSS Standby switch sends all control traffic to the VSS Active switch for processing.

Virtual Switch Link

For the two switches of the VSS to act as one network element, they need to share control information and data traffic.

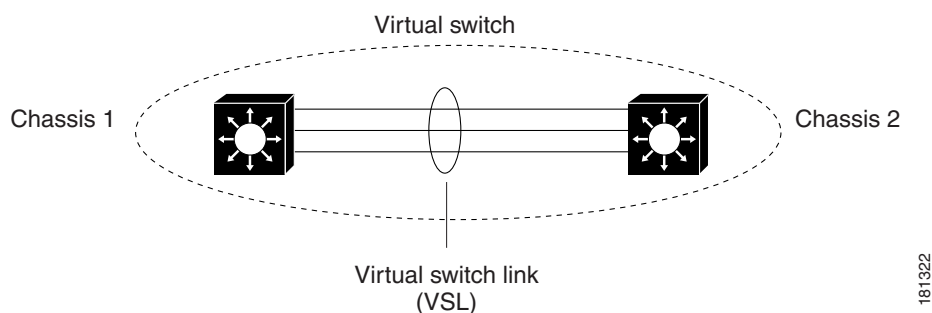
The virtual switch link (VSL) is a special link that carries control and data traffic between the two switches of a VSS, as shown in [Figure 5-3](#). The VSL is implemented as an EtherChannel with up to eight links. The VSL gives control and management traffic higher priority than data traffic so that control and management messages are never discarded. Data traffic is load balanced among the VSL links by the EtherChannel load-balancing algorithm.



Note

EtherChannel load balancing method is a global configuration; VSL observes that method of load balancing.

Figure 5-3 Virtual Switch Link



When you configure VSL, all existing configurations are removed from the interface except for specific allowed commands. When you configure VSL, the system puts the interface into a restricted mode. This means that only specific configuration commands can be configured on the interface.

The following VSL configuration commands are inserted automatically on all VSL member ports:

- **switchport mode trunk**
- **switchport nonegotiate**
- **no lldp transmit**
- **no lldp receive**
- **no cdp enable**
- **service-policy output VSL-Queuing-Policy**

In VSL restricted mode, only these configuration commands are available:

- **channel-group**
- **default**
- **description**
- **exit**
- **load-interval**
- **logging**
- **no**
- **power**

- **service-policy**
- **shutdown**

Multichassis EtherChannel

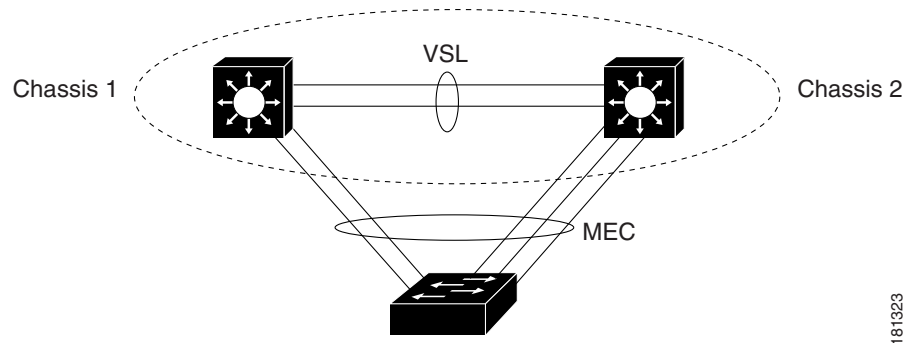


Note

Beginning with Cisco Release IOS XE 3.5.0E and IOS 15.2(1)SG, Layer 3 MEC is supported on the Catalyst 4500 series switch. Cisco Release IOS XE 3.4.0SG does not support Layer 3 MEC.

An EtherChannel (also known as a port channel) is a collection of two or more physical links that combine to form one logical link. Layer 2 protocols operate on the EtherChannel as a single logical entity. A VSS enables the creation of Multi-Chassis EtherChannel (MEC), which is an Etherchannel whose member ports can be distributed across the member switches in a VSS. Because non-VSS switches connected to a VSS view the MEC as a standard EtherChannel, non-VSS switches can connect in a dual homed manner. [Figure 5-4](#) displays a dual-homed connection for an MEC into the VSS; VSS is seen as a single logical switch. Traffic traversing an MEC can be load balanced locally within a VSS member switch much as in standard EtherChannels. Cisco MEC supports the bundling protocols LACP and PAGP as well as ON mode.

Figure 5-4 VSS with MEC



VSS supports a maximum of 256 EtherChannels. This limit applies to the total number of regular EtherChannels and MECs. Because the VSL requires two EtherChannel numbers (one for each switch in the VSS), there are 254 user-configurable EtherChannels.

For information on how to configure Layer 3 Multichassis EtherChannels, see [For information on how to configure Layer 3 Multichassis EtherChannels, see, page 5-5](#)

VSS Functionality

The following sections describe the main functionality of a VSS:

- [Redundancy and High Availability, page 5-6](#)
- [Packet Handling, page 5-6](#)

- [System Management, page 5-6](#)
- [Quad-Supervisor \(In-chassis Standby Supervisor Engine\) Support, page 5-6](#)
- [Asymmetric chassis support, page 5-7](#)
- [Interface Naming Convention, page 5-7](#)
- [Module Number Convention, page 5-7](#)
- [Key Software Features not Supported on VSS, page 5-8](#)

Redundancy and High Availability

In a VSS, supervisor engine redundancy operates between the VSS Active and VSS Standby switch, using stateful switchover (SSO) and nonstop forwarding (NSF). The peer switch exchange configuration and state information across the VSL and the VSS Standby supervisor engine runs in SSO-HOT mode.

The VSS Standby switch monitors the VSS Active switch using the VSL. If it detects failure, the VSS Standby switch initiates a switchover and takes on the VSS Active role. When the failed switch recovers, it takes on the VSS Standby role.

If either the VSS Active switch fails or all links that belong to the VSL port-channel fail, the VSS Standby switch initiates a switchover and assumes the role of the VSS Active switch. If the previous VSS Active switch has failed, it reloads and boots as the VSS Standby switch. However, if only the VSL port-channel failure caused the switchover, the previous VSS Active switch enters recovery mode (provided dual-active detection is configured). In this scenario, the previous VSS Active chassis (now in recovery mode) carries no traffic and only monitors the VSL link. When one link in the VSL port-channel is up, the recovery mode switch reloads and boots as a VSS Standby chassis. For additional information about dual-active detection, see the [“Dual-Active Detection” section on page 5-23](#).

Packet Handling

The VSS Active supervisor engine runs the Layer 2 and Layer 3 protocols and features for the VSS and manages all ports on both switches.

The VSS uses VSL to communicate protocol and system information between the peer switches and to carry data traffic between the switches when required.

Both switches perform packet forwarding for ingress traffic on their interfaces. If possible, ingress traffic is forwarded to an outgoing interface on the same switch to minimize data traffic that must traverse the VSL.

System Management

The VSS Active supervisor engine acts as a single point of control for the VSS. For example, the VSS Active supervisor engine handles OIR of switching modules on both switches. The VSS Active supervisor engine uses VSL to send messages to and from local ports on the VSS Standby switch.

The command console on the VSS Active supervisor engine is used to control both switches. In virtual switch mode, the command console on the VSS Standby supervisor engine blocks attempts to enter configuration mode.

The VSS Standby switch runs a subset of system management tasks. For example, the VSS Standby switch handles its own power management, linecard bringup, and other local hardware management.

Quad-Supervisor (In-chassis Standby Supervisor Engine) Support

Beginning with IOS XE release 3.8.0E, Cisco Catalyst 4500-E switches configured with Supervisor Engines 7-E, 7-LE, or 8E support Quad-Supervisor VSS Mode.

With Quad-Supervisor VSS mode, each chassis in the VSS supports a redundant supervisor engine, called the in-chassis standby (ICS). The ICS supervisor engines use Route Processor Redundancy (RPR) mode, and remain in RPR Standby Cold redundancy state. If the Active supervisor fails, the ICS supervisor boots fully, and becomes the Active supervisor within the chassis, while the local chassis remains nonoperational till SSO redundancy is established with the VSS Active supervisor.

The following table displays a matrix of the chassis that support Quad-Supervisor VSS mode, and the corresponding number of supervisors required in each case.

Chassis	4507R+E	4507R-E	4510R-E	4510R+E
4503-E	3	3	3	3
4506-E	3	3	3	3
4507R+E	4	4	4	4
457R-E	4	4	4	4
4510R+E	4	4	4	4
4510R+E	4	4	4	4

Asymmetric chassis support

Ensure that both participating switches in the VSS have the same supervisor engine type. The chassis can differ in type (i.e., +E and -E chassis can be in a single VSS) and also can differ in the number of slots in chassis. VSS cannot be formed between different flavors of Catalyst 4500X (e.g., 4500X-16 and 4500X-32).

Interface Naming Convention

In VSS mode, interfaces are specified using the switch number (in addition to slot and port), because the same slot numbers are used on both chassis. For example, the **interface 1/5/4** command specifies port 4 of the switching module in slot 5 of switch 1. The **interface 2/5/4** command specifies port 4 on the switching module in slot 5 of switch 2.

Module Number Convention

IOS treats modules in both chassis as if they belong to one single chassis and the module number space is 1-20.

Switch 1 receives a module number from 1-10 and switch 2 receives a number from 11-20, irrespective the chassis type, supervisor type, or number of slots in a chassis. For example, on a 3-slot chassis VSS, the module numbers on switch 1 would be 1, 2, and 3, and on switch 2, the numbers would be 11, 12, and 13. The module number on switch 2 always starts from 11.

The **show switch virtual slot-map** command provides virtual to physical slot mapping. The following is a sample output:

Virtual Slot No	Remote Switch No	Physical Slot No	Module Uptime
1	1	1	00:24:14
2	1	2	00:23:46
3	1	3	-
4	1	4	-

5	1	5	-
6	1	6	-
7	1	7	-
8	1	8	-
9	1	9	-
10	1	10	-
11	2	1	00:22:03
12	2	2	00:24:43
13	2	3	00:24:43
14	2	4	-
15	2	5	-
16	2	6	-
17	2	7	-
18	2	8	-
19	2	9	-
20	2	10	-

Key Software Features not Supported on VSS

With some exceptions, the VSS maintains feature parity with the standalone Catalyst 4500 or 4500-X series switches. Major exceptions include:

- CFM D8.1
- Energywise
- Mediatrace (Medianet active video monitoring feature)
- Metadata (Medianet feature)
- Per VLAN Learning
- UDE
- UDLR
- VMPS Client

Hardware Requirements

The following sections describe the hardware requirements of a VSS:

- [Chassis and Modules, page 5-8](#)
- [VSL Hardware Requirements, page 5-9](#)
- [Multichassis EtherChannel Requirements, page 5-10](#)

Chassis and Modules

[Table 5-1](#) describes the hardware requirements for the VSS chassis and modules.

Table 5-1 VSS Hardware Requirements

Hardware	Count	Requirements
Chassis	2	VSS is available on a Catalyst 4500-X switch and on chassis that support Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, and 7-E. Note +E and -E chassis can be mixed; R-E chassis are not supported with Supervisor Engine 9-E
Supervisor Engines	2	VSS is available on Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, and 7-E, and on the Catalyst 4500-X switch series. All supervisor engines or systems in a VSS must match precisely.
Linecard	0 to as many linecard slots are available in a chassis.	WS-X4748-12X48U+E WS-X4712-SFP-E WS-X4724-SFP-E WS-X4748-SFP-E WS-X4748-RJ45V+E WS-X4712-SFP+E WS-X4640-CSFP-E WS-X4748-UPOE+E WS-X4748-RJ45-E WS-X4606-X2-E WS-X4648-RJ45V-E WS-X4648-RJ45V+E WS-X4648-RJ45-E WS-X4624-SFP-E WS-X4612-SFP-E WS-X4548-RJ45V+ WS-X4448-GB-SFP WS-X4306-GB WS-X4248-RJ45V WS-X4248-FE-SFP WS-X4148-RJ WS-X4148-FX-MT

VSL Hardware Requirements

The VSL EtherChannel supports both 10-Gigabit Ethernet ports and 1- Gigabit Ethernet ports.

We recommend that you use at least two of the 10-Gigabit/1-Gigabit Ethernet ports to create the VSL between the two switches. You cannot combine 10-Gigabit and 1-Gigabit Ethernet ports in a VSL port-channel.

Be aware of the following:

- You can add additional physical links to the VSL EtherChannel with the 10-Gigabit Ethernet ports on any supported supervisor engine or linecard.
- Oversubscribed linecard ports can be used for VSL but total bandwidth requirements of VSL or any traffic drop because of a certain hashing mechanism must be accounted for before using oversubscribed linecard ports for VSL.
- VSL ports can have only 10 Gigabit Ethernet port mode on a WS-X4606-X2-E linecard; non-VSL ports can be configured as 10 or 1 Gigabit Ethernet port mode.
- 1 Gigabit Ethernet ports on line card X4606-X2-E cannot be used as VSL links.

Multichassis EtherChannel Requirements

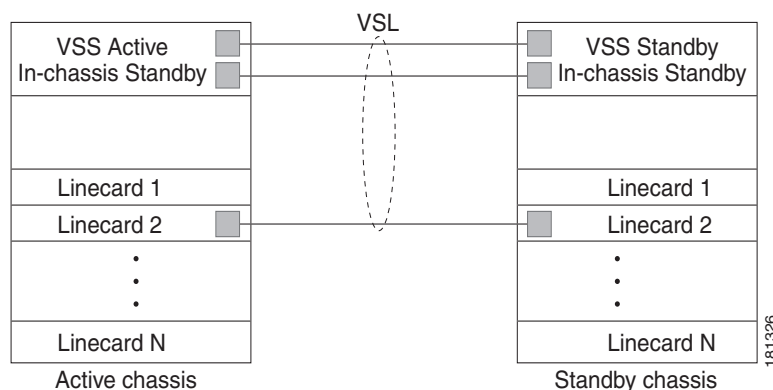
Physical links from any of the supervisor engines or linecard modules can be used to implement a Multichassis EtherChannel (MEC).

Understanding VSL Topology

A VSS contains two switches that communicate using the VSL, which is a special port group.

We recommend that you configure at least two of the 10-Gigabit/1-Gigabit Ethernet ports as VSL, selecting ports from different modules. [Figure 5-5](#) shows a example topology.

Figure 5-5 VSL Topology Example



VSS Redundancy

The following sections describe how redundancy in a VSS supports network high availability:

- [Overview, page 5-11](#)
- [RPR and SSO Redundancy, page 5-11](#)
- [Switch Roles in a VSS, page 5-12](#)
- [Failed Switch Recovery, page 5-12](#)
- [VSL Failure, page 5-13](#)
- [User Actions, page 5-13](#)

Overview

A VSS operates stateful switchover (SSO) between the VSS Active and VSS Standby supervisor engines. Compared to standalone mode, a VSS has the following important differences in its redundancy model:

- The VSS Active and VSS Standby supervisor engines are hosted in separate switches and use the VSL to exchange information.
- The VSS Active supervisor engine controls both switches of the VSS. The VSS Active supervisor engine runs the Layer 2 and Layer 3 control protocols and manages the switching modules on both switches.
- The VSS Active and VSS Standby switches perform data traffic forwarding.

If the VSS Active supervisor engine fails, the VSS Standby supervisor engine initiates a switchover and assumes the VSS Active role.

RPR and SSO Redundancy

Beginning in IOS XE release 3.8.0E, Quad-Supervisor VSS mode supports intra-chassis redundancy. When the VSS Active fails, the in-chassis standby supervisor operates in RPR mode and becomes the VSS Standby.

A VSS operates with stateful switchover (SSO) redundancy if it meets the following requirements:

- Both supervisor engines must be running the same software version, unless it is in the process of software upgrade.

**Note**

If the supervisors are running different software versions, the system will reach SSO only if the two versions are ISSU-compatible. However, this is not supported in the current release.

- VSL-related configuration in the two switches must match.
- SSO and nonstop forwarding (NSF) must be configured on each switch.

**Note**

See the [“SSO Dependencies” section on page 5-26](#) for additional details about the requirements for SSO redundancy on a VSS. See [Chapter 13, “Configuring Cisco NSF with SSO Supervisor Engine Redundancy”](#) for information about configuring SSO and NSF.

With SSO redundancy, the VSS Standby supervisor engine is always ready to assume control following a fault on the VSS Active supervisor engine. Configuration, forwarding, and state information are synchronized from the VSS Active supervisor engine to the redundant supervisor engine at startup and whenever changes to the VSS Active supervisor engine configuration occur. If a switchover occurs, traffic disruption is minimized.

If a VSS does not meet the requirements for SSO redundancy, it will be incapable of establishing a relationship with the peer switch.

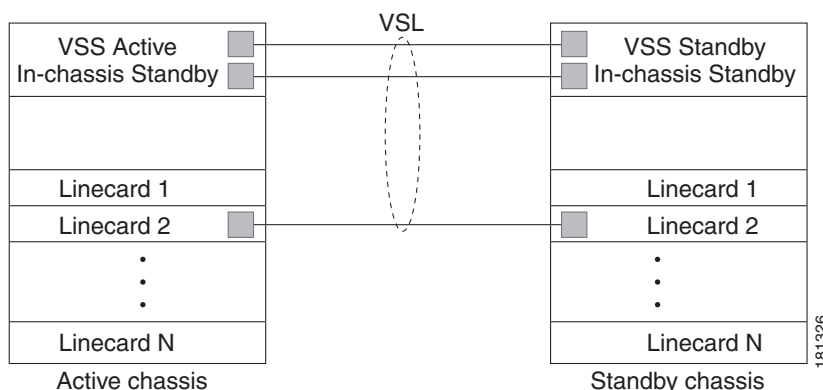
The VSS runs stateful switchover (SSO) between the VSS Active and VSS Standby supervisor engines (see [Figure 5-6](#)). The VSS determines the role of each supervisor engine during initialization.

The supervisor engine in the VSS Standby switch runs in hot standby state. The VSS uses the VSL link to synchronize configuration data from the VSS Active to the VSS Standby supervisor engine. Also, protocols and features that support high availability synchronize their events and state information to the VSS Standby supervisor engine.

Switch Roles in a VSS

Figure 5-6 illustrates the switches' roles in a VSS.

Figure 5-6 Switches' Roles in a VSS



Failed Switch Recovery

If the VSS Active switch or supervisor engine fails, the VSS initiates a stateful switchover (SSO) and the former VSS Standby supervisor engine assumes the VSS Active role. The failed switch performs recovery action by reloading the supervisor engine.

In Quad-Supervisor VSS mode, if the VSS Active switch or supervisor engine fails, the VSS initiates a stateful switchover (SSO) and the former VSS Standby supervisor engine assumes the VSS Active role. The in-chassis standby (ICS) on the failed switch becomes the VSS Standby and the former VSS Active becomes the ICS for the VSS Standby.

If the VSS Standby switch or supervisor engine fails, no switchover is required. The failed switch performs recovery action by reloading the supervisor engine.

The VSL links are unavailable while the failed switch recovers. After the switch reloads, it becomes the new VSS Standby switch and the VSS reinitializes the VSL links between the two switches.

The switching modules on the failed switch are unavailable during recovery, so the VSS operates only with the MEC links that terminate on the VSS Active switch. The bandwidth of the VSS is reduced until the failed switch has completed its recovery and become operational again. Any devices that are connected only to the failed switch experience an outage.



Note

The VSS may experience a brief data path disruption when the switching modules in the VSS Standby switch become operational after the SSO.

After the SSO, much of the processing power of the VSS Active supervisor engine is consumed in bringing up a large number of ports simultaneously in the VSS Standby switch. As a result, some links might be brought up before the supervisor engine has configured forwarding for the links, causing traffic to those links to be lost until the configuration is complete. This condition is especially disruptive if the link is an MEC link and it is running in "ON" mode. This is why it is recommended that MEC ports always have either PAGP or LACP mode of EtherChannel configured.

**Note**

We recommend not configuring LACP independent mode (standalone-mode) for MEC because ports on the VSS Standby switch (while it boots) come up tens of seconds before the control plane is fully functional. This behavior causes a port to start working in independent mode and might cause traffic loss until the port is bundled.

VSL Failure

To ensure fast recovery from VSL failures, fast link failure detection is enabled in virtual switch mode on all VSL port channel members.

**Note**

Fast link notification is based upon internal hardware assisted BFD sessions between the pair of physical VSL links.

If a single VSL physical link goes down, the VSS adjusts the port group so that the failed link is not selected.

If the VSS Standby switch detects complete VSL link failure, it initiates a stateful switchover (SSO). If the VSS Active switch has failed (causing the VSL links to go down), the scenario is switch failure, as described in the previous section.

If only the VSL has failed and the VSS Active switch is still operational, this is a dual-active scenario. The VSS detects that both switches are operating in VSS Active mode and performs recovery action. See the [“Dual-Active Detection” section on page 5-23](#) for additional details about the dual-active scenario.

User Actions

From the VSS Active switch command console, you can initiate a VSS switchover or a reload.

If you enter the **reload** command from the command console, it performs a reload on the switch where reload is issued.

To reload only the VSS Standby switch, use the **redundancy reload peer** command.

To force a switchover from the VSS Active to the VSS Standby supervisor engine, use the **redundancy force-switchover** command.

To reset both the VSS Active and Standby switch, use the **redundancy reload shelf** command.

To reset the in-chassis standby (ICS), use the **hw-module module <supervisor slot number> reset** command.

Multichassis EtherChannels

These sections describe multichassis EtherChannels (MECs):

- [Overview, page 5-14](#)
- [MEC Failure Scenarios, page 5-14](#)

Overview

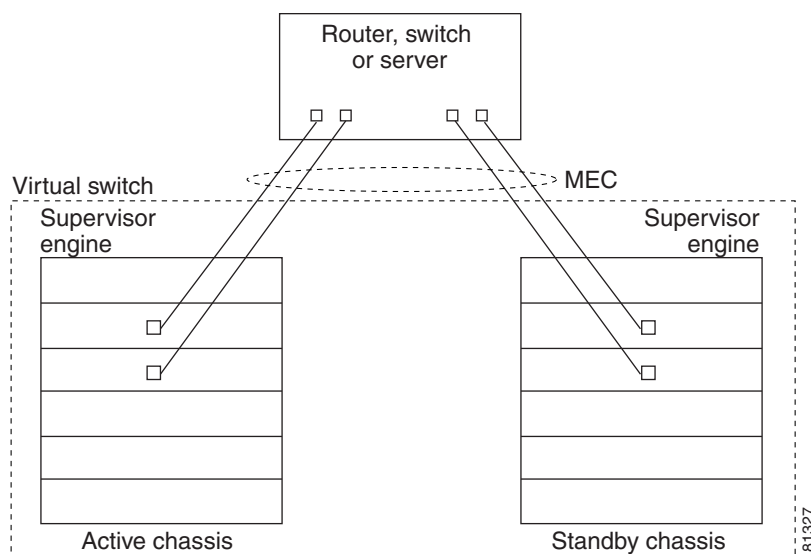
A multichassis EtherChannel is an EtherChannel with ports that terminate on both switches of the VSS (see [Figure 5-7](#)). A VSS MEC can connect to any network element that supports EtherChannel (such as a host, server, router, or switch).

At the VSS, an MEC is an EtherChannel with additional capability: the VSS balances the load across ports in each switch independently. For example, if traffic enters the VSS Active switch, the VSS will select an MEC link from the VSS Active switch. This MEC capability ensures that data traffic does not unnecessarily traverse the VSL.

Each MEC can optionally be configured to support either PAgP or LACP. These protocols run only on the VSS Active switch. PAgP or LACP control packets destined for an MEC link on the VSS Standby switch are sent across VSL.

An MEC can support up to eight physical links, which can be distributed in any proportion between the VSS Active and VSS Standby switch.

Figure 5-7 MEC Topology



MEC Failure Scenarios

We recommend that you configure the MEC with at least one link to each switch. This configuration conserves VSL bandwidth (traffic egress link is on the same switch as the ingress link), and increases network reliability (if one VSS supervisor engine fails, the MEC is still operational).

The following sections describe possible failures and the resulting impacts:

- [Single MEC Link Failure, page 5-15](#)
- [All MEC Links to the VSS Active Switch Fail, page 5-15](#)
- [All MEC Links to the VSS Standby Switch Fail, page 5-15](#)
- [All MEC Links Fail, page 5-15](#)
- [VSS Standby Switch Failure, page 5-15](#)
- [VSS Active Switch Failure, page 5-15](#)

Single MEC Link Failure

If a link within the MEC fails (and other links in the MEC are still operational), the MEC redistributes the load among the operational links, as in a regular port.

All MEC Links to the VSS Active Switch Fail

If all links to the VSS Active switch fail, the MEC becomes a regular EtherChannel with operational links to the VSS Standby switch.

Data traffic terminating on the VSS Active switch reaches the MEC by crossing the VSL to the VSS Standby switch. Control protocols continue to run in the VSS Active switch. Protocol messages reach the MEC by crossing the VSL.

All MEC Links to the VSS Standby Switch Fail

If all links fail to the VSS Standby switch, the MEC becomes a regular EtherChannel with operational links to the VSS Active switch.

Control protocols continue to run in the VSS Active switch. All control and data traffic from the VSS Standby switch reaches the MEC by crossing the VSL to the VSS Active switch.

All MEC Links Fail

If all links in an MEC fail, the logical interface for the EtherChannel is set to unavailable. Layer 2 control protocols perform the same corrective action as for a link-down event on a regular EtherChannel.

On adjacent switches, routing protocols and Spanning Tree Protocol (STP) perform the same corrective action as for a regular EtherChannel.

VSS Standby Switch Failure

If the VSS Standby switch fails, the MEC becomes a regular EtherChannel with operational links on the VSS Active switch. Connected peer switches detect the link failures, and adjust their load-balancing algorithms to use only the links to the VSS Active switch.

In Quad-Supervisor VSS mode, the in-chassis standby (ICS supervisor in the VSS Standby switch becomes the VSS Standby supervisor and the former VSS Standby supervisor becomes the ICS.

VSS Active Switch Failure

VSS Active switch failure results in a stateful switchover (SSO). See the [“VSS Redundancy” section on page 5-10](#) for details about SSO on a VSS. After the switchover, the MEC is operational on the new VSS Active switch. Connected peer switches detect the link failures (to the failed switch), and adjust their load-balancing algorithms to use only the links to the new VSS Active switch.

Packet Handling

In a VSS, the VSS Active supervisor engine runs the Layer 2 and Layer 3 protocols and features for the VSS and manages the ports on both switches.

The VSS uses the VSL to communicate system and protocol information between the peer switches and to carry data traffic between the two switches.

Both switches perform packet forwarding for ingress traffic on their local interfaces. The VSS minimizes the amount of data traffic that must traverse the VSL.

The following sections describe packet handling in a VSS:

- [Traffic on the VSL, page 5-16](#)
- [Layer 2 Protocols, page 5-16](#)
- [Layer 3 Protocols, page 5-18](#)

Traffic on the VSL

The VSL carries data traffic and in-band control traffic between the two switches. All frames forwarded over the VSL link are encapsulated with a special header (up to ten bytes for data traffic and 18 bytes for control packets), which provides information for the VSS to forward the packet on the peer switch.

The VSL transports control messages between the two switches. Messages include protocol messages that are processed by the VSS Active supervisor engine, but received or transmitted by interfaces on the VSS Standby switch. Control traffic also includes module programming between the VSS Active supervisor engine and switching modules on the VSS Standby switch.

The VSS needs to transmit data traffic over the VSL under the following circumstances:

- Layer 2 traffic flooded over a VLAN (even for dual-homed links).
- Packets processed by software on the VSS Active supervisor engine where the ingress interface is on the VSS Standby switch.
- The packet destination is on the peer switch, such as the following examples:
 - Traffic within a VLAN where the known destination interface is on the peer switch.
 - Traffic that is replicated for a multicast group and the multicast receivers are on the peer switch.
 - The known unicast destination MAC address is on the peer switch.
 - The packet is a MAC notification frame destined for a port on the peer switch.

VSL also transports system data, such as NetFlow export data and SNMP data, from the VSS Standby switch to the VSS Active supervisor engine.

To preserve the VSL bandwidth for critical functions, the VSS uses strategies to minimize user data traffic that must traverse the VSL. For example, if an access switch is dual-homed (attached with an MEC terminating on both VSS switches), the VSS transmits packets to the access switch using a link on the same switch as the ingress link.

Traffic on the VSL is load-balanced with the same global hashing algorithms available for EtherChannels (the default algorithm is source-destination IP).

Layer 2 Protocols

The VSS Active supervisor engine runs the Layer 2 protocols (such as STP and VTP) for the switching modules on both switches. Protocol messages that are transmitted and received on the VSS Standby switch switching modules must traverse the VSL to reach the VSS Active supervisor engine.

All Layer 2 protocols in VSS work similarly in standalone mode. The following sections describe the difference in behavior for some protocols in VSS:

- [Spanning Tree Protocol, page 5-17](#)
- [EtherChannel Control Protocols, page 5-17](#)
- [Jumbo frame size restriction, page 5-17](#)
- [SPAN, page 5-17](#)

- [Private VLANs, page 5-17](#)

Spanning Tree Protocol

The VSS Active switch runs Spanning Tree Protocol (STP). The VSS Standby switch redirects STP BPDUs across the VSL to the VSS Active switch.

The STP bridge ID is commonly derived from the chassis MAC address. To ensure that the bridge ID does not change after a switchover, the VSS continues to use the original chassis MAC address for the STP Bridge ID.

EtherChannel Control Protocols

Link Aggregation Control Protocol (LACP) and Port Aggregation Protocol (PAgP) packets contain a device identifier. The VSS defines a common device identifier for both chassis. You should use PAgP or LACP on MECs instead of mode ON, although all three modes are supported.

A new PAgP enhancement has been defined for assisting with dual-active scenario detection. For additional information, see the [“Dual-Active Detection” section on page 5-23](#).

Jumbo frame size restriction

The maximum jumbo frame size supported on a VSS interface is 9188 bytes (MTU of 9170 bytes). This accommodates the overhead of transporting packets between the two member switches over VSL.

Not all frames traverse VSL. So, packets confined to one of the member switches could have a size of 9216 bytes (MTU of 9198 bytes). Such frames may require diversion over VSL when a failure occurs. This is why the *max configured MTU* on non-VSL front panel ports is 9170.



Note The MTU CLI is unavailable on a VSL interface. It is set internally to 9198 (Max frame size of 9216), addressing the overhead of VSL.

For example, if we send traffic between two ports on the active switch, no overhead exists. However, overhead exists when we send packets between ports of active to ports of standby. Even more overhead exists when we send packets from standby ports to the active CPU. The higher limit accommodates the worst case and guarantees consistent forwarding under all scenarios.

SPAN

VSS supports all SPAN features for non-VSL interfaces.



Note SPAN on VSL ports is not supported; VSL ports can be neither a SPAN source, nor a SPAN destination.

The number of SPAN sessions available on a VSS matches that on a single switch running in standalone mode.

Private VLANs

Private VLANs on VSS work similarly in standalone mode. The only exception is that the native VLAN on isolated trunk ports must be configured explicitly. Refer to [Chapter 47, “Configuring Private VLANs”](#) for details on how to configure the native VLAN on isolated trunk ports.

Layer 3 Protocols

The VSS Active supervisor engine runs the Layer 3 protocols and features for the VSS. All layer 3 protocol packets are sent to and processed by the VSS Active supervisor engine. Both member switches perform hardware forwarding for ingress traffic on their interfaces. If possible, to minimize data traffic that must traverse the VSL, ingress traffic is forwarded to an outgoing interface on the same switch. When software forwarding is required, packets are sent to the VSS Active supervisor engine for processing.

The same router MAC address, assigned by the VSS Active supervisor engine, is used for all Layer 3 interfaces on both VSS member switches. After a switchover, the original router MAC address is still used. The router MAC address is configurable and can be chosen from three options: virtual-mac (derived from domainId), chassis-mac (preserved after switchover), and user-configured MAC address. VSS uses virtual MAC address as the default.

The following sections describe Layer 3 protocols for a VSS:

- [IPv4, page 5-18](#)
- [IPv6, page 5-18](#)
- [IPv4 Multicast, page 5-19](#)
- [Software Features, page 5-19](#)

IPv4

The supervisor engine on the VSS Active switch runs the IPv4 routing protocols and performs any required software forwarding. All routing protocol packets received on the VSS Standby switch are redirected to the VSS Active supervisor engine across the VSL. The VSS Active supervisor engine generates all routing protocol packets to be sent out over ports on either VSS member switch.

Hardware forwarding is distributed across both members on the VSS. The supervisor engine on the VSS Active switch sends Forwarding Information Base (FIB) updates to the VSS Standby supervisor engine, which installs all routes and adjacencies in its hardware.

Packets intended for a local adjacency (reachable by local ports) are forwarded locally on the ingress switch. Packets intended for a remote adjacency (reachable by remote ports) must traverse the VSL.

The supervisor engine on the VSS Active switch performs all software forwarding (for protocols such as IPX) and feature processing (such as fragmentation and TTL exceed). If a switchover occurs, software forwarding is disrupted until the new VSS Active supervisor engine obtains the latest CEF and other forwarding information.

In virtual switch mode, the requirements to support non-stop forwarding (NSF) match those in standalone redundant mode of operation.

From a routing peer perspective, Multi-Chassis EtherChannels (MEC) remain operational during a switchover (only the links to the failed switch are down, but the routing adjacencies remain valid).

The VSS achieves Layer 3 load-balancing over all paths in the FIB entries, be it local or remote.

IPv6

VSS supports IPv6 unicast and multicast as it is there on standalone system.

IPv4 Multicast

The IPv4 multicast protocols run on the VSS Active supervisor engine. Internet Group Management Protocol (IGMP) and Protocol Independent Multicast (PIM) protocol packets received on the VSS Standby supervisor engine are transmitted across VSL to the VSS Active supervisor engine. The VSS Active supervisor engine generates IGMP and PIM protocol packets to be sent over ports on either VSS member.

The VSS Active supervisor engine syncs Multicast Forwarding Information Base (MFIB) state to the VSS Standby supervisor engine. On both member switches, all multicast routes are loaded in hardware with replica expansion table (RET) entries programmed for only local outgoing interfaces. Both member switches are capable of performing hardware forwarding.



Note

To avoid multicast route changes as a result of the switchover, we recommend that all links carrying multicast traffic be configured as MEC rather than Equal Cost Multipath (ECMP).

For packets traversing VSL, all Layer 3 multicast replication occurs on the egress switch. If there are multiple receivers on the egress switch, only one packet is replicated and forwarded over the VSL, and then replicated to all local egress ports.

Software Features

Software features run only on the VSS Active supervisor engine. Incoming packets to the VSS Standby switch that require software processing are sent across the VSL to the VSS Active supervisor engine.

System Monitoring

The following sections describe system monitoring and system management for a VSS:

- [Environmental Monitoring, page 5-19](#)
- [File System Access, page 5-19](#)
- [Diagnostics, page 5-20](#)
- [Network Management, page 5-21](#)

Environmental Monitoring

Environmental monitoring runs on both supervisor engines. The VSS Standby switch reports notifications to the VSS Active supervisor engine. The VSS Active switch gathers log messages for both switches. The VSS Active switch synchronizes the calendar and system clock to the VSS Standby switch.

File System Access

File system access on VSS is the same as it is on dual supervisor standalone system. All files on a standby switch are accessible with slave prefix as following:

```
Switch# dir ?
/all          List all files
/recursive    List files recursively
all-filesystems List files on all filesystems
bootflash:    Directory or file name
```

```

cat4000_flash:      Directory or file name
cns:                Directory or file name
crashinfo:          Directory or file name
kinfo:              Directory or file name
null:               Directory or file name
nvram:              Directory or file name
revrcsf:            Directory or file name
slavebootflash:     Directory or file name
slavecat4000_flash: Directory or file name
slavecrashinfo:     Directory or file name
slavekinfo:          Directory or file name
slavenvram:          Directory or file name
slaveslot0:          Directory or file name
slaveusb0:           Directory or file name
slot0:              Directory or file name
system:              Directory or file name
tar:                 Directory or file name
tmpsys:              Directory or file name
usb0:                Directory or file name
|                    Output modifiers

```

All file or directory name with prefix "slave" show vss standby files.

In Quad-Supervisor VSS mode, the following output is displayed:

```

Switch dir ?
/all List all files
/recursive List files recursively
all-filestems List files on
all filestems
bootflash-ics: Directory or file name
bootflash: Directory or file name
cat4000_flash: Directory or file name
cns: Directory or file name
crashinfo: Directory or file name
kinfo: Directory or file name
lcfpga: Directory or file name
null: Directory or file name
nvram: Directory or file name
revrcsf: Directory or file name
slavebootflash-ics: Directory or file name
slavebootflash: Directory or file name
slavecat4000_flash: Directory or file name
slavecrashinfo: Directory or file name
slavekinfo: Directory or file name
slavenvram: Directory or file nameslaveslot0: Directory or file name
slaveusb0: Directory or file nameslot0: Directory or file name
smi: Directory or file name
system: Directory or file name
tar: Directory or file name
tmpsys: Directory or file nameusb0: Directory or file name| Output modifiers

```



Note

The in-chassis standby (ICS) bootflash is displayed as `bootflash-ics:` and `slavebootflash-ics`. The ICS USB0: and `slot0:` is not accessible from the VSS Active switch, and is not displayed in the output.

Diagnostics

Bootup diagnostics are run independently on both switches. Online diagnostics can be invoked on the basis of virtual slots, which provide accessibility to modules on both switches. Use the **show switch virtual slot-map** command to display the virtual to physical slot mapping.

```
Switch# show switch virtual slot-map
Virtual Slot to Remote Switch/Physical Slot Mapping Table:
```

Virtual Slot No	Remote Switch No	Physical Slot No	Module Uptime
1	1	1	-
2	1	2	-
3	1	3	02:43:51
4	1	4	-
5	1	5	-
6	1	6	02:45:20
7	1	7	-
8	1	8	02:43:50
9	1	9	-
10	1	10	-
11	2	1	02:46:50
12	2	2	02:46:50
13	2	3	-
14	2	4	-
15	2	5	02:42:23
16	2	6	-
17	2	7	-
18	2	8	-
19	2	9	-
20	2	10	-

Network Management

The following sections describe network management for a VSS:

- [Telnet over SSH Sessions and the Web Browser User Interface, page 5-21](#)
- [SNMP, page 5-21](#)
- [Command Console, page 5-22](#)
- [Accessing the Remote Console on VSS, page 5-22](#)
- [Copying Files to Bootflash, page 5-22](#)
- [Transferring a Large File over VSL, page 5-23](#)

Telnet over SSH Sessions and the Web Browser User Interface

A VSS supports remote access using Telnet over SSH sessions and the Cisco web browser user interface. All remote access is directed to the VSS Active supervisor engine, which manages the whole VSS. If the VSS performs a switchover, Telnet over SSH sessions and web browser sessions are disconnected.

SNMP

The SNMP agent runs on the VSS Active supervisor engine.

CISCO-VIRTUAL-SWITCH-MIB is a new MIB for virtual switch mode and contains the following main components:

- cvsGlobalObjects — Domain #, Switch #, Switch Mode
- cvsCoreSwitchConfig — Switch Priority
- cvsChassisTable — Switch Role and Uptime

- cvsModuleTable — Information on the physical modules listed in the ENTITY-MIB entPhysicalTable, whose entPhysicalClass is module(9)
- cvsVSLConnectionTable — VSL Port Count, Operational State
- cvsVSLStatsTable — Total Packets, Total Error Packets
- cvsVSLPortStatsTable — TX/RX Good, Bad, Bi-dir and Uni-dir Packets

Command Console

Because the management plane of the two switches are common (that is, both switches in a VSS can be configured and managed from Active switch itself), you do not require access to the Standby console. However, the consoles of both switches are available by connecting console cables to both supervisor engine console ports. Availability of the Standby console does not imply that you can configure the switch from Standby console as well. Config mode is not available on the Standby and **show** commands are limited in availability. Observe that all **show** commands, even for remote ports, are available on the Active switch.

The console on the VSS Standby switch will indicate that switch is operating in VSS Standby mode by adding the characters “-stdby” to the command line prompt. You cannot enter configuration mode on the VSS Standby switch console.

The following example shows the prompt on the VSS Standby console:

```
Switch-standby> sh clock
*14:04:58.705 UTC Tue Nov 20 2012
```

Accessing the Remote Console on VSS



Note

The **remote login** command is not supported on switches running Quad-Supervisor VSS mode.

Remote console (the Standby's console) can be accessed from the Local (Active) switch. This is available on a standalone system and works similarly on VSS. To access the remote console from the Active, you can use the **remote login** command with a VSS-Standby module number. Observe that the module number is a virtual slot and it would be an In-Chassis-Active supervisor module number on the remote chassis.

```
Switch# remote login module 11
Connecting to standby virtual console
Type "exit" or "quit" to end this session

9 Switch-standby-console>
```

Because the Standby console is not available in config mode and only partially available in EXEC mode, distributed features like Netflow and Wireshark have special exemptions for respective commands (that is, these commands are allowed). Refer to [Chapter 76, “Configuring Flexible NetFlow”](#) and [Chapter 71, “Configuring Wireshark”](#) for details.

Copying Files to Bootflash

When you copy a file to a bootflash on the Active, it is not automatically copied to the Standby bootflash. This means that when you perform an ISSU upgrade or downgrade, both switches must receive the files individually. This behavior matches that on a dual-supervisor standalone system. Similarly, the removal of a file on one switch does not cause the removal of the same file on the other switch.

Transferring a Large File over VSL

Because the management plane of the VSS switches are performed through the Active, you might need to send a large-config/image file from one switch to another (that is, sending a file transfer over VSL). When you do this, the VSL link becomes “busy.” Because data is flowing on a front panel port, it [the data] is significantly slower than what you might see on a dual-supervisor standalone system because in the latter, this action occurs through dedicated EOBC link.

On VSS, copying a large file from one switch to another may take several minutes. Hence, you should do this only when needed. Consider a wait of several minutes before file transfer completes.

Dual-Active Detection

If the VSL fails, the VSS Standby switch cannot determine the state of the VSS Active switch. To ensure that switchover occurs without delay, the VSS Standby switch assumes the VSS Active switch has failed and initiates switchover to take over the VSS Active role.

If the original VSS Active switch is still operational, both switch are now VSS Active. This situation is called a *dual-active scenario*. A dual-active scenario can have adverse effects on network stability, because both switches use the same IP addresses, SSH keys, and STP bridge ID. The VSS must detect a dual-active scenario and take recovery action.

The VSS supports the methods, Enhanced PAgP and Fast-Hello, for detecting a dual-active scenario. PAgP uses messaging over the MEC links to communicate between the two switches through a neighbor switch. Enhanced PAgP requires a neighbor switch that supports the PAgP enhancements.

The dual-active detection and recovery methods are described in the following sections:

- [Dual-Active Detection Using Enhanced PAgP, page 5-23](#)
- [Dual-Active Detection Using Fast-Hello, page 5-24](#)
- [Recovery Actions, page 5-24](#)

Dual-Active Detection Using Enhanced PAgP

Port aggregation protocol (PAgP) is a Cisco-proprietary protocol for managing EtherChannels. If a VSS MEC terminates to a Cisco switch, you can run PAgP protocol on the MEC. If PAgP is running on the MECs between the VSS and an upstream or downstream switch, the VSS can use PAgP to detect a dual-active scenario. The MEC must have at least one port on each switch of the VSS.

In virtual switch mode, PAgP messages include a new type length value (TLV) which contains the ID of the VSS Active switch. Only switches in virtual switch mode send the new TLV.

For dual-active detection to operate successfully, one or more of the connected switches must be able to process the new TLV. Catalyst 4500, Catalyst 4500-X, and Catalyst 49xx series switches have this capability. For a list of other Cisco products that support enhanced PAgP, refer to Release Notes for Cisco IOS Release at this URL:

http://www.cisco.com/en/US/products/ps6350/tsd_products_support_series_home.html

When the VSS Standby switch detects VSL failure, it initiates SSO and becomes VSS Active. Subsequent PAgP messages to the connected switch from the newly VSS Active switch contain the new VSS Active ID. The connected switch sends PAgP messages with the new VSS Active ID to both VSS switches.

If the formerly VSS Active switch is still operational, it detects the dual-active scenario because the VSS Active ID in the PAGP messages changes. This switch initiates recovery actions as described in the [“Recovery Actions” section on page 5-24](#).

Dual-Active Detection Using Fast-Hello

Dual-Active fast-hello employs fast-hello Layer 2 messages over a direct Ethernet connection. When the VSL goes down, the event is communicated to the peer switch. If the switch was operating as the active before the VSL went down, it goes into recovery mode upon receipt of a VSL down indication from the peer switch. This method is faster than IP BFD and ePAGP and does not require a neighboring switch.

Fast-Hello Link

A fast-hello link is configured between two VSS members with the intention of detecting a dual-active condition. Configuring dual-active fast-hello automatically removes all configurations from the specified interfaces, and restricts the interface to dual-active configuration commands. The following commands are allowed only in restricted mode on a fast-hello interface:

- default**—Sets a command to its defaults
- description**—Describes the interface
- dual-active**—Specifies a virtual switch dual-active config
- exit**—Exits from the fast hello interface configuration mode
- load-interval**—Specifies the interval for load calculation on an interface
- logging**—Configures logging for interface
- no**—Negates a command or set its defaults
- shutdown**—Shuts down the selected interface

No data traffic other than fast-hello can be used by fast-hello links.

For details on how to configure fast-hello dual-active detection, see the [“Configuring Fast-Hello Dual-Active Detection” section on page 5-53](#).

Recovery Actions

An VSS Active switch that detects a dual-active condition shuts down (by err-disabling) all of its non-VSL interfaces to remove itself from the network, and waits in recovery mode until the VSL links have recovered. You might need to intervene directly to fix the VSL failure. When the shut down switch detects that VSL is operational again, the switch reloads and returns to service as the VSS Standby switch.

Loopback interfaces are also shut down in recovery mode. The loopback interfaces are operationally down and not err-disabled.



Note

If the running configuration of the switch in recovery mode has been changed without saving, the switch will not automatically reload. In this situation, you must write the configuration to memory and then reload manually using the **reload** command. Only configuration changes applied to VSL ports on the switch can be saved. All other configuration changes are discarded as the node reboots as VSS standby.

When a switch becomes active (either due to dual-active scenario or otherwise), the IP address configured for fa1 management interface is associated with the active switch. By default, the switch in recovery mode will not have any IP address for the fa1 interface on its supervisor engine. To ensure IP connectivity to the switch during recovery, you can configure a recovery IP address. (IP address configuration is mandatory if you want IP connectivity while switch is in recovery.) When a switch enters recovery mode, the IP address for the management interface on its supervisor engine is associated with the recovery IP address.

The recovery IP address for a management interface can be verified in the output of commands such as **show ip interface brief** and **show interfaces**.

Configuring a Recovery IP Address

The recovery IP address is the IP address that is used for the fa1 interface (of a switch) while in recovery mode.

To configure the recovery IP address for the fa1 interface, perform the following task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters configuration mode.
Step 2	Switch (config)# switch virtual domain <i>domain-id</i>	Specifies virtual switch domain.
Step 3	Switch (config-vs-domain)# [no] dual-active recovery [switch n] ip address <i>recovery-ip-address</i> <i>recovery-ip-mask</i>	Configures a recovery IP address. <i>n</i> is the VSS switch ID.

The following example shows how to set a recovery IP address 10.1.1.1:

```
Switch# configure terminal
Switch(config)# switch virtual domain 19
Switch(config-vs-domain)# dual-active recovery ip address 10.1.1.1 255.255.255.0
```

By default, **ip address** is not configured for recovery mode. So, the switch-fa1 interface is not associated with an IP address while the switch is in recovery mode. This ensures that two devices do not respond to the same IP address.

Without the **switch n** option, the (same) *recovery ip address* is used by either switch when it enters recovery mode. By definition, there is only one switch (in a given VSS system) in recovery mode at a time, making one recovery ip address sufficient.

If the two switches must use different IP addresses when the respective switch is in recovery mode, use the **switch n** option.

You can configure recovery IP addresses without the **switch n** option and with the **switch n** option simultaneously (for a total of three IP addresses, one global and one per switch). When done, the per-switch IP address takes precedence. If no per-switch IP address exists, the global IP address is used. Following are two examples:

Scenario 1

The VSS System is configured as follows:

- Global IP address- GIP
- switch 1 IP address - IP1
- switch 2 IP address - IP2

In this scenario, if switch 1 enters recovery mode, it will use IP1 for the fa1 interface on switch 1. Conversely, if switch 2 enters recovery mode, it will use IP2 for the fa1 interface on switch2.

Scenario 2

The VSS system is configured as follows:

- Global IP address - GIP
- switch 1 IP address - IP1
- switch 2 specific IP address

In this scenario, if switch 1 enters recovery mode, it will use IP1 for the fa1 interface on the switch 1. Conversely, if switch 2 enters recovery mode, it will use GIP for the fa1 interface on switch2.

VSS Initialization

A VSS is formed when the two switches and the VSL link between them become operational. The peer switch communicates over the VSL to negotiate the switches' roles.

If only one switch becomes operational, it assumes the VSS Active role. The VSS forms when the second switch becomes operational and both switches bring up their VSL interfaces.

VSS initialization is described in the following sections:

- [Virtual Switch Link Protocol, page 5-26](#)
- [SSO Dependencies, page 5-26](#)
- [Initialization Procedure, page 5-27](#)

Virtual Switch Link Protocol

The Virtual Switch Link Protocol (VSLP) consists of several protocols that contribute to virtual switch initialization. The VSLP includes the following protocols:

- Role Resolution Protocol

The peer switch use Role Resolution Protocol (RRP) to negotiate the role (VSS Active or VSS Standby) for each switch.

- Link Management Protocol

The Link Management Protocol (LMP) runs on all VSL links, and exchanges information required to establish communication between the two switches.

LMP identifies and rejects any unidirectional links. If LMP flags a unidirectional link, the switch that detects the condition brings the link down and up to restart the VSLP negotiation. VSL moves the control traffic to another port if necessary.

SSO Dependencies

For the VSS to operate with SSO redundancy, the VSS must meet the following conditions:

- Identical software versions (except during ISSU with compatible versions)
- VSL configuration consistency

During the startup sequence, the VSS Standby switch sends virtual switch information from the startup-config file to the VSS Active switch.

The VSS Active switch ensures that the following information matches correctly on both switches:

- Switch virtual domain
 - Switch virtual node
 - Switch priority (optional)
 - VSL port channel: switch virtual link identifier
 - VSL ports: channel-group number, shutdown, total number of VSL ports
- If the VSS detects a mismatch, it prints out an error message on the VSS Active switch console and the VSS Standby switch does not bootup. There are various ways to recover from this situation. If the switch is not running live traffic, you can either disconnect the VSL links or shutdown VSL ports on the peer, which would boot in VSS Active mode. You can make the necessary changes afterwards and reboot the switch and ensure VSL links are connected and not put in shutdown mode. Alternatively, you could clear the VSS rommon variable (VS_SWITCH_NUMBER) and allow the switch to boot in standalone mode. This method requires that no traffic flows through this switch. Once the switch is in standalone mode, you can convert it to VSS and then reboot it.
 - SSO and NSF enabled
- SSO and NSF must be configured and enabled on both switches. For detailed information on configuring and verifying SSO and NSF, see [Chapter 13, “Configuring Cisco NSF with SSO Supervisor Engine Redundancy.”](#)

If these conditions are unsatisfied, the VSS stops booting and ensures that the forwarding plane is not performing forwarding. For a description of SSO and RPR, see the [“VSS Redundancy” section on page 5-10](#).

Initialization Procedure

The following sections describe the VSS initialization procedure:

- [VSL Initialization, page 5-27](#)
- [System Initialization, page 5-28](#)
- [VSL Down, page 5-28](#)

VSL Initialization

A VSS is formed when the two switches and the VSL link between them become operational. Because both switches need to be assigned their role (VSS Active or VSS Standby) before completing initialization, VSL is brought online before the rest of the system is initialized. The initialization sequence is as follows:

1. The VSS initializes all cards with VSL ports, and then initializes the VSL ports.
2. The two switch communicate over VSL to negotiate their roles (VSS Active or VSS Standby).
3. The VSS Active switch completes the boot sequence, including the consistency check described in the [“SSO Dependencies” section on page 5-26](#).
4. If the consistency check completed successfully, the VSS Standby switch comes up in SSO VSS Standby mode. If the consistency check failed, the VSS Standby switch comes up in RPR mode.
5. The VSS Active switch synchronizes configuration and application data to the VSS Standby switch. If VSS is either forming for the first time or a mismatch exists between VSL information sent by the Standby switch and what is on the Active switch, the new configuration is absorbed in the

startup-config. This means that if the Active switch was running prior to the Standby switch and unsaved configurations existed, they would be written to the startup-config if the Standby switch sends mismatched VSL information.

System Initialization

If you boot both switches simultaneously, the switch configured as Switch 1 boots as VSS Active and the one with Switch 2 boots as VSS Standby. If priority is configured, the higher priority switch becomes active.

If you boot only one switch, the VSL ports remain inactive, and the switch boots as VSS Active. When you subsequently boot the other switch, the VSL links become active, and the new switch boots as VSS Standby. Because preemption is not supported, if a VSS Active is already running, the peer switch would always receive the VSS Standby role, even if its priority is higher than that of the Active's.

VSL Down

If the VSL is down when both switches try to boot up, the situation is similar to a dual-active scenario.

One of the switch becomes VSS Active and the other switch initiates recovery from the dual-active scenario. For further information, see the [“Configuring Dual-Active Detection”](#) section on page 5-52.

VSS Configuration Guidelines and Restrictions

The following sections describe restrictions and guidelines for VSS configuration:

- [General VSS Restrictions and Guidelines, page 5-28](#)
- [Multichassis EtherChannel Restrictions and Guidelines, page 5-29](#)
- [Dual-Active Detection Restrictions and Guidelines, page 5-30](#)

General VSS Restrictions and Guidelines

When configuring the VSS, note the following guidelines and restrictions:

- Beginning in Cisco IOS XE 3.8.0E, Quad-Supervisor VSS mode is supported on the Catalyst 4500 series switches.
- In Cisco IOS XE 3.4.0E (15.1(2)SG, E, VSS did not support SMI (both Director and Client).
Beginning with Cisco IOS XE 3.5.0E (15.2(1)E, VSS supports SmartInstall Director but not SMI Client.
Beginning with Cisco IOS XE 3.6.0E (15.2(2)E), VSS supports SmartInstall Director and SMI Client.
VSS [mode] is transparent to SMI except for the changes in interface names.
- The SMI Director has only one instance on VSS and runs on the VSS active switch. The standby Catalyst 4500 switch in a VSS is not listed as a director in the output of the **sh vstack status** command.
- The VSS configurations in the startup-config file must match on both switches; that is, the domain must match, the switch ID must be unique, and the VSL ports' information must match the physical connection.

- There is no restriction to configure oversubscribed linecard ports as VSL. The responsibility of bandwidth availability for a given network requirement lies with the network operator.
- VSL portchannel must have more than one port in the channel, preferably distributed on more than one module. If the VSL consists of only one link, its failure causes a Dual-Active operation of the VSS. Also, all VSL links configured on one module may cause a Dual-Active operation, if the module goes down..
- Classification and marking based on 'qos-group' in a QoS policy-map is not supported in VSS.
- The following older generation linecards (WS-X42xy to WS-X45xy) are supported with the VSS feature:
 - WS-X4148-RJ
 - WS-X4148-RJ
 - WS-X4148-FX-MT
 - WS-X4306-GB
 - WS-X4548-RJ45V+
 - WS-X4448-GB-SFP
 - WS-X4248-FE-SFP
 - WS-X4248-RJ45V

Please remove all other linecards from your system when converting from standalone to VSS mode.

- Do not attach a QoS policy with the maximum queue-limit (8184) to a large number of targets in a VSS system. This will cause continuous reloads on the standby supervisor engine.
- When an asymmetric virtual switch (i.e. a VSS comprising of chassis with different slot capacities) boots initially after conversion from standalone mode, the entPhysicalDescr object for the standby chassis does not hold the correct value. The entPhysicalDescr objects for both the active and standby chassis will match and hold the value for the active chassis.

After the running configuration is saved and a shelf reload occurs, this behaviour is not observed - the entPhysicalDescr objects for both chassis accurately reflects the correct chassis types.

Multichassis EtherChannel Restrictions and Guidelines

When configuring MECs, note the following guidelines and restrictions:

- Port Security over EtherChannels is not supported.
- All links in an MEC must terminate locally on the VSS Active or VSS Standby switch of the same virtual domain.
- An MEC can be connected to another MEC on a different VSS domain.
- Policers applied on an MEC are applied on two switches independently; if a policer is applied for 100 Mbps of conforming action, it will apply 100Mbps on both switches, resulting in a total conforming rate of 200 Mbps. To mitigate this, you can reduce the policer rate. In a more restrictive case, a rate of 50 Mbps might be necessary to achieve a maximum of 100Mbps. In a more liberal case, where conforming action of 200 Mbps is not a problem, policing rate could be kept to 100Mbps.

Dual-Active Detection Restrictions and Guidelines

When configuring dual-active detection, note the following guidelines and restrictions:

- For line redundancy, we recommend configuring at least two ports per switch for dual-active detection. For module redundancy, the two ports can be on different modules in each switch, and should be on different modules than the VSL ports, if feasible.
- Only trusted PAgP channels are relied upon to detect dual-active mode of operation.

Configuring a VSS

These sections describe how to configure a VSS:

- [Configuring Easy VSS, page 5-30](#)
- [Converting to a VSS, page 5-32](#)
- [Converting to Quad-Supervisor VSS, page 5-37](#)
- [Displaying VSS Information, page 5-39](#)
- [Converting a VSS to Standalone Switch, page 5-41](#)
- [Configuring VSS Parameters, page 5-42](#)
- [Configuring Multichassis EtherChannels, page 5-48](#)
- [Configuring Dual-Active Detection, page 5-52](#)

Configuring Easy VSS

Beginning with Cisco IOS XE 3.6.0E (IOS 15.2(2)E), the Catalyst 4500 series switch supports Easy VSS, which enables you to configure VSS with a single command on the active switch and no action on the VSS standby switch.

The active switch can gather information from all switches that are Layer 3 *reachable*.

**Note**

Quad-Supervisor VSS mode is not supported with Easy VSS.

**Note**

Both switches are directly connected to each other using Layer 3 physical interfaces and are reachable through these interfaces. These physical interfaces are candidate VSL interfaces and are displayed in a list of "potential" VSL interfaces in the output of the `vs1 ?` command in easy-vss mode. This output also displays a list of indirectly-reachable Layer 3 interfaces.

Cisco IOS XE 3.6.0E (IOS 15.2(2)E) only supports reachability using a default route. Management and user-created VRF are not supported.

**Note**

Switches are reachable to each other through management interfaces. Reachability to neighboring switches using a management interface isn't supported although the management interface appears in the candidate VSL list.

Switches can be Layer 3 reachable indirectly but directly connected. The directly-connected physical interfaces display in the output of the **vsl?** command, which displays all switches that have direct physical connections.

Alternatively, you can make a physical interface Layer 3 “capable” (i.e., make two switches reachable via directly connected Layer 3 links), by performing the following steps on both switches (A and B):

	Command	Purpose
Step 1	Switch(config)# interface <i>interface</i>	Selects interface and switches to interface configuration mode.
Step 2	Switch(config-if)# no switchport	Converts the switch to a Layer 3 interface.
Step 3	Switch(config-if)# ip add <i>a.a.a.a b.b.b.b</i>	Configures an IP address for temporary use.
Step 4	Switch(config-if)# exit	Exits interface configuration mode.

```
On Switch-A
Switch-A(config)# int G2/15
Switch-A(config-if)# no switchport
Switch-A(config-if)# ip address 5.5.5.6 255.255.255.0
```

```
On Switch-B
Switch-B(config)# int G3/15
Switch-B(config-if)# no switchport
Switch-B(config-if)# ip address 5.5.5.5 255.255.255.0
Ping 5.5.5.6 from switch-B
```

Issuing the **switch convert mode easy-virtual-switch** exec command on a VSS active switch displays a list of potential VSS standby switches - those that are directly connected and hardware compatible. From the displayed list, the sub-command **vsl ?** derives input from interfaces that belong to the switch where we are executing the command.

Perform the following task on the VSS active switch that you want to make the master switch, which manages the standby switch after VSS boot-up:

	Command	Purpose
Step 1	Switch# switch convert mode easy-virtual-switch	Switches to easy VSS sub-mode
Step 2	Switch(easy-vss)# VSL ? and Switch(easy-vss)# VSL local-interface	Displays a list of local inter-faces (with their peer interfaces, switch-ip and switch-name). Assigns the local interfaces that we want to convert to VSL. Choose interfaces under the column Local Interfaces under 'VSL?'
Step 3	Switch(easy-vss)# exit	Return to exec command mode.

The following example illustrates use of the **vsl ?** command:

```
SwitchA# switch convert mode easy-virtual-switch
# (easy-vss)# VLS ?
Local Interface      Remote Interface      Hostname      Standby-IP
GigabitEthernet2/15  GigabitEthernet3/15   Switch-B      5.5.5.5
GigabitEthernet2/17  GigabitEthernet3/17   Switch-B      5.5.5.5
GigabitEthernet2/4   GigabitEthernet3/4    Switch-C      4.4.4.4
```

The switch on which we execute the above commands becomes the master switch after VSS boots. Local Interfaces lists interfaces on the switch where we are executing the commands. Remote Interfaces lists the interfaces on the peer switch connected with the local interfaces.

Select a maximum of eight VSL local interfaces (i.e., interfaces under the Local Interface column).

This example forces both the master and standby switches to reboot and come up in VSS. Now, we have two interfaces as VSL members with the local interfaces GigabitEthernet 2/15 and GigabitEthernet 2/17.

```
SwitchA# switch convert mode easy-virtual-switch
SwitchA(easy-vss)# VSL GigabitEthernet2/15 GigabitEthernet2/17
```



Note 10G and 1G interfaces cannot be mixed. Chosen interfaces should belong to the same peer.

The master switch shares the tftp image path with the standby switch. On reboot, if the tftp path is used for loading the image, both switches boot with the same image.

Converting to a VSS

By default, the Catalyst 4500 series switch is configured to operate in standalone mode (the switch works independently). The VSS combines two standalone switches into one virtual switch, operating in virtual switch mode.



Note When you convert two standalone switches into one VSS, all non-VSL configuration settings on the VSS Standby switch will revert to the default configuration.



Note Preferably, conversion to VSS should be done on a maintenance window. If you plan to use the same port channel number for VSL, default the existing port channel configurations that are available on standalone switches. Then, follow the guidelines in section [Configuring VSL Port Channel and Ports, page 5-34](#).

To convert two standalone switches into a VSS, you perform the following major activities:

- Save the standalone configuration files.
- Configure each switch for required VSS configurations.
- Convert to a VSS.

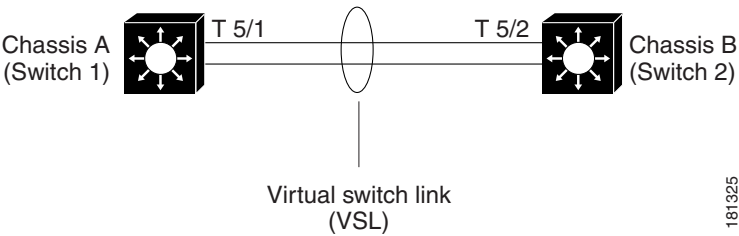
In virtual switch mode, both switches use the same configuration file. When you make configuration changes on the VSS Active switch, these changes are automatically propagated to the VSS Standby switch.

The tasks required to convert the standalone switch to a VSS are detailed in the following sections:

- [Backing Up the Standalone Configuration, page 5-33](#)
- [Configuring SSO and NSF, page 5-34](#)
- [Assigning Virtual Switch Domain and Switch Numbers, page 5-34](#)
- [Configuring VSL Port Channel and Ports, page 5-34](#)
- [Converting the Switch to Virtual Switch Mode, page 5-36](#)
- [\(Optional\) Configuring VSS Standby Switch Modules, page 5-39](#)

In the procedures that follow, the example commands assume the configuration shown in [Figure 5-8](#).

Figure 5-8 Example VSS



Two chassis, A and B, are converted into a VSS with virtual switch domain 100. Interface 10-Gigabit Ethernet 5/1 on Switch 1 is connected to interface 10-Gigabit Ethernet 5/2 on Switch 2 to form the VSL.



Note

The port channels 10 and 20 mentioned in the config steps below are merely exemplary. You can configure any port channel number from 1-64 for VSL port channel.

Backing Up the Standalone Configuration

Save the configuration files for both switches operating in standalone mode. You need these files to revert to standalone mode from virtual switch mode.

Perform this task on both switches:

	Command	Purpose
Step 1	Switch-1# copy running-config startup-config	(Optional) Saves the running configuration to startup configuration.
Step 2	Switch-1# copy startup-config bootflash:old-startup-config Or Switch-1# copy startup-config slot0:old-startup-config	Copies the startup configuration to a backup file.
Step 3	Switch-1# copy startup-config slavebootflash:old-startup-config Or Switch-1# copy startup-config slaveslot0:old-startup-config	Note This step is required for switches in Quad-Supervisor VSS mode only. Copies the startup configuration to the standby supervisors.

Configuring SSO and NSF

SSO and NSF are configured as default on VSS.

Assigning Virtual Switch Domain and Switch Numbers

You must configure the same virtual switch domain number on both switches of the VSS. The virtual switch domain is a number between 1 and 255, and must be unique for each VSS in your network (the domain number is incorporated into various identifiers to ensure that these identifiers are unique across the network).


Within the VSS, you must configure one switch to be switch number 1 and the other switch to be switch number 2.

To configure the virtual switch domain and switch number on both switches, perform this task on Switch 1:

	Command	Purpose
Step 1	Switch-1(config)# switch virtual domain 100	Configures the virtual switch domain on Switch A.
Step 2	Switch-1(config-vs-domain)# switch 1	Configures Switch A as virtual switch number 1.
Step 3	Switch-1(config-vs-domain)# exit	Exits config-vs-domain.

Perform the following task on Switch 2:

	Command	Purpose
Step 1	Switch-2(config)# switch virtual domain 100	Configures the virtual switch domain on Switch B.
Step 2	Switch-2(config-vs-domain)# switch 2	Configures Switch B as virtual switch number 2.
Step 3	Switch-2(config-vs-domain)# exit	Exits config-vs-domain.



Note

The switch number is not stored in the startup or running configuration, because both switches use the same configuration file (but must not have the same switch number).

Configuring VSL Port Channel and Ports

The VSL is configured with a unique port channel on each switch. During the conversion, the VSS configures both port channels on the VSS Active switch. If the VSS Standby switch VSL port channel number has been configured for another use, the VSS comes up in RPR mode. To avoid this situation, check that both port channel numbers are available on both of the switches.

Check the port channel number with the **show running-config interface port-channel** command. The command displays an error message if the port channel is available for VSL. For example, the following command shows that port channel 20 is available on Switch 1:

```
Switch-1 # show running-config interface port-channel 20
% Invalid input detected at '^' marker.
```

To configure the VSL port channels, perform this task on Switch 1:

**Note**

The port channels 10 and 20 mentioned in the configuration steps below are exemplary only. You can configure any port channel number from 1-64 for VSL port channel.

	Command	Purpose
Step 1	Switch-1(config)# interface port-channel 10	Configures port channel 10 on Switch 1.
Step 2	Switch-1(config)# switchport	Convert to a Layer 2 port.
Step 3	Switch-1(config-if)# switch virtual link 1	Associates Switch 1 as owner of port channel 10.
Step 4	Switch-1(config-if)# no shutdown	Activates the port channel.
Step 5	Switch-1(config-if)# exit	Exits interface configuration.

Perform the following task on Switch 2:

	Command	Purpose
Step 1	Switch-2(config)# interface port-channel 20	Configures port channel 20 on Switch 2.
Step 2	Switch-2(config)# switchport	Convert to a Layer 2 port.
Step 3	Switch-2(config-if)# switch virtual link 2	Associates Switch 2 as owner of port channel 20.
Step 4	Switch-2(config-if)# no shutdown	Activates the port channel.
Step 5	Switch-2(config-if)# exit	Exits interface configuration mode.

You must add the VSL physical ports to the port channel. In the following example, interfaces 10-Gigabit Ethernet 3/1 and 3/2 on Switch 1 are connected to interfaces 10-Gigabit Ethernet 5/2 and 5/3 on Switch 2.

**Tip**

For line redundancy, we recommend configuring at least two ports per switch for the VSL. For module redundancy, the two ports can be on different switching modules in each chassis.

To configure the VSL ports, perform this task on Switch 1:

	Command	Purpose
Step 1	Switch-1(config)# interface range tengigabitethernet 3/1-2	Enters configuration mode for interface range tengigabitethernet 3/1-2 on Switch 1.
Step 2	Switch-1(config-if)# channel-group 10 mode on	Adds this interface to channel group 10.

**Note**

1G ports, which are converted from 10G ports using a connector, are not supported for VSL. This impacts Sup7-E and Sup7L-E ports.

On Switch 2, perform this task:

	Command	Purpose
Step 1	Switch-2(config)# interface range tengigabitethernet 5/2-3	Enters configuration mode for interface range tengigabitethernet 5/2-3 on Switch 2.
Step 2	Switch-2(config-if)# channel-group 20 mode on	Adds this interface to channel group 20.

**Note**

1G ports, which are converted from 10G ports using a connector, are not supported for VSL. This impacts Sup7-E and Sup7L-E ports.

Converting the Switch to Virtual Switch Mode

Conversion to virtual switch mode requires a restart for both switches. After the reboot, commands that specify interfaces with module/port now include the switch number. For example, a port on a switching module is specified by switch/module/port.

Prior to the restart, the VSS converts the startup configuration to use the switch/module/port convention. A backup copy of the startup configuration file is saved in bootflash. This file is assigned a default name, but you are also prompted to override the default name if you want to change it.

To convert Switch 1 to virtual switch mode, perform this task:

Command	Purpose
Switch-1# switch convert mode virtual	<p>Converts Switch 1 to virtual switch mode.</p> <p>After you enter the command, you are prompted to confirm the action. Enter yes.</p> <p>The system creates a converted configuration file, and saves the file to the bootflash of the VSS active supervisor. In Quad-Supervisor VSS mode, the configuration files must be copied from the active supervisor to the in-chassis standby supervisor, using the copy bootflash: <image_name> slavebootflash-ics: <image_name> command.</p>

To convert Switch 2 to virtual switch mode, perform this task on Switch 2:

Command	Purpose
Switch-2# switch convert mode virtual	<p>Converts Switch 2 to virtual switch mode.</p> <p>After you enter the command, you are prompted to confirm the action. Enter yes.</p> <p>The system creates a converted configuration file, and saves the file to the bootflash.</p>

**Note**

After you confirm the command (by entering **yes** at the prompt), the running configuration is automatically saved as the startup configuration and the switch reboots. After the reboot, the switch is in virtual switch mode, so you must specify interfaces with three identifiers (switch/module/port).

When switches are being converted to VSS, you should not set them to ignore startup-config. If done, the switch can be enabled to parse the startup-config at the rommon prompt. Ignoring startup-config in VSS mode, causes a switch to boot in a semi-VSS mode, which can only be corrected by a reboot and by enabling the parsing of startup-config.

Converting to Quad-Supervisor VSS

You can convert to Quad-Supervisor VSS mode using one of the following ways:

- [Converting an Existing VSS Switch to Quad-Supervisor VSS Mode](#)
- [Converting a Standalone Switch to Quad-Supervisor VSS, page 5-37](#)

Converting an Existing VSS Switch to Quad-Supervisor VSS Mode

To convert to Quad-Supervisor VSS from an existing VSS setup, perform the following task:

- | | |
|---------------|---|
| Step 1 | Perform an ISSU or a switch reload, to upgrade the chassis active supervisors with the Cisco IOS XE image that supports Quad-Supervisor VSS mode. Ensure that the switch reaches Stateful Switchover (SSO) in the VSS. The BOOT variable must point to the path of the Cisco IOS XE image and must be saved in the startup configuration. For more information about ISSU upgrade, see In-Service Software Upgrade (ISSU) on a VSS, page 5-56 . |
| Step 2 | Insert the redundant supervisors in the appropriate slots. Configure ROM monitor to auto-boot on the standby supervisors of both the VSS Active and the VSS Standby switches, using config register. Load the Cisco IOS XE image on both the standby supervisors and ensure that both supervisors are brought up in RPR mode as in-chassis standby (ICS) supervisors. The in-chassis active supervisor then syncs its startup configuration to the ICS. |
| Step 3 | Copy the Cisco IOS XE image on the standby supervisors of both the switches, from the VSS Active switch console, using the IOS copy command so that the boot variable has a valid path to point to |

Command	Purpose
Switch# copy bootflash: <image_name> bootflash-ics: <image_name>	Copies the image from the VSS Active switch to the ICS bootflash system of the VSS Active.
Switch# copy bootflash: <image_name> slavebootflash-ics: <image_name>	Copies the image from the VSS Active switch to the ICS bootflash system of the in-chassis standby supervisor.

The ICS bootflash filesystem is mounted with the name `bootflash-ics:`. On the VSS Active switch, the ICS bootflash filesystem of the VSS standby switch is available as `slavebootflash-ics:`.

Converting a Standalone Switch to Quad-Supervisor VSS

To convert a standalone switch to Quad-Supervisor VSS mode, perform the following task:

- Step 1** Copy the Cisco IOS XE image that supports Quad-Supervisor VSS mode on the bootflash: on the active and standby supervisors on both switches.
- Step 2** Follow the steps in [Converting to a VSS, page 5-32](#).

Example of Show Module Output for Switches in Quad- Supervisor VSS Mode

The **show module** command should display similar output in Quad-Supervisor VSS mode:

Switch# **show module**

Switch Number: 1 Role: Virtual Switch Active

Chassis Type : WS-C4507R+E

Power consumed by backplane : 40 Watts

Mod	Ports	Card Type	Model	Serial No.
1	48	10/100/1000BaseT Premium POE E Series	WS-X4748-RJ45V+E	CAT1737L6TE
2	24	1000BaseX SFP	WS-X4724-SFP-E	CAT1742L0TX
3	8	Sup 8-E 10GE (SFP+), 1000BaseX (SFP)	WS-X45-SUP8-E	CAT1825L0GH
4	8	Sup 8-E 10GE (SFP+), 1000BaseX (SFP)		

M	MAC addresses	Hw	Fw	Sw	Status
1	885a.9244.d734 to 885a.9244.d763	1.3			Ok
2	4c00.821a.6dc0 to 4c00.821a.6dd7	0.3			Ok
3	c067.af69.c400 to c067.af69.c407	1.1	15.1(1r)SG5	03.08.00.E.	Ok
4	c067.af69.c408 to c067.af69.c40f				Provision

Mod	Redundancy role	Operating mode	Redundancy status
3	Active Supervisor	SSO	Active
4	ICS Supervisor	RPR	Standby Cold

Switch Number: 2 Role: Virtual Switch Standby

Chassis Type : WS-C4507R+E

Power consumed by backplane : 40 Watts

Mod	Ports	Card Type	Model	Serial No.
1	48	1000BaseX SFP	WS-X4748-SFP-E	CAT1745L1NZ
3	8	Sup 8-E 10GE (SFP+), 1000BaseX (SFP)	WS-X45-SUP8-E	CAT1736L7A4
4	8	Sup 8-E 10GE (SFP+), 1000BaseX (SFP)		
5	48	10/100/1000BaseT EEE (RJ45)	WS-X4748-RJ45-E	CAT1740L08E
6	48	1000BaseX SFP	WS-X4448-GB-SFP	JAE1133TZ26

M	MAC addresses	Hw	Fw	Sw	Status
1	e4c7.22f5.9a9c to e4c7.22f5.9acb	0.4			Ok
3	c067.afe3.0600 to c067.afe3.0607	1.0	15.1(1r)SG5	03.08.00.E.	Ok
4	c067.afe3.0608 to c067.afe3.060f				Provision
5	885a.92e1.e100 to 885a.92e1.e12f	1.1			Ok
6	001d.4510.9b30 to 001d.4510.9b5f	1.3			Ok

Mod	Redundancy role	Operating mode	Redundancy status
3	Standby Supervisor	SSO	Standby hot
4	ICS Supervisor	RPR	Standby cold

(Optional) Configuring VSS Standby Switch Modules



Note

You cannot configure or provision modules on VSS.

When switches form initial VSS relationships, they send module information to each other and this information is pushed to the configuration and used subsequently for provisioning, provided the switch is booting and the peer is down or not present.

The following example shows the module provisioning information:

```
module provision switch 1
  slot 1 slot-type 148 port-type 60 number 4   virtual-slot 17
  slot 2 slot-type 137 port-type 31 number 16  virtual-slot 18
  slot 3 slot-type 227 port-type 60 number 8   virtual-slot 19
  slot 4 slot-type 225 port-type 61 number 48  virtual-slot 20
  slot 5 slot-type 82  port-type 31 number 2   virtual-slot 21
module provision switch 2
  slot 1 slot-type 148 port-type 60 number 4   virtual-slot 33
  slot 2 slot-type 227 port-type 60 number 8   virtual-slot 34
  slot 3 slot-type 137 port-type 31 number 16  virtual-slot 35
  slot 4 slot-type 225 port-type 61 number 48  virtual-slot 36
  slot 5 slot-type 82  port-type 31 number 2   virtual-slot 37
```

These commands are not available to the user and that various numbers used in these commands are internal to the system and used to identify a module. These commands are written to the startup-config when a switch detects a given module while it is running in VSS mode. When reconverted to standalone mode, these commands are removed from the startup-config.

Displaying VSS Information

To display basic information about the VSS, perform one of these tasks:

Command	Purpose
Switch# show switch virtual	Displays the virtual switch domain number, and the switch number and role for each of the switches.
Switch# show switch virtual role	Displays the role, switch number, and priority for each of the switch in the VSS.
Switch# show switch virtual link	Displays the status of the VSL.

The following example shows the information output from these commands:

```
Switch# show switch virtual
Executing the command on VSS member switch role = VSS Active, id = 1

Switch mode                : Virtual Switch
Virtual switch domain number : 100
Local switch number        : 1
Local switch operational role: Virtual Switch Active
Peer switch number         : 2
Peer switch operational role : Virtual Switch Standby

Executing the command on VSS member switch role = VSS Standby, id = 2

Switch mode                : Virtual Switch
```

```

Virtual switch domain number : 100
Local switch number          : 2
Local switch operational role: Virtual Switch Standby
Peer switch number           : 1
Peer switch operational role : Virtual Switch Active

```

Switch# **show switch virtual role**

Executing the command on VSS member switch role = VSS Active, id = 1

RRP information for Instance 1

```

-----
Valid  Flags   Peer      Preferred  Reserved
      Count      Peer
-----
TRUE   V       1         1          1

Switch  Switch Status  Preempt      Priority  Role      Local  Remote
      Number      Oper (Conf)  Oper (Conf)  Oper (Conf)  SID    SID
-----
LOCAL   1       UP    FALSE(N )    100(100)  ACTIVE   0      0
REMOTE  2       UP    FALSE(N )    100(100)  STANDBY  7496   7678

```

Peer 0 represents the local switch

Flags : V - Valid
In dual-active recovery mode: No

Executing the command on VSS member switch role = VSS Standby, id = 2

RRP information for Instance 2

```

-----
Valid  Flags   Peer      Preferred  Reserved
      Count      Peer
-----
TRUE   V       1         1          1

Switch  Switch Status  Preempt      Priority  Role      Local  Remote
      Number      Oper (Conf)  Oper (Conf)  Oper (Conf)  SID    SID
-----
LOCAL   2       UP    FALSE(N )    100(100)  STANDBY   0      0
REMOTE  1       UP    FALSE(N )    100(100)  ACTIVE   7678   7496

```

Peer 0 represents the local switch

Flags : V - Valid
In dual-active recovery mode: No

Switch# **show switch virtual link**

Executing the command on VSS member switch role = VSS Active, id = 1

```

VSL Status : UP
VSL Uptime : 13 minutes
VSL Control Link : Tel1/1/1

```

Executing the command on VSS member switch role = VSS Standby, id = 2

```
VSL Status : UP
VSL Uptime : 13 minutes
VSL Control Link : Te2/1/1
```

Converting a VSS to Standalone Switch

To convert a VSS into two standalone systems, you perform the following major steps:

- [Copying the VSS Configuration to a Backup File, page 5-41](#)
- [Converting the VSS Active Switch to Standalone, page 5-41](#)
- [Converting the VSS Standby Switch to Standalone, page 5-42](#)

Copying the VSS Configuration to a Backup File

Save the configuration file from the VSS Active switch. You may need this file if you convert to virtual switch mode again. You only need to save the file from the VSS Active switch, because the configuration file on the VSS Standby switch is identical to the file on the VSS Active switch.

	Command	Purpose
Step 1	Switch-1# copy running-config startup-config	(Optional) Saves the running configuration to startup configuration. This step is only required if there are unsaved changes in the running configuration that you want to preserve.
Step 2	Switch-1# copy startup-config bootflash:vs-startup-config	Copies the startup configuration to a backup file.

Converting the VSS Active Switch to Standalone

When you convert the VSS Active switch to standalone mode, the VSS Active switch removes the provisioning and configuration information related to VSL links and the peer chassis modules, saves the configuration file, and performs a reload. The switch comes up in standalone mode with only the configuration data relevant to the standalone system.

The VSS Standby switch of the VSS becomes VSS Active. VSL links on this switch are down because the peer is now unavailable.

To convert the VSS Active switch to standalone mode, perform this task on the VSS Active switch:

Command	Purpose
Switch-1# switch convert mode stand-alone	Converts Switch 1 to standalone mode. After you enter the command, you are prompted to confirm the action. Enter yes .

Conversion from VSS to standalone causes all physical interfaces to be administratively shutdown and written to the startup-config. This is a safeguard against a standalone system arriving in the network alive and conflicting with a bridge or router MAC address, which might still be there if one of the VSS switches is still running in VSS mode.

We do not recommend that you convert a VSS to standalone in a live network.

Converting the VSS Standby Switch to Standalone

When you convert the new VSS Active switch to standalone mode, the switch removes the provisioning and configuration information related to VSL links and the peer switch modules, saves the configuration file and performs a reload. The switch comes up in standalone mode with only its own provisioning and configuration data.

To convert the peer switch to standalone, perform this task on the VSS Standby switch:

Command	Purpose
Switch-2# switch convert mode stand-alone	Converts Switch 2 to standalone mode. After you enter the command, you are prompted to confirm the action. Enter yes .

Configuring VSS Parameters

These sections describe how to configure VSS parameters:

- [Configuring VSL Switch Priority, page 5-43](#)
- [Configuring a VSL, page 5-44](#)
- [Adding and Deleting a VSL Port After the Bootup, page 5-44](#)
- [Displaying VSL Information, page 5-45](#)
- [Configuring VSL QoS, page 5-46](#)
- [Configuring the Router MAC Address, page 5-47](#)

Configuring VSL Switch Priority

To configure the switch priority, perform this task:

	Command	Purpose
Step 1	Switch(config)# switch virtual domain 100	Enters configuration mode for the virtual switch domain.
Step 2	Switch(config-vs-domain)# switch [1 2] priority [priority_num]	<p>Configures the priority for the switch. The switch with the higher priority assumes the VSS Active role. The range is 1 (lowest priority) to 255 (highest priority); the default is 100.</p> <p>Note</p> <ul style="list-style-type: none"> The new priority value only takes effect after you save the configuration and perform a reload of the VSS. If the higher priority switch is currently in VSS Standby state, you can make it the VSS Active switch by initiating a switchover with the redundancy force-switchover command. <p>The show switch virtual role command displays the operating priority and the configured priority for each switch in the VSS.</p> <ul style="list-style-type: none"> The no form of the command resets the priority value to the default value of 100. The new value takes effect after you save the configuration and perform a reload.
Step 3	Switch# show switch virtual role	Displays the current priority.



Note

If you make configuration changes to the switch priority, the changes only take effect after you save the running configuration to the startup configuration file and perform a reload. The **show switch virtual role** command shows the operating and configured priority values. You can manually set the VSS Standby switch to VSS Active using the **redundancy force-switchover** command.

This example shows how to configure virtual switch priority:

```
Switch(config)# switch virtual domain 100
Switch(config-vs-domain)# switch 1 priority 200
Switch(config-vs-domain)# exit
```

This example shows how to display priority information for the VSS:

```
Switch# show switch virtual role
Switch  Switch Status  Preempt   Priority  Role      Session ID
        Number         Oper (Conf) Oper (Conf) Local Remote
-----
LOCAL   1      UP        FALSE(N)  100(200)  ACTIVE    0        0
REMOTE  2      UP        FALSE(N)  100(100)  STANDBY   8158     1991
```

In dual-active recovery mode: No

Configuring a VSL

To configure a port channel to be a VSL, perform this task:

	Command	Purpose
Step 1	Switch(config)# interface port-channel <i>channel_num</i>	Enters configuration mode for the specified port channel.
Step 2	Switch(config-if)# switch virtual link <i>switch_num</i>	Assigns the port channel to the virtual link for the specified switch.



Note

We recommend that you configure the VSL prior to converting the switch into a VSS.

This example shows how to configure the VSL:

```
Switch-1(config)# interface port-channel 10
Switch-1(config-if)# switch virtual link 1
Switch-1(config-if)# no shutdown (If the port is admin shutdown)
Switch-1(config)# interface tenGigabitEthernet 5/1
Switch-1(config-if)# channel-group 10 mode on
Switch-1(config-if)# no shutdown (If the port is admin shutdown)

Switch-2(config)# interface port-channel 25
Switch-2(config-if)# switch virtual link 2
Switch-2(config-if)# no shutdown (If the port is admin shutdown)
Switch-2(config-if)# interface tenGigabitEthernet 5/2
Switch-2(config-if)# channel-group 25 mode on
Switch-2(config-if)# no shutdown (If the port is admin shutdown)
```

Adding and Deleting a VSL Port After the Bootup

At any time, you can add and delete VSL ports from a port-channel to increase the number of links in the VSL, to move the port from one port to another, or to remove it from VSL.

Before adding or deleting VSL ports, do the following:

- Ensure all ports are physically connected to the peer switch. The peer port must also be configured for VSL.
- Shutdown the port before configuring VSL. When both ports on the link are configured for VSL, **unshut** them.
- Spread VSL ports across multiple modules.
- While deleting a port, retain at least one “active” VSL port pair. Else, a dual-active operation could occur.
- To save link flap and high CPU, shutdown the ports before VSL is unconfigured.
- After adding, deleting, or modifying VSL ports, write the config to nvram (that is, startup-config).
- If you need to move ports to another port, account for the bandwidth requirement of VSL. You should add an additional VSL link in the channel, move ports and remove additional links in the channel.

Displaying VSL Information

To display information about the VSL, perform one of these tasks:

Command	Purpose
Switch# show switch virtual link	Displays information about the VSL.
Switch# show switch virtual link port-channel	Displays information about the VSL port channel.
Switch# show switch virtual link port	Displays information about the VSL ports.

This example shows how to display VSL information:

```
Switch# show switch virtual link
VSL Status : UP
VSL Uptime : 1 day, 3 hours, 39 minutes
VSL Control Link : Te 1/5/1

Switch# show switch virtual link port-channel

Flags:  D - down          P - bundled in port-channel
        I - stand-alone  s - suspended
        H - Hot-standby  (LACP only)
        R - Layer3       S - Layer2
        U - in use       N - not in use, no aggregation
        f - failed to allocate aggregator

        M - not in use, no aggregation due to minimum links not met
        m - not in use, port not aggregated due to minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated

Group  Port-channel  Protocol  Ports
-----+-----+-----+-----
10     Po10(RU)       -         Te1/5/4(P) Te1/5/5(P)
20     Po20(RU)       -         Te2/5/4(P) Te2/5/5(P)

Switch# show switch virtual link port
LMP summary

Link info:          Configured: 1          Operational: 1

Interface Flag State      Peer Peer      Peer  Peer      Timer(s)running
Flag MAC              Switch Interface (Time remaining)
-----
Gi1/3/11  vfsp operational  vfsp f866.f296.be00 2      Gi2/1/11  T4(708ms)
                                                T5(29.91s)

Flags:  v - Valid flag set          f - Bi-directional flag set
        s - Negotiation flag set    p - Peer detected flag set

Timers: T4 - Hello Tx Timer    T5 - Hello Rx Timer

LMP Status

Last operational      Current packet      Last Diag      Time since
Interface Failure state      State              Result          Last Diag
-----
Gi1/3/11  No failure          Hello bidir          Never ran      --
```

```

LMP hello timer

Interface      State      Hello Tx (T4) ms      Hello Rx (T5*) ms
      Cfg      Cur      Rem      Cfg      Cur      Rem
-----
Gi1/3/11      operational -          1000      708      -          30000      29144

*T5 = min_rx * multiplier
Cfg : Configured Time
Cur : Current Time
Rem : Remaining Time

```

Configuring VSL QoS

When a physical port is configured as a member of a VSL port-channel, a queuing policy is automatically attached to the VSL member ports. This queuing policy provides a dedicated queue for VSS Management, VSLP, BFD, Layer 2 and Layer 3 control protocols, and voice and video data traffic. Each queue is provided with a minimum bandwidth, ensuring that VSS management and control protocol packets are not dropped when congestion occurs on the VSL. The bandwidth assigned to a class of traffic is the minimum bandwidth that is guaranteed to the class during congestion. The VSL link uses Transmit Queue Sharing, where the output link bandwidth is shared among multiple queues of a given VSL port. Any modification or removal of VSL Queuing policy is restricted in a VSS system.

The following command sequence is inserted automatically by software.

```

interface TenGigabitEthernet1/1/1
  switchport mode trunk
  switchport nonegotiate
  no lldp transmit
  no lldp receive
  no cdp enable
  channel-group 10 mode on
  service-policy output VSL-Queuing-Policy
end

Switch# show policy-map VSL-Queuing-Policy
  Policy Map VSL-Queuing-Policy
    Class VSL-MGMT-PACKETS
      bandwidth percent 5
    Class VSL-L2-CONTROL-PACKETS
      bandwidth percent 5
    Class VSL-L3-CONTROL-PACKETS
      bandwidth percent 5
    Class VSL-VOICE-VIDEO-TRAFFIC
      bandwidth percent 30
    Class VSL-SIGNALING-NETWORK-MGMT
      bandwidth percent 10
    Class VSL-MULTIMEDIA-TRAFFIC
      bandwidth percent 20
    Class VSL-DATA-PACKETS
      bandwidth percent 20
    Class class-default
      bandwidth percent 5

class-map match-any VSL-MGMT-PACKETS
  match access-group name VSL-MGMT

class-map match-any VSL-DATA-PACKETS
  match any

class-map match-any VSL-L2-CONTROL-PACKETS

```



```

match access-group name VSL-DOT1x
match access-group name VSL-BPDU
match access-group name VSL-CDP
match access-group name VSL-LLDP
match access-group name VSL-SSTP
match access-group name VSL-GARP

class-map match-any VSL-L3-CONTROL-PACKETS
  match access-group name VSL-IPV4-ROUTING
  match access-group name VSL-BFD
  match access-group name VSL-DHCP-CLIENT-TO-SERVER
  match access-group name VSL-DHCP-SERVER-TO-CLIENT
  match access-group name VSL-DHCP-SERVER-TO-SERVER
  match access-group name VSL-IPV6-ROUTING

class-map match-any VSL-MULTIMEDIA-TRAFFIC
  match dscp af41
  match dscp af42
  match dscp af43
  match dscp af31
  match dscp af32
  match dscp af33
  match dscp af21
  match dscp af22
  match dscp af23

class-map match-any VSL-VOICE-VIDEO-TRAFFIC
  match dscp ef
  match dscp cs4
  match dscp cs5

class-map match-any VSL-SIGNALING-NETWORK-MGMT
  match dscp cs2
  match dscp cs3
  match dscp cs6
  match dscp cs7

```

Configuring the Router MAC Address

On VSS, all routing protocols are centralized on the active supervisor engine. A common router MAC address is used for Layer 3 interfaces on both active and standby switches. Additionally, to ensure non-stop forwarding, the same router MAC address is used after switchover to Standby, so that all layer 3 peers see a consistent router MAC address.

There are three ways to configure a router MAC address on VSS:

- **HHH**—Manually set a router MAC address. Ensure that this MAC address is reserved for this usage.
- **chassis**—Use the mac-address range reserved for Chassis. This is the Cisco MAC address assigned to the chassis.
- **use-virtual**—Use the mac-address range reserved for the VSS. This is the served Cisco MAC address pool, which is derived from a base MAC address +vvs domain-id.

By default, the virtual domain based router MAC address is used. Any change of router MAC address configuration requires a reboot of both VSS supervisor engines

The follow table shows how to configure the router MAC address.

Command	Purpose
Switch(config)# switch virtual domain <i>domain_id</i>	Enters VSS configuration mode.
Switch(config-vs-domain)# mac-address use-virtual	Assigns the router MAC address from a reserved pool of domain-based addresses. Note This is the default. This is shown in the configuration, even if it the default.
Switch(config-vs-domain)# mac-address <i>mac-address</i>	Assigns the router MAC address in three 2-byte hexadecimal numbers.
Switch(config-vs-domain)# mac-address chassis	Specifies the router MAC address as the last address of chassis MAC address range.

Configuring Multichassis EtherChannels

Configure multichassis EtherChannels (MECs) as you would for a regular EtherChannel. The VSS will recognize that the EtherChannel is an MEC when ports from both switches are added to the EtherChannel. You can verify the MEC configuration by entering the **show etherchannel** command.

One VSS supports a maximum of 256 port channels.

To configure Layer 3 Multichassis EtherChannels, create the port channel logical interface and then put the Ethernet interfaces from both the VSS active and VSS standby into the port channel.

These sections describe Layer 3 EtherChannel configuration:

- [Creating Port Channel Logical Interfaces, page 5-48](#)
- [Configuring Physical Interfaces as Layer 3 EtherChannels, page 5-49](#)

Creating Port Channel Logical Interfaces



Note

To move an IP address from a physical interface to an EtherChannel, you must delete the IP address from the physical interface before configuring it on the port channel interface.

To create a port channel interface for a Layer 3 EtherChannel, perform this task:

	Command	Purpose
Step 1	Switch(config)# interface port-channel <i>port_channel_number</i>	Creates the port channel interface. The value for <i>port_channel_number</i> can range from 1 to 64.
Step 2	Switch(config-if)# ip address <i>ip_address mask</i>	Assigns an IP address and subnet mask to the EtherChannel.
Step 3	Switch(config-if)# end	Exits configuration mode.
Step 4	Switch# show running-config interface port-channel <i>port_channel_number</i>	Verifies the configuration.

This example shows how to create port channel interface 1:

```
Switch# configure terminal
Switch(config)# interface port-channel 1
```

```
Switch(config-if)# ip address 172.32.52.10 255.255.255.0
Switch(config-if)# end
```

This example shows how to verify the configuration of port channel interface 1:

```
Switch# show running-config interface port-channel 1
Building configuration...
```

```
Current configuration:
!
interface Port-channel1
 ip address 172.32.52.10 255.255.255.0
end
```

```
Switch#
```

Configuring Physical Interfaces as Layer 3 EtherChannels

To configure physical interfaces as Layer 3 EtherChannels, perform this task for each interface:

	Command	Purpose
Step 1	Switch(config)# interface { fastethernet gigabitethernet tengigabitethernet } <i>slot/port</i>	Selects a physical interface to configure.
Step 2	Switch(config-if)# no switchport	Makes this a Layer 3 routed port.
Step 3	Switch(config-if)# no ip address	Ensures that no IP address is assigned to the physical interface.
Step 4	Switch(config-if)# channel-group <i>port_channel_number</i> mode { active on auto passive desirable }	Configures the interface in a port channel and specifies the PAgP or LACP mode. If you use PAgP, enter the keywords auto or desirable . If you use LACP, enter the keywords active or passive .
Step 5	Switch(config-if)# end	Exits configuration mode.
Step 6	Switch# show running-config interface port-channel <i>port_channel_number</i> Switch# show running-config interface { fastethernet gigabitethernet tengigabitethernet } <i>slot/port</i> Switch# show interfaces { fastethernet gigabitethernet tengigabitethernet } <i>slot/port</i> etherchannel Switch# show etherchannel 1 port-channel	Verifies the configuration.

This example shows how to configure Gigabit Ethernet interfaces 1/3/26 and 2/2/26 into port channel 1 with PAgP mode **desirable**:

```
Switch(config)# conf terminal
Switch(config)# int gigabitEthernet 1/3/26
Switch(config-if)# no switchport
Switch(config-if)# no ip address
Switch(config-if)# channel-group 1 mode desirable
Switch(config-if)# exit
Switch(config)# int gigabitEthernet 2/2/6
Switch(config-if)# no switchport
```

```
Switch(config-if)# no ip address
Switch(config-if)# channel-group 1 mode desirable
Switch(config-if)# end
```

**Note**

See the “[Configuring a Range of Interfaces](#)” section on page 9-5 for information about the **range** keyword.

The following two examples show how to verify the configuration of GigabitEthernet interface 1/3/26:

```
Switch# show running-config interface gigabitEthernet 1/3/26
Building configuration...
```

```
Current configuration : 101 bytes
!
interface GigabitEthernet1/3/26
  no switchport
  no ip address
  channel-group 1 mode desirable
end
```

```
Switch# show interfaces gigabitEthernet 1/3/26 etherchannel
```

```
Port state      = Up Mstr In-Bndl
Channel group = 1          Mode = Desirable-Sl      Gcchange = 0
Port-channel   = Po1       GC   = 0x00010001       Pseudo port-channel = Po1
Port index     = 0         Load = 0x00             Protocol =   PAgP

Flags:  S - Device is sending Slow hello.  C - Device is in Consistent state.
        A - Device is in Auto mode.         P - Device learns on physical port.
        d - PAgP is down.

Timers:  H - Hello timer is running.        Q - Quit timer is running.
        S - Switching timer is running.     I - Interface timer is running.
```

Local information:

Port	Flags	State	Timers	Hello Interval	Partner Count	PAgP Priority	Learning Method	Group Ifindex
Gi1/3/26	SC	U6/S7	H	30s	1	128	Any	632

Partner's information:

Port	Partner Name	Partner Device ID	Partner Port	Age	Flags	Partner Group Cap.
Gi1/3/26	3750x	2c54.2dd4.ad80	Gi1/0/25	20s	SAC	50001

```
Age of the port in the current state: 0d:00h:04m:04s
Switch#
```

This example shows how to verify the configuration of port channel interface 1 after the interfaces have been configured:

```
Switch# show etherchannel 1 port-channel
Port-channels in the group:
-----
```

```
Port-channel: Po1
-----
```

```
Age of the Port-channel   = 0d:00h:53m:41s
Logical slot/port         = 21/1           Number of ports = 2
GC                         = 0x00010001
Passive port list         = Gi1/3/26 Gi2/2/26
Port state                 = Port-channel L3-Ag Ag-Inuse
Protocol                   =   PAgP
```

```
Port security          = Disabled
```

```
Ports in the Port-channel:
```

Index	Load	Port	EC state	No of bits
0	00	Gi1/3/26	Desirable-Sl	0
1	00	Gi2/2/26	Desirable-Sl	0

```
Time since last port bundled:    0d:00h:05m:25s    Gi2/2/26
Switch#
```

This example shows how to display a one-line summary per channel group:

```
Switch# show etherchannel summary
Flags:  D - down          P - bundled in port-channel
        I - stand-alone  s - suspended
        H - Hot-standby (LACP only)
        R - Layer3       S - Layer2
        U - in use       f - failed to allocate aggregator

        M - not in use, minimum links not met
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port
```

```
Number of channel-groups in use: 3
Number of aggregators:          3
```

Group	Port-channel	Protocol	Ports
1	Po1(RU)	PAgP	Gi1/3/26(P) Gi2/2/26(P)
10	Po10(SU)	-	Te1/1/1(P) Te1/1/4(D)
20	Po20(SU)	-	Te2/1/1(P)

Prior to Cisco Release IOS XE 3.5.0E and IOS 15.2(1)SG, when you tried to add a port to an EtherChannel from different chassis of the VSS system, an error message displayed:

```
Switch(config)# int gi2/3/26
Switch(config-if)# no switchport
Switch(config-if)# channel-group 50 mode on
Switch(config-if)#
Switch(config)# int gi1/1/48
Switch(config-if)# no switchport
Switch(config-if)# channel-group 50 mode on
Layer 3 MEC is not supported: GigabitEthernet1/1/48 on switch 1 cannot be part of
port-channel 50 with members on switch 2.
Command rejected: conflicts with Unsupported Layer 3 MEC
Switch(config-if)#
```

Configuring Dual-Active Detection

The following sections describe how to configure dual-active detection:

- [Configuring Enhanced PAgP Dual-Active Detection, page 5-52](#)
- [Configuring Fast-Hello Dual-Active Detection, page 5-53](#)
- [Displaying Dual-Active Detection, page 5-54](#)

Configuring Enhanced PAgP Dual-Active Detection

If enhanced PAgP is running on the MECs between the VSS and its access switches, the VSS can use enhanced PAgP messaging to detect a dual-active scenario.

By default, PAgP dual-active detection is enabled. However, the enhanced messages are only sent on port channels with trust mode enabled (see the trust mode description in the note).



Note

Before changing PAgP dual-active detection configuration, ensure that all port channels with trust mode enabled are in administrative down state. Use the **shutdown** command in interface configuration mode for the port channel. Remember to use the **no shutdown** command to reactivate the port channel when you are finished configuring dual-active detection.

To enable or disable PAgP dual-active detection, perform this task:

	Command	Purpose
Step 1	Switch(config)# switch virtual domain <i>domain_id</i>	Enters virtual switch submenu.
Step 2	Switch(config-vs-domain)# dual-active detection pagp	Enables sending of the enhanced PAgP messages.

You must configure trust mode on the port channels that will detect PAgP dual-active detection. By default, trust mode is disabled.



Note

If PAgP dual-active detection is enabled, you must place the port channel in administrative down state before changing the trust mode. Use the **shutdown** command in interface configuration mode for the port channel. Remember to use the **no shutdown** command to reactivate the port channels when you are finished configuring trust mode on the port channel.

To configure trust mode on a port channel, perform this task:

	Command	Purpose
Step 1	Switch(config)# switch virtual domain <i>domain_id</i>	Enters virtual switch submenu.
Step 2	Switch(config-vs-domain)# dual-active detection pagp trust channel-group <i>group_number</i>	Enables trust mode for the specified port channel.

This example shows how to enable PAgP dual-active detection:

```
Switch(config)# interface port-channel 20
Switch(config-if)# shutdown
Switch(config-if)# exit
Switch(config)# switch virtual domain 100
```

```
Switch(config-vs-domain)# dual-active detection pagp
Switch(config-vs-domain)# dual-active detection pagp trust channel-group 20
Switch(config-vs-domain)# exit
Switch(config)# interface port-channel 20
Switch(config-if)# no shutdown
Switch(config-if)# exit
```

This example shows the error message if you try to enable PAgP dual-active detection when a trusted port channel is not shut down first:

```
Switch(config)# switch virtual domain 100
Switch(config-vs-domain)# dual-active detection pagp
```

This example shows the error message if you try to configure trust mode for a port channel that is not shut down first:

```
Switch(config)# switch virtual domain 100
Switch(config-vs-domain)# dual-active detection pagp trust channel-group 20
```

Trusted port-channel 20 is not administratively down. To change the pagp dual-active trust configuration, "shutdown" the port-channel first. Remember to "no shutdown" the port-channel afterwards.

Configuring Fast-Hello Dual-Active Detection

To configure an interface as part of a dual-active detection pair, you need to configure dual-active fast-hello on the interface. Although fast hello dual-active detection is enabled by default, you must configure dual-active interface pairs to act as fast hello dual-active messaging links.

To enable or disable fast-hello dual-active detection, perform this task:

	Command	Purpose
Step 1	Switch(config) # switch virtual domain <i>domain_id</i>	Enters virtual switch submenu.
Step 2	Switch(config-vs-domain) # dual-active detection fast-hello	Enables the fast hello dual-active detection method. Note Fast hello dual-active detection is enabled by default.
Step 3	Switch(config-vs-domain) # exit	Exits virtual switch submenu.
Step 4	Switch(config) # interface <i>type switch/slot/port</i>	Selects the interface to configure. Note This interface must be directly connected to the other chassis and must not be a VSL link.
Step 5	Switch(config-if) # dual-active fast-hello	Enables fast hello dual-active detection on the interface, automatically removes all other configuration from the interface, and restricts the interface to dual-active configuration commands.
Step 6	Switch(config-if) # no shutdown	Activates the interface.
Step 7	Switch(config-if) # exit	Exits interface configuration mode.
Step 8	Switch(config) # exit	Exits global configuration mode.
Step 9	Switch) # show run interface <i>type switch/slot/port</i>	Displays status of dual-active fast-hello configuration.

When you configure fast hello dual-active interface pairs, note the following information:

- You can configure a maximum of four interfaces on each chassis to connect with the other chassis in dual-active interface pairs. Attempting to configure more than four interfaces causes an error message to display (and your command is rejected).
- Each interface must be directly connected to the other chassis and must not be a VSL link. We recommend using links from a switching module not used by the VSL.
- Each interface must be a physical port. Logical ports such as an SVI are not supported.
- The fast-hello links are Layer 2 ports.
- Configuring fast hello dual-active mode automatically removes all existing configuration from the interface and restricts the interface to fast hello dual-active configuration commands. It can only be used for “fast-hello” traffic.
- Unidirectional link detection (UDLD) is disabled on fast hello dual-active interface pairs.
- Do not configure fast-hello ports on an oversubscribed line card. Doing so might lead to a loss of fast-hello messages, impacting the functionality of fast-hello based dual-active detection.

This example shows how to configure an interface for fast hello dual-active detection:

```
Switch(config)# switch virtual domain 255
Switch(config-vs-domain)# dual-active detection fast-hello
Switch(config-vs-domain)# exit
Switch(config)# interface fastethernet 1/2/40
Switch(config-if)# dual-active fast-hello
WARNING: Interface FastEthernet1/2/40 placed in restricted config mode. All extraneous
configs removed!
Switch(config-if)# no shutdown
Switch(config-if)# exit
Switch(config)# exit
Switch# show run interface fastethernet 1/2/40
interface FastEthernet1/2/40
 switchport mode access
 switchport nonegotiate
 dual-active fast-hello
 no switchport
 no ip address
 dual-active fast-hello
end
```

Displaying Dual-Active Detection

To display information about dual-active detection, perform this task:

Command	Purpose
Switch# show switch virtual [dual-active {pagp fast-hello summary} link [counters detail port-channel ports] redundancy role slot-map]	Displays information about dual-active detection configuration and status.

This example shows how to display the summary status for dual-active detection:

```
Switch# show switch virtual dual-active summary
Switch(recovery-mode)# show switch virtual dual-act summary
Pagp dual-active detection enabled: Yes
In dual-active recovery mode: Yes
  Triggered by: PagP
  Triggered on Interface: Gi1/3/11
```



```
Received id: e8b7.488e.b7c0
Expected id: e8b7.488e.b700
```

This example shows how to display the summary status for dual-active detection when recovery is triggered by RRP rather than PagP:

```
Switch# show switch virtual dual-active summary
Switch(recovery-mode)# show switch virtual dual-act summary
Pagp dual-active detection enabled: Yes
In dual-active recovery mode: Yes
  Triggered by: RRP
```

This example shows how to display PAGP status and the channel groups with trust mode enabled:

```
Switch# show pagp dual-active
PAGP dual-active detection enabled: Yes
PAGP dual-active version: 1.1

Channel group 25 dual-active detect capability w/nbrs
Dual-Active trusted group: Yes
```

Port	Dual-Active Detect Capable	Partner Name	Partner Port	Partner Version
Gi1/3/11	Yes	g9-68	Gi1/11	1.1
Gi2/2/12	Yes	g9-68	Gi1/12	1.1

This example shows how to display the status of links configured as fast-hello:

```
Switch# show switch virtual dual-active fast-hello

Executing the command on VSS member switch role = VSS Active, id = 2

Fast-hello dual-active detection enabled: Yes

Fast-hello dual-active interfaces:
```

Port	Local State	Peer Port
Gi2/2/11	Dual Active Capable	Gi1/1/5

```

Executing the command on VSS member switch role = VSS Standby, id = 1

Fast-hello dual-active detection enabled: Yes

Fast-hello dual-active interfaces:
```

Port	Local State	Peer Port
Gi1/1/5	Dual Active Capable	Gi2/2/11

This example shows how to display the status of packet exchanges between the individual fast-hello links:

```
Switch# show switch virtual dual-active fast-hello counters

Executing the command on VSS member switch role = VSS Active, id = 2

Dual-active fast-hello link counters:
```

Port	Tx OK	Rx OK
Gi2/2/11	762	759

```

Executing the command on VSS member switch role = VSS Standby, id = 1
```

```

Dual-active fast-hello link counters:
      Tx      Rx
Port      OK      OK
-----
Gi1/1/5    762    759

```

This example shows how to display the status of total packets exchanged between the fast-hello links on the VSS:

```
Switch# show switch virtual dual-active fast-hello packet
```

Executing the command on VSS member switch role = VSS Active, id = 2

```

Dual-active fast-hello packet counters:
SwitchId : 2
Transmitted:
  total      = 465
Received:
  total      = 465

```

Executing the command on VSS member switch role = VSS Standby, id = 1

```

Dual-active fast-hello packet counters:
SwitchId : 1
Transmitted:
  total      = 465
Received:
  total      = 465

```

In-Service Software Upgrade (ISSU) on a VSS

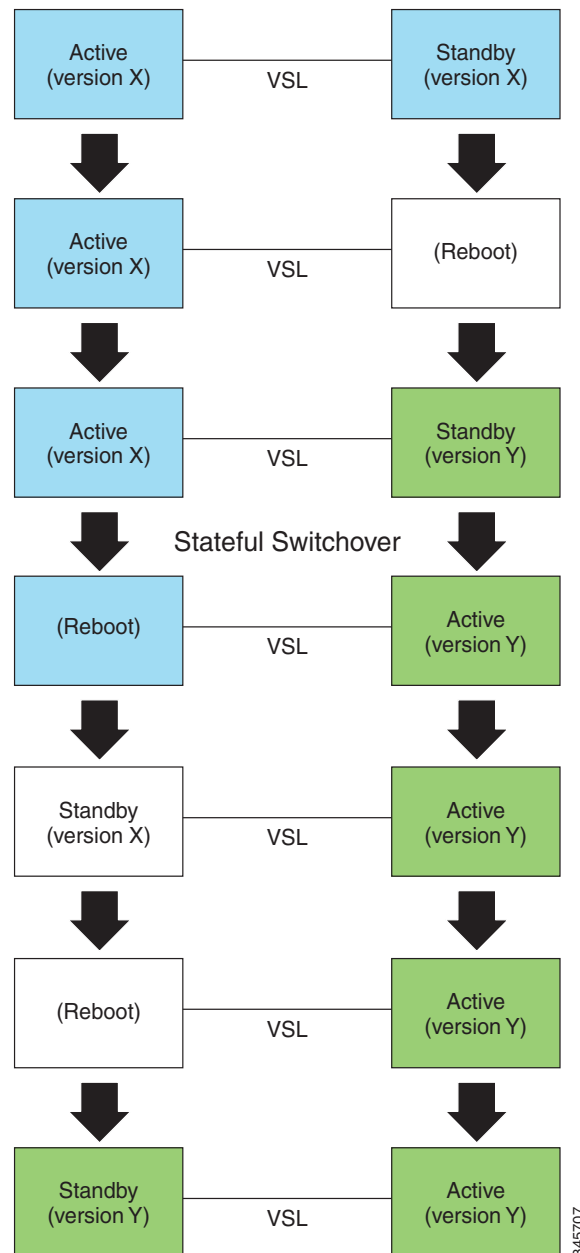
Topics include

- [VSS ISSU Concept, page 5-56](#)
- [Traffic and Network Protocol Disruption During ISSU in a VSS, page 5-57](#)
- [Related Documents, page 5-58](#)
- [Prerequisites to Performing ISSU, page 5-58](#)
- [About Performing ISSU, page 5-59](#)
- [How to Perform the ISSU Process, page 5-64](#)
- [License installation and subsequent VSS formation are now complete., page 5-86](#)

VSS ISSU Concept

In a VSS, the supervisor engines on the peer switches maintain an SSO (stateful switchover) relationship between themselves. This facilitates the ability to perform a software upgrade (or downgrade) on both the VSS supervisor engines.

[Figure 5-9](#) below depicts (at a conceptual level) the sequence of events that take place when the VSS system is upgraded from software version X to version Y.

Figure 5-9 Upgrading VSS System

Note that at any given instant, at least one of the switches is Active. The Active switch is in operation, i.e. continues to forward traffic and participate in network control protocols throughout the duration of the upgrade operation.

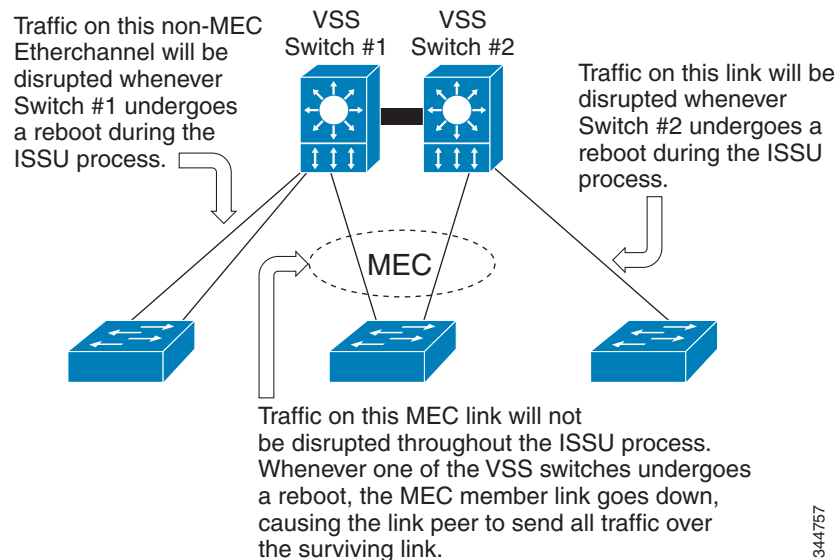
Traffic and Network Protocol Disruption During ISSU in a VSS

Figure 5-9 indicates that both switches in a VSS reboot at some point during the upgrade process.

When a switch reboots, all the network links that terminate on that switch undergo a link-down event. This means that network devices that are connected to the switch that is rebooting will observe a disruption in service, unless the connection is over an MEC that contains at least one link that terminates

on the other switch. If a peer device is connected to the VSS over an MEC that has links terminating in both switches, that device will not experience a disruption of service during the software upgrade process. This is illustrated in Figure 5-10.

Figure 5-10 Connecting a Peer Device to VSS to Avoid Service Disruption



Related Documents

Related Topic	Document Title
Performing ISSU	<i>Cisco IOS Software: Guide to Performing In Service Software Upgrades</i>

Prerequisites to Performing ISSU

Before performing ISSU, you must meet these prerequisites:

- Ensure that the current Cisco IOS XE version running in the system supports ISSU. Also ensure that the target version supports ISSU.
- In a Quad-Supervisor VSS setup, ensure that you copy the Cisco IOS XE image on the standby supervisors of both the switches, from the VSS Active switch console, using the IOS **copy** command so that the boot variable has a valid path to point to.

You can enter various commands on the switch to determine supervisor engine versioning and Cisco IOS XE software compatibility. Alternatively, you can use the ISSU application on Cisco Feature Navigator to determine this.

- The type of the pre- and post-upgrade images must match precisely. Identical. ISSU is not supported from a Universal_lite to a Universal image, or vice versa. ISSU is also not supported from a k9 image to a non-k9 image, or vice versa.

- VSS must be functionally in SSO mode; that is, both switches must be powered up and operational, with one supervisor engine running as the SSO active, and the other as the SSO standby.
- The pre- and post-upgrade Cisco IOS XE software image files must both be available in the local file systems (bootflash, SD card, or USB) of both the Active and the standby supervisor engines before you begin the ISSU process.

Both supervisor engines should be running the pre-upgrade image, and should have booted from the image location in the local file system. (bootflash, SD card, or USB).



Note Beginning in Cisco IOS XE release 3.8.0E, in VSS mode, `usb0:` and `slot0:` are not available on the in-chassis standby (ICS).



Note The **show version** command can be used to confirm that the supervisor engine has actually booted from the pre-upgrade image location in the local filesystem.



Note Auto-boot must be enabled in the rommon for ISSU to succeed. The config-register value displayed in the output of **show version** can be used to confirm this.

- It is advisable to take measures to mitigate the effects of switch down-time. ISSU in a VSS will result in loss of service on non-MEC links, and peers must be prepared for this. On links connected over MECs, Nonstop Forwarding (NSF) must be configured and working properly. If you do not have NSF enabled, see the *Cisco Nonstop Forwarding* document for further information on how to enable and configure NSF.
- Autoboot is turned on and the current booted image matches the one specified in the BOOT environmental variable. For details on how to configure and verify these, please refer to [“Modifying the Boot Field and Using the boot Command”](#) section on page 3-28.
- The **no ip routing** command is not supported - both before starting the ISSU process, and at any time during the ISSU process.
- Save the image on the same partition in both supervisor engines (e.g. if it is saved at slot0: in the active supervisor engine it should be saved at slaveslot0:). Similarly if it is at bootflash: in the active supervisor engine, it should be at savebootflash: in the standby supervisor engine.

About Performing ISSU



Note Do not make any hardware changes while performing ISSU.

Before you perform ISSU, you should understand the following concepts:

- [Performing an ISSU Upgrade: Two Methods](#), page 5-59
- [Guidelines for Performing ISSU](#), page 5-63

Performing an ISSU Upgrade: Two Methods

There are two ways to perform an ISSU upgrade:

- manually using a sequence of four commands
- automatically; using a single command

ISSU using the four-command sequence

The manual ISSU upgrade process involves issuing four distinct ISSU EXEC commands in sequence

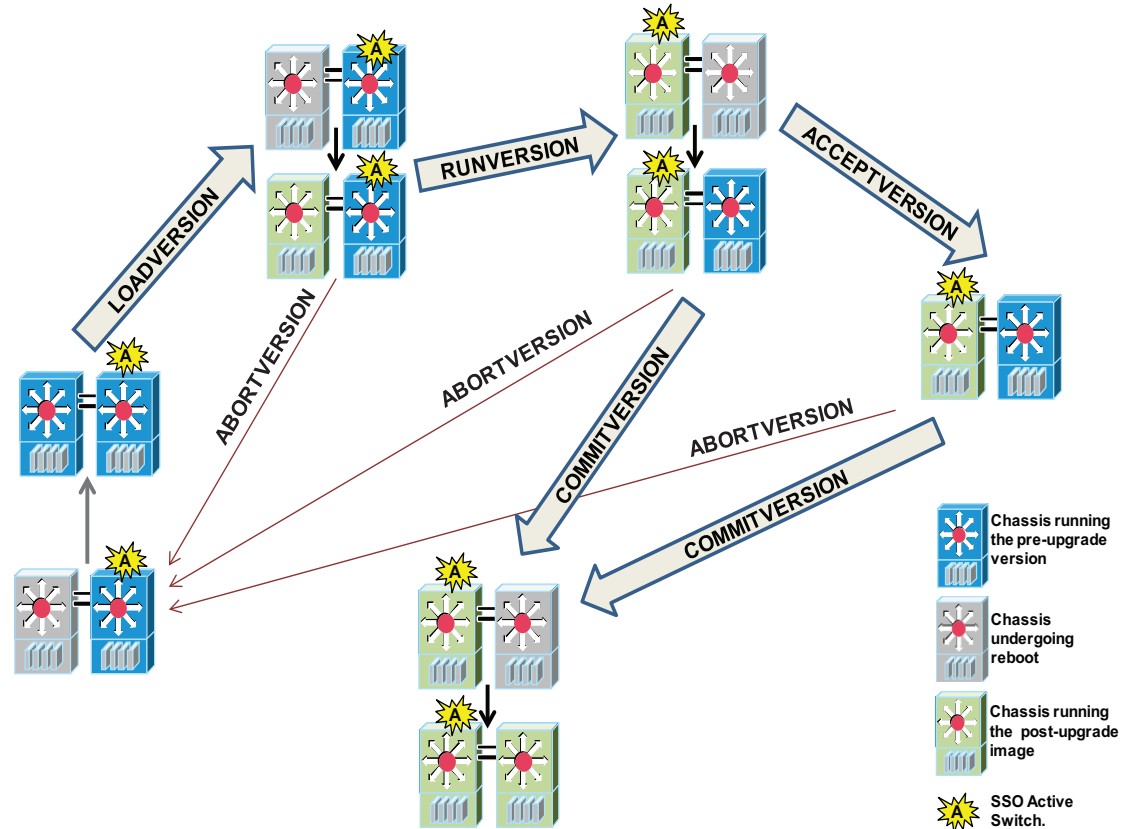
- **issu loadversion**
- **issu runversion**
- **issu acceptversion**
- **issu commitversion**

A fifth command, **issu abortversion**, enables you to abort the ISSU upgrade process at any time, and to revert to the initial system state.

These four commands take the VSS through a series of states that culminate in the Active and standby supervisor engines running the post-upgrade IOS XE image. The VSS continues to operate throughout the entire process; however as explained in [Traffic and Network Protocol Disruption During ISSU in a VSS, page 5-57](#), service is disrupted on network links that terminate on interfaces that reside in the switch that is undergoing a reboot.

[Figure 5-11](#) depicts the states through which the VSS Active and standby supervisor engines progress as the sequence of four commands entered. It also shows the effect of the **issu abortversion** command at any given point during the process.

Figure 5-11 States of VSS Active and Standby during Command Execution



34-4800

During the ISSU process, several **show** commands are available to evaluate the success of each command before proceeding to the next step.

ISSU using the Single Command Sequence (**issu changeversion**)

The use of multiple ISSU commands dictates an additional level of care to ensure no service disruption. However, in some scenarios, this upgrade procedure might be cumbersome and of minimal value. A typical example is during a network upgrade that involves performing an ISSU upgrade on a large number of Catalyst 4500 switches. In these cases, we recommend that you first perform the manual (four command) ISSU upgrade procedure on one VSS (possibly in a lab environment) to verify successful upgrade. Then, use the single **issu changeversion** procedure to perform an automatic ISSU on the rest of the Catalyst 4500 switches in the network.

The **issu changeversion** command launches a single-step complete ISSU upgrade cycle. It performs the logic for all four of the standard commands (**issu loadversion**, **issu runversion**, **issu acceptversion**, and **issu commitversion**) without user intervention, streamlining the upgrade through a single CLI step.

Additionally, **issu changeversion** allows the upgrade process to be scheduled for a future time. This enables you to stage a number of systems to perform upgrades sequentially when a potential disruption would be least harmful.

After the standby supervisor engine initializes and the system reaches a terminal state (SSO), the upgrade process is complete and the BOOT variable is permanently written with the new IOS XE software image. Hence, a reset on any RP will keep the system booting the new software image. Console and syslog messages will be generated to notify anyone monitoring the upgrade that the state transition has occurred.

Similar to the normal ISSU upgrade procedure, the in-progress upgrade procedure initiated by the **issu changeversion** command can be aborted with the **issu abortversion** command. If the system detects any problems or detects an unhealthy system during an upgrade, the upgrade might be automatically aborted.

When the **issu runversion** command is entered during the four step manual upgrade process, if any incompatible ISSU clients exist, the upgrade process reports them and their side effects, and allows the user to abort the upgrade. While performing a single-step upgrade process, when the process reaches the runversion state, it will either automatically continue with the upgrade provided the base clients are compatible, or automatically abort because of client incompatibility.

Changeversion: “quick” option

The **issu changeversion** command provides a “quick” option that can reduce the time required to perform the automatic ISSU upgrade. When the **quick** command option is applied, the ISSU upgrade state transition differs from that illustrated in [Figure 5-9](#). With this option, the state progression up to the loadversion stage remains the same as described in the figure, but the runversion and commitversion stages are combined. This progression skips the step in the upgrade procedure that loads the old software version on the new standby (old active) supervisor, thereby reducing the time required for the automatic ISSU upgrade by about a third.

Scheduled Changeversion: “in” and “at” Options

issu changeversion provides **in** and **at** command options that enable you to schedule a future automatic ISSU upgrade.

The **at** command option schedules an automatic ISSU upgrade to begin at a specific time. This option specifies an exact time (*hh:mm*, 24 hour format) in the next 24 hours at which the upgrade will occur.

The **in** command option schedules an automatic ISSU upgrade to begin after a certain amount of time has elapsed. This option specifies the number of hours and minutes (*hh:mm* format) that must elapse before an upgrade will occur, with a maximum value of 99:59.

Changeversion Deployment Scenario

The typical **issu changeversion** command usage scenario is for experienced users with a large installed base. These users typically validate a new image using a topology and configuration similar to their production network. The validation process should be done using both the existing multi-command process and the new **issu changeversion** command process. Once users certify an IOS XE software image and want to roll it out broadly, they can use the single command process to perform an efficient upgrade of their network.

Aborting an In-Progress Changeversion Procedure

The **issu changeversion** command functionality is designed to perform an ISSU software upgrade without user intervention. However, status messages are displayed to the console as the upgrade transitions through the various states. If any anomalies are noticed during the automatic upgrade, perhaps with peers or other parts of the network, you can use the **issu abortversion** command to manually abort the upgrade at any point in the process prior to the commitversion operation.

Guidelines for Performing ISSU

Be aware of the following guidelines while performing the ISSU process:

- Even with ISSU, it is recommended that upgrades be performed during a maintenance window.
- As explained in [Traffic and Network Protocol Disruption During ISSU in a VSS, page 5-57](#), ISSU on VSS may cause loss of network connectivity to both the VSS switches at some point during the process (although not at the same time). The mitigation steps as explained in that section must be implemented.
- The new features should not be enabled (if they require change of configuration) during the ISSU process.



Note Enabling them will cause the system to enter RPR mode because commands are only supported on the new version.

- In a downgrade scenario, if any feature is not available in the downgrade revision of the Cisco IOS XE software handle, that feature should be disabled prior to initiating the ISSU process.



Note On a Catalyst 4500 switch in VSS or a standalone chassis with redundant supervisors, an ISSU upgrade from any prior release to the Cisco IOS XE Release 3.11.10E specifically will fail. Instead, when upgrading to the Cisco IOS XE Release 3.11.10E, you must perform a reload-based upgrade where both the chassis or both supervisors are reloaded at the same time.

Compatibility Matrix

ISSU requires additional information to determine compatibility between software versions. Therefore, a compatibility matrix is defined that contains information about other IOS XE software image with respect to the one in question.

This compatibility matrix represents the compatibility of two software versions, one running on the active and the other on the standby supervisor engine, and to allow the system to determine the highest operating mode it can achieve. Incompatible versions will not be able to progress to SSO operational mode. Because SSO is a pre-requisite for a VSS, incompatibility will also lead to the loss of VSS relationship between the supervisors engines in the two switches.

The compatibility matrix represents the compatibility relationship a Cisco IOS XE software image has with all of the other Cisco IOS XE software versions within the designated support window (for example, all of those software versions the IOS XE software image “knows” about) and is populated and released with every IOS XE software image. The matrix stores compatibility information between its own release and prior releases. It is always the newest release that contains the latest information about compatibility with existing releases in the field. The compatibility matrix is available within the Cisco IOS XE software image and on Cisco.com so that users can determine in advance whether a successful upgrade can be achieved using the ISSU process.

You can perform the ISSU process when the old and new Cisco IOS XE software are compatible. The compatibility matrix information stores the compatibility among releases as follows:

- **Compatible**—The base-level system infrastructure and all optional HA-aware subsystems are compatible. An in-service upgrade or downgrade between these versions will succeed with minimal service impact. The matrix entry designates the images to be compatible (C).
- **Base-level compatible**—One or more of the optional HA-aware subsystems is not compatible. An in-service upgrade or downgrade between these versions will succeed; however, some subsystems will not be able to maintain state always during the transition from the old to the new version of Cisco IOS XE. The matrix entry designates the images to be base-level compatible (B).
- **Incompatible**—A core set of system infrastructure exists in Cisco IOS XE that must be able to interoperate in a stateful manner for SSO to function correctly. If any of these required features or subsystems is not interoperable, then the two versions of the Cisco IOS XE software image are declared to be incompatible. An in-service upgrade or downgrade between these versions is not

possible. The matrix entry designates the images to be incompatible (I). The system operates in RPR mode during the upgrade process when the versions of Cisco IOS XE at the active and standby supervisor engines are incompatible.

- Cisco IOS XE determines the compatibility between the active and the standby IOS XE software dynamically during Standby boot up. The matrix is represented by “x”.

To display the compatibility matrix data between two software versions on a given system, enter the **show issu comp-matrix stored** command.

**Note**

This command is useful *only for verification purposes* because it is available *only after* the ISSU process has started. You might want to check the compatibility matrix prior to starting ISSU. Use the Feature Navigator to obtain the needed information:

<http://tools.cisco.com/ITDIT/CFN/jsp/index.jsp>

Compatibility Verification Using Cisco Feature Navigator

The ISSU application on Cisco Feature Navigator allows you to:

- Select a specific software bundle.
- Identify which software images are compatible with the selected software image.
- Compare two IOS XE software images and understand the compatibility level of the software images (that is, compatible, base-level compatible, and incompatible), or dynamically determined.
- Compare two software images and see the client compatibility for each ISSU client.
- Provide links to release notes for the software image.

How to Perform the ISSU Process

Unlike SSO, which is a mode of operation for the device and a prerequisite for performing ISSU, the ISSU process is a series of steps performed while the switch is in operation. The steps result in an upgrade to new or modified Cisco IOS XE software, and have a minimal impact to traffic.

**Note**

For an illustration of the process flow for ISSU, refer to [Figure 5-11](#).

This section includes the following topics:

- [Verifying the ISSU Software Installation, page 5-65](#)
- [Verifying Redundancy Mode Before Beginning the ISSU Process, page 5-65](#)
- [Verifying the ISSU State Before Beginning the ISSU Process, page 5-67](#)
- [ISSU using the Four-command Sequence: Step 1 \(loadversion\), page 5-68](#)
- [ISSU using the Four-command Sequence: Step 2 \(runversion\), page 5-70](#)
- [ISSU using the Four Command Sequence: Step 3 \(acceptversion\), page 5-72](#)
- [ISSU using the Four Command Sequence: Step 4 \(commitversion\), page 5-72](#)
- [Using changeversion to Automate an ISSU Upgrade, page 5-73](#)

- [Aborting a Software Upgrade During ISSU, page 5-79](#)
- [Configuring the Rollback Timer to Safeguard Against Upgrade Issues, page 5-81](#)
- [The ISSU Compatibility Matrix, page 5-82](#)

Verifying the ISSU Software Installation

During the ISSU process, there are five valid states: disabled, init, load version, run version, and system reset. Use the **show issu state** command to obtain the current ISSU state:

- Disabled state—The state for the standby supervisor engine while this supervisor engine is resetting.
- Init state—The initial state for two supervisor engines, one active and one standby, before the ISSU process is started. It is also the final state after the ISSU process completes.
- Load version (LV) state—The standby supervisor engine is loaded with the new version of Cisco IOS XE software.
- Run version (RV) state—The **issu runversion** command forces the switchover of the supervisor engines. The newly active supervisor engine runs the new Cisco IOS XE software image.
- While running ISSU, if both supervisor engines are reset (because of a power outage, for example), the ISSU context is lost and the system returns to the Init state. Both supervisor engines return to the old software.

You can verify the ISSU software upgrade by entering **show** commands to provide information on the state of the during the ISSU process:

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	Switch# show issu state [detail]	Displays the current state of the ISSU process.
Step 3	Switch# show redundancy	Displays current or historical status, mode, and related redundancy information about the device.
Step 4	Switch# show switch virtual	Identifies which switch of the VSS is currently performing the Active role, and which switch the Standby.

This example shows how to display the state and the current status of the supervisor engine during the ISSU process:

```
Switch> enable
Switch# show issu state
Switch# show redundancy
Switch# show switch virtual
```

Verifying Redundancy Mode Before Beginning the ISSU Process

Before you begin the ISSU process, verify that the VSS is operating correctly; one supervisor engine operates as the SSO Active and the peer supervisor engine in the other switch operating as the SSO Hot Standby.

The following example displays verification that the system operating correctly as a VSS. Slot 1/1 (the supervisor engine in slot 1 of Switch 1) is the active supervisor engine, and Slot 2/1 (the supervisor engine in slot 1 of Switch 2) is the standby supervisor engine.

```

Switch# show redundancy states
    my state = 13 -ACTIVE
    peer state = 8 -STANDBY HOT
        Mode = Duplex
        Unit = Primary
        Unit ID = 1

Redundancy Mode (Operational) = Stateful Switchover
Redundancy Mode (Configured) = Stateful Switchover
    Redundancy State = Stateful Switchover
        Manual Swact = enabled

Communications = Up

    client count = 77
    client_notification_TMR = 240000 milliseconds
        keep_alive TMR = 9000 milliseconds
        keep_alive count = 0
    keep_alive threshold = 18
        RF debug mask = 0

Switch# show redundancy
Redundant System Information :

-----
    Available system uptime = 11 minutes
Switchovers system experienced = 0
    Standby failures = 0
    Last switchover reason = none

        Hardware Mode = Duplex
    Configured Redundancy Mode = Stateful Switchover
    Operating Redundancy Mode = Stateful Switchover
        Maintenance Mode = Disabled
        Communications = Up

Current Processor Information :
-----
    Active Location = slot 1/1
    Current Software state = ACTIVE
    Uptime in current state = 9 minutes
        Image Version = Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3
Switch Software (cat4500e-UNIVERSAL-M), Version 03.03.00.SGN1.33 CISCO INTERNAL USE ONLY
UNIVERSAL PRODUCTION K10 IOSD VERSION , synced to END_OF_FLO_ISP
Copyright (c) 1986-2012 by Cisco Systems, Inc.
Compiled Thu 09-Aug
        BOOT =
bootflash:cat4500e-universal.SSA.03.03.00.SGN1.33.151-2.SGN1.33.bin,12;
    Configuration register = 0x2102

Peer Processor Information :
-----
    Standby Location = slot 2/1
    Current Software state = STANDBY HOT
    Uptime in current state = 8 minutes
        Image Version = Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3
Switch Software (cat4500e-UNIVERSAL-M), Version 03.03.00.SGN1.33 CISCO INTERNAL USE ONLY
UNIVERSAL PRODUCTION K10 IOSD VERSION , synced to END_OF_FLO_ISP
Copyright (c) 1986-2012 by Cisco Systems, Inc.
Compiled Thu 09-Au
        BOOT =
bootflash:cat4500e-universal.SSA.03.03.00.SGN1.33.151-2.SGN1.33.bin,12;
    Configuration register = 0x2102

```

```
Switch# show switch virtual
Switch mode           : Virtual Switch
Virtual switch domain number : 16
Local switch number   : 1
Local switch operational role: Virtual Switch Active
Peer switch number    : 2
Peer switch operational role : Virtual Switch Standby
Switch#
```

Verifying the ISSU State Before Beginning the ISSU Process

Ensure that both the supervisor engines are configured to auto-boot, and that they have currently been booted from the pre-upgrade image residing on the local file system. This is verified by the values of the BOOT variable and the configuration register (refer to the sample output of **show redundancy** command in the previous section).

Ensure that the active and standby supervisor engines are up and in ISSU Init state and that both supervisor engines are running the same current image.

The following example displays the ISSU state before the process begins:

```
Switch# show issu state detail
                Slot = 1
                RP State = Active
                ISSU State = Init
                Operating Mode = Stateful Switchover
                Current Image =
bootflash:cat4500e-universal.SSA.03.03.00.SGN1.33.151-2.SGN1.33.bin
                Pre-ISSU (Original) Image = N/A
                Post-ISSU (Targeted) Image = N/A

                Slot = 11
                RP State = Standby
                ISSU State = Init
                Operating Mode = Stateful Switchover
                Current Image =
bootflash:cat4500e-universal.SSA.03.03.00.SGN1.33.151-2.SGN1.33.bin
                Pre-ISSU (Original) Image = N/A
                Post-ISSU (Targeted) Image = N/A
```

Note that the Standby slot number is reported as 11, which is a Virtual Slot number that corresponds to physical slot 1 on Switch 2. The correspondence between the Virtual Slot number and the physical location of the slot can be determined using the **show switch virtual slot-map** command, as shown in the following example:

```
Switch# show switch virtual slot-map
Virtual Slot to Remote Switch/Physical Slot Mapping Table:
```

Virtual Slot No	Remote Switch No	Physical Slot No	Module Uptime
1	1	1	00:33:04
2	1	2	00:32:50
3	1	3	00:32:36
4	1	4	-
5	1	5	-
6	1	6	-
7	1	7	-
8	1	8	-
9	1	9	-

```

10      1      10      -
11      2      1      00:31:14
12      2      2      00:33:33
13      2      3      00:33:33
14      2      4      -
15      2      5      -
16      2      6      -
17      2      7      -
18      2      8      -
19      2      9      -
20      2     10      -
Switch#

```

The new version of the Cisco IOS XE software must be present on both of the supervisor engines. The directory information displayed for each of the supervisor engines shows that the new version is present.

```

Switch# dir bootflash:
Directory of bootflash:/

29122  -rw-   119519232  Aug 13 2012 19:13:14 +00:00
cat4500e-universal.SSA.03.03.00.SGN1.34.151-2.SGN1.34.bin
29125  -rw-   119286584  Aug 13 2012 22:30:02 +00:00
cat4500e-universal.SSA.03.03.00.SGN1.33.151-2.SGN1.33.bin

820875264 bytes total (581672960 bytes free)
Switch# dir slavebootflash:
Directory of slavebootflash:/

58370  -rw-   119286584  Aug 14 2012 11:25:38 +00:00
cat4500e-universal.SSA.03.03.00.SGN1.33.151-2.SGN1.33.bin
58372  -rw-   119519232  Aug 14 2012 11:40:47 +00:00
cat4500e-universal.SSA.03.03.00.SGN1.34.151-2.SGN1.34.bin

822910976 bytes total (583716864 bytes free)
Switch#

```

ISSU using the Four-command Sequence: Step 1 (loadversion)

This task describes the first step of the ISSU four-command sequence, loadversion, wherein the standby supervisor engine is loaded with the post-upgrade image.

Please ensure that you have read the [Prerequisites to Performing ISSU, page 5-58](#) section, and implemented the appropriate steps.

Perform the following steps at the active supervisor engine:

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	Switch# issu loadversion [<i>active-slot</i>] <i>active-image-new</i> [<i>standby-slot</i>] <i>standby-image-new</i>	Starts the ISSU process and (optionally) overrides the automatic rollback when the new Cisco IOS XE software version is detected to be incompatible. It may take several minutes after entering the issu loadversion command for Cisco IOS XE software to load onto the standby supervisor engine and for the standby supervisor engine to transition to SSO mode. This causes the standby supervisor engine to reload with the new software image. If used, the <i>active-slot</i> and <i>standby-slot</i> numbers should be specified as Virtual Slot numbers. Use the show switch virtual slot-map command to determine the correspondence between Virtual slot numbers and their physical locations.
Step 3	Switch# show issu state [<i>detail</i>]	Displays the state of ISSU during the ISSU process. At this point in the ISSU process, use this command to check that the standby supervisor engine is loaded and is in SSO mode. It may take several minutes after entering the issu loadversion command for Cisco IOS XE software to load onto the standby supervisor engine and the standby supervisor engine to transition to SSO mode. If you enter the show issu state command too quickly, you may not see the information you need.
Step 4	Switch# show redundancy [<i>states</i>]	Displays redundancy facility state information.

This example shows how to start the ISSU process, boot the standby supervisor engine in the Standby Hot state, and load the standby supervisor engine slot (6) with the new IOS XE software image:

```
Switch> enable
Switch# issu loadversion 1 bootflash:new_image 11 slavebootflash:new_image
%issu loadversion executed successfully, Standby is being reloaded
Switch# show issu state detail
      Slot = 1
      RP State = Active
      ISSU State = Load Version
      Operating Mode = Stateful Switchover
      Current Image = bootflash:old_image
      Pre-ISSU (Original) Image = bootflash:old_image
      Post-ISSU (Targeted) Image = bootflash:new_image

      Slot = 11
      RP State = Standby
      ISSU State = Load Version
      Operating Mode = Stateful Switchover
      Current Image = bootflash:new_image
      Pre-ISSU (Original) Image = bootflash:old_image
      Post-ISSU (Targeted) Image = bootflash:new_image
```

```

Switch# show redundancy states
    my state = 13 -ACTIVE
    peer state = 8 -STANDBY HOT
        Mode = Duplex
        Unit = Primary
        Unit ID = 1

Redundancy Mode (Operational) = Stateful Switchover
Redundancy Mode (Configured) = Stateful Switchover
    Redundancy State = Stateful Switchover
        Manual Swact = enabled

Communications = Up

    client count = 474
    client_notification_TMR = 240000 milliseconds
        keep_alive TMR = 9000 milliseconds
        keep_alive count = 1
    keep_alive threshold = 18
        RF debug mask = 0

```

ISSU using the Four-command Sequence: Step 2 (runversion)

This task describes the second step of the ISSU four-command sequence, runversion, wherein a switchover occurs and the standby supervisor engine, which is now loaded with the post-upgrade image, takes over as the new Active.

At the end of the loadversion step, the following message is logged:

```
*Aug 14 13:07:08.240: %INSTALLER-7-ISSU_OP_SUCC: Peer state is [STANDBY SSO]; Please issue the runversion command
```

Now, you are ready to proceed to the next step.

Perform the following steps at the active supervisor engine.

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	Switch# issu runversion [<i>standby-slot</i>] [<i>standby-image-new</i>]	Forces a switchover from the active to the standby supervisor engine and reloads the former active (current standby) supervisor engines with the old IOS XE image. As with any SSO switchover, you are prompted to save the running configuration if you have changed it. Respond as appropriate. When you enter the issu runversion command, an SSO switchover will be performed, and NSF procedures will be invoked if so configured.
Step 3	Switch# show issu state [<i>detail</i>]	Displays the state of the during the ISSU process. At this point in the ISSU process, use this command to check that a switchover occurs to slot 11.
Step 4	Switch# show redundancy [<i>states</i>]	Displays redundancy facility state information.

This example shows how to cause a switchover to the former standby supervisor engine (slot 11), reset the former active supervisor engine and reload it with the old IOS XE software image so it becomes the standby supervisor engine:

```
Switch> enable
Switch# issu runversion 11 slavebootflash:new_image
%issu runversion initiated successfully
```

A switchover happens at this point. At the new active supervisor engine, do the following after old active supervisor engine comes up as standby.

```
Switch# show issu state detail

          Slot = 11
          RP State = Active
          ISSU State = Run Version
          Operating Mode = Stateful Switchover
          Current Image = bootflash:new_image
          Pre-ISSU (Original) Image = bootflash:old_image
          Post-ISSU (Targeted) Image = bootflash:new_image

          Slot = 1
          RP State = Standby
          ISSU State = Run Version
          Operating Mode = Stateful Switchover
          Current Image = bootflash:old_image
          Pre-ISSU (Original) Image = bootflash:old_image
          Post-ISSU (Targeted) Image = bootflash:new_image
```



Note

The new active supervisor engine is now running the new version of software, and the standby supervisor engine is running the old version of software and is in the standby hot state.

```
Switch# show redundancy states

my state = 13 -ACTIVE
peer state = 8 -STANDBY HOT
  Mode = Duplex
  Unit = Primary
  Unit ID = 11

Redundancy Mode (Operational) = Stateful Switchover
Redundancy Mode (Configured) = Stateful Switchover
  Redundancy State = Stateful Switchover
  Manual Swact = enabled

Communications = Up

  client count = 474
  client_notification_TMR = 240000 milliseconds
  keep_alive TMR = 9000 milliseconds
  keep_alive count = 0
  keep_alive threshold = 18
  RF debug mask = 0
```

Once Runversion has completed, the new active supervisor engine will be running the new version of software and the previously active supervisor engine will now become the standby supervisor engine. The standby supervisor engine will be reset and reloaded, but it will remain on the previous version of software and come back online in Standby hot status.

Use the **show redundancy**, **show redundancy states**, and **show issu state [detailed]** commands described previously to verify that the standby supervisor engine is running the pre-upgrade version and that the Active is running the post-upgrade version.

ISSU using the Four Command Sequence: Step 3 (acceptversion)

This step is optional. It is needed only if you wish to stop the ISSU rollback timer. Otherwise you may proceed to the next step (**commitversion**)

Cisco IOS XE software maintains an ISSU rollback timer to safeguard against an upgrade that may leave the new active supervisor engine in a state in which communication with the standby supervisor engine is severed. By default, this duration is 45 minutes. If the **commitversion** command is not applied before the rollback timer duration expires, the VSS reverts to the pre-upgrade version.

The **acceptversion** command stops the rollback timer. This means that you can maintain the system in the current state (runversion, with the post-upgrade version running on the active supervisor engine, and pre-upgrade image running on the standby supervisor engine) for an extended duration, and proceed to the commitversion state only when you are satisfied with the behavior of the post-upgrade software version.

This optional task describes how to stop the rollback timer.

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	Switch# issu acceptversion [<i>active-slot</i>] [<i>active-image-new</i>]	Halts the rollback timer and ensures the new Cisco IOS XE ISSU process is not automatically aborted during the ISSU process. Enter the issu acceptversion command within the time period specified by the rollback timer to acknowledge that the supervisor engine has achieved connectivity to the outside world; otherwise, the ISSU process is terminated, when the rollback timer expires, and the system reverts to the previous version of Cisco IOS XE software by switching to the standby supervisor engine.
Step 3	Switch# show issu rollback-timer	Displays the amount of time left before an automatic rollback will occur.

This example displays the Timer before you stop it. In the following example, the “Automatic Rollback Time” information indicates the amount of time remaining before an automatic rollback will occur.

```
Switch> enable
Switch# show issu rollback-timer
    Rollback Process State = 00:31:09 remaining
    Configured Rollback Time = 00:45:00

Switch# issu acceptversion 611 bootflash:new_image
% Rollback timer stopped. Please issue the commitversion command.
Switch# show issu rollback-timer
    Rollback Process State = Not in progress
    Configured Rollback Time = 00:45:00
```

ISSU using the Four Command Sequence: Step 4 (commitversion)

The commitversion step reloads the standby supervisor engine with the post-upgrade image.

Perform the following steps at the active supervisor engine:

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	Switch# issu commitversion [<i>standby-slot</i>] [<i>standby-image-new</i>]	Allows the new Cisco IOS XE software image to be loaded into the standby supervisor engine.
Step 3	Switch# show redundancy [<i>states</i>]	Displays redundancy facility state information.
Step 4	Switch# show issu state [<i>detail</i>]	Displays the state of the during the ISSU process. At this point in the ISSU process, use this command to check that the standby supervisor engine is loaded with the new image.

This example shows how to reset and reload the current standby supervisor engine (slot 1) with the new Cisco IOS XE software version. After you enter the **commitversion** command, the standby supervisor engine boots in the Standby Hot state.

```
Switch> enable
Switch# issu commitversion 1 slavebootflash:new_image
%issu commitversion executed successfully
```

As in prior states, the **show redundancy**, **show redundancy states**, **show issu state [detailed]**, and **show switch virtual** commands can be used to verify that the VSS has reached the desired state.

At the end of the commitversion state, the ISSU process has completed. At this stage, any further Cisco IOS XE software version upgrade or downgrade will require that a new ISSU process be invoked anew.

Using changeversion to Automate an ISSU Upgrade

This task describes how to use the **issu changeversion** command to perform a one step ISSU upgrade.

Please ensure that you have read [Prerequisites to Performing ISSU, page 5-58](#), and implemented the appropriate steps.

Perform the following steps at the active supervisor engine:

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	Switch# issu changeversion [<i>active-slot</i> <i>active-image-new</i>] [<i>standby-slot</i> <i>standby-image-new</i>] [at <i>hh:mm</i> in <i>hh:mm</i>] [quick]	Initiates a single-step complete upgrade process cycle. Performs the logic of the four standard commands (issu loadversion, issu runversion, issu acceptversion, and issu commitversion) without user intervention. <i>active-slot</i> —Defines the active slot number (the Virtual slot number). Use the show switch virtual slot-map command to determine the virtual slot number from the physical slot number. <i>new-image</i> —Specifies IOS XE image URL to be upgraded to. <i>standby-slot</i> —Defines the standby slot number (the Virtual slot number). <i>standby-image</i> —Specifies the standby IOS XE image URL. at hh:mm —Schedules an ISSU upgrade to begin in the future. Provides an exact time (<i>hh:mm</i> , 24 hour format) in the next 24 hours when the upgrade will occur. in hh:mm —Schedules an ISSU upgrade to begin in the future. Provides the number of hours and minutes (<i>hh:mm</i> format) that will elapse before an upgrade will occur (99:59 max). quick —Upon switchover, boots the standby supervisor engine with the new, rather than old, image for faster upgrade.
Step 3	Switch# show issu state [detail]	Displays the state of the during the ISSU process. At this point in the ISSU process, use this command to check that the standby supervisor engine is loaded and is in SSO mode.
Step 4	Switch# show redundancy [states]	Displays redundancy facility state information.

This example shows how to initiate an ISSU upgrade process using the **issu changeversion** command. The outputs of the **show switch virtual**, **show issu state detail**, **show redundancy**, and **show redundancy states** commands are included to show the supervisor state before and after the upgrade procedure.

```
Switch> enable
Switch# show switch virtual
Switch mode           : Virtual Switch
Virtual switch domain number : 16
Local switch number   : 1
Local switch operational role: Virtual Switch Active
Peer switch number    : 2
Peer switch operational role : Virtual Switch Standby
Switch#
Switch#show redundancy states
      my state = 13 -ACTIVE
      peer state = 8 -STANDBY HOT
```

```

        Mode = Duplex
        Unit = Primary
        Unit ID = 1

Redundancy Mode (Operational) = Stateful Switchover
Redundancy Mode (Configured) = Stateful Switchover
Redundancy State = Stateful Switchover
Manual Swact = enabled

Communications = Up

        client count = 74
        client_notification_TMR = 240000 milliseconds
        keep_alive TMR = 9000 milliseconds
        keep_alive count = 0
        keep_alive threshold = 18
        RF debug mask = 0

Switch# show redundancy
Redundant System Information :

-----
        Available system uptime = 3 hours, 50 minutes
        Switchovers system experienced = 2
        Standby failures = 1
        Last switchover reason = active unit removed

        Hardware Mode = Duplex
        Configured Redundancy Mode = Stateful Switchover
        Operating Redundancy Mode = Stateful Switchover
        Maintenance Mode = Disabled
        Communications = Up

Current Processor Information :
-----
        Active Location = slot 1/1
        Current Software state = ACTIVE
        Uptime in current state = 45 minutes
        Image Version = Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3
Switch Software (cat4500e-UNIVERSAL-M), Version 03.03.00.SGN1.33 CISCO INTERNAL USE ONLY
UNIVERSAL PRODUCTION K10 IOSD VERSION , synced to END_OF_FLO_ISP
Copyright (c) 1986-2012 by Cisco Systems, Inc.
Compiled Thu 09-Aug
        BOOT =
bootflash:cat4500e-universal.SSA.03.03.00.SGN1.33.151-2.SGN1.33.bin,12;bootflash:cat4500e-
universal.SSA.03.03.00.SGN1.34.151-2.SGN1.34.bin,12;
        Configuration register = 0x2102

Peer Processor Information :
-----
        Standby Location = slot 2/1
        Current Software state = STANDBY HOT
        Uptime in current state = 25 minutes
        Image Version = Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3
Switch Software (cat4500e-UNIVERSAL-M), Version 03.03.00.SGN1.33 CISCO INTERNAL USE ONLY
UNIVERSAL PRODUCTION K10 IOSD VERSION , synced to END_OF_FLO_ISP
Copyright (c) 1986-2012 by Cisco Systems, Inc.
Compiled Thu 09-Au
        BOOT =
bootflash:cat4500e-universal.SSA.03.03.00.SGN1.33.151-2.SGN1.33.bin,12;bootflash:cat4500e-
universal.SSA.03.03.00.SGN1.34.151-2.SGN1.34.bin,12;
        Configuration register = 0x2102

```

Switch# **show issu state detail**

```

Slot = 1
RP State = Active
ISSU State = Init
Operating Mode = Stateful Switchover
Current Image =
bootflash:cat4500e-universal.SSA.03.03.00.SGN1.33.151-2.SGN1.33.bin
Pre-ISSU (Original) Image = N/A
Post-ISSU (Targeted) Image = N/A

Slot = 11
RP State = Standby
ISSU State = Init
Operating Mode = Stateful Switchover
Current Image =
bootflash:cat4500e-universal.SSA.03.03.00.SGN1.33.151-2.SGN1.33.bin
Pre-ISSU (Original) Image = N/A
Post-ISSU (Targeted) Image = N/A

```

Switch# **show switch virtual slot-map**

Virtual Slot to Remote Switch/Physical Slot Mapping Table:

Virtual Slot No	Remote Switch No	Physical Slot No	Module Uptime
1	1	1	00:44:19
2	1	2	00:44:05
3	1	3	00:43:49
4	1	4	-
5	1	5	-
6	1	6	-
7	1	7	-
8	1	8	-
9	1	9	-
10	1	10	-
11	2	1	00:26:40
12	2	2	00:44:48
13	2	3	00:44:48
14	2	4	-
15	2	5	-
16	2	6	-
17	2	7	-
18	2	8	-
19	2	9	-
20	2	10	-

Switch# **dir bootflash:**

Directory of bootflash:/

```

29122 -rw- 119519232 Aug 13 2012 19:13:14 +00:00
cat4500e-universal.SSA.03.03.00.SGN1.34.151-2.SGN1.34.bin
29125 -rw- 119286584 Aug 13 2012 22:30:02 +00:00
cat4500e-universal.SSA.03.03.00.SGN1.33.151-2.SGN1.33.bin

```

820875264 bytes total (581648384 bytes free)

Switch# **dir slavebootflash:**

Directory of slavebootflash:/

```

58372 -rw- 119519232 Aug 14 2012 11:40:47 +00:00
cat4500e-universal.SSA.03.03.00.SGN1.34.151-2.SGN1.34.bin
58370 -rw- 119286584 Aug 14 2012 11:25:38 +00:00
cat4500e-universal.SSA.03.03.00.SGN1.33.151-2.SGN1.33.bin

```

```

822910976 bytes total (583688192 bytes free)
Switch# issu changeversion
bootflash:cat4500e-universal.SSA.03.03.00.SGN1.34.151-2.SGN1.34.bin
%% 'issu changeversion' is now executing 'issu loadversion'
% issu loadversion executed successfully, Standby is being reloaded

%% changeversion finished executing loadversion, waiting for standby to reload and reach
SSO ...

```

Switch 2 goes down, reboots with the post-upgrade image, then reaches SSO Hot Standby state.

...

```

*Aug 14 15:45:45.931: %RF-5-RF_TERMINAL_STATE: Terminal state reached for (SSO)
*Aug 14 15:48:45.958: %INSTALLER-7-ISSU_OP_SUCC:  issu changeversion is now executing
'issu runversion'Please stand by while rebooting the system...

```

A Stateful Switchover occurs. Switch 2 takes over as the Active switch. Switch 1 goes down, then reboots (still with the pre-upgrade image) and reaches SSO Hot Standby state.

(From this point on, the console logs are gathered on Switch 2)

```

*Aug 14 15:54:49.164: %INSTALLER-7-ISSU_OP_SUCC:  issu changeversion is now executing
'issu commitversion'

```

Switch 1 goes down again, then boots up (this time with the post-upgrade image), and comes up as SSO Hot Standby

```

Switch# show switch virtual
Switch mode           : Virtual Switch
Virtual switch domain number : 16
Local switch number   : 2
Local switch operational role: Virtual Switch Active
Peer switch number    : 1
Peer switch operational role : Virtual Switch Standby

Switch# show switch virtual slot-map
Virtual Slot to Remote Switch/Physical Slot Mapping Table:

```

Virtual Slot No	Remote Switch No	Physical Slot No	Module Uptime
1	1	1	00:01:21
2	1	2	00:19:12
3	1	3	00:19:12
4	1	4	-
5	1	5	-
6	1	6	-
7	1	7	-
8	1	8	-
9	1	9	-
10	1	10	-
11	2	1	00:18:43
12	2	2	00:18:17
13	2	3	00:18:16
14	2	4	-
15	2	5	-
16	2	6	-
17	2	7	-
18	2	8	-
19	2	9	-
20	2	10	-

```

Switch#show issu state detail

```

```

        Slot = 11
        RP State = Active
        ISSU State = Init
        Operating Mode = Stateful Switchover
        Current Image =
bootflash:cat4500e-universal.SSA.03.03.00.SGN1.34.151-2.SGN1.34.bin
        Pre-ISSU (Original) Image = N/A
        Post-ISSU (Targeted) Image = N/A

        Slot = 1
        RP State = Standby
        ISSU State = Init
        Operating Mode = Stateful Switchover
        Current Image =
bootflash:cat4500e-universal.SSA.03.03.00.SGN1.34.151-2.SGN1.34.bin
        Pre-ISSU (Original) Image = N/A
        Post-ISSU (Targeted) Image = N/A

```

Switch# **show redundancy states**

```

    my state = 13 -ACTIVE
    peer state = 8 -STANDBY HOT
        Mode = Duplex
        Unit = Primary
        Unit ID = 11

```

```

Redundancy Mode (Operational) = Stateful Switchover
Redundancy Mode (Configured) = Stateful Switchover
    Redundancy State = Stateful Switchover
        Manual Swact = enabled

```

Communications = Up

```

    client count = 74
    client_notification_TMR = 240000 milliseconds
        keep_alive TMR = 9000 milliseconds
        keep_alive count = 0
        keep_alive threshold = 18
        RF debug mask = 0

```

Switch# **show redundancy**

Redundant System Information :

```

-----
    Available system uptime = 4 hours, 16 minutes
    Switchovers system experienced = 3
        Standby failures = 1
        Last switchover reason = active unit removed

```

```

        Hardware Mode = Duplex
    Configured Redundancy Mode = Stateful Switchover
    Operating Redundancy Mode = Stateful Switchover
        Maintenance Mode = Disabled
        Communications = Up

```

Current Processor Information :

```

-----
        Active Location = slot 2/1
        Current Software state = ACTIVE
        Uptime in current state = 21 minutes
        Image Version = Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3
Switch Software (cat4500e-UNIVERSAL-M), Version 03.03.00.SGN1.34 CISCO INTERNAL USE ONLY
UNIVERSAL PRODUCTION K10 IOSD VERSION , synced to END_OF_FLO_ISP
Copyright (c) 1986-2012 by Cisco Systems, Inc.

```



```

Compiled Fri 10-Aug
      BOOT =
bootflash:cat4500e-universal.SSA.03.03.00.SGN1.34.151-2.SGN1.34.bin,12;bootflash:cat4500e-
universal.SSA.03.03.00.SGN1.33.151-2.SGN1.33.bin,12;
      Configuration register = 0x2102

Peer Processor Information :
-----
      Standby Location = slot 1/1
      Current Software state = STANDBY HOT
      Uptime in current state = 1 minute
      Image Version = Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3
Switch Software (cat4500e-UNIVERSAL-M), Version 03.03.00.SGN1.34 CISCO INTERNAL USE ONLY
UNIVERSAL PRODUCTION K10 IOSD VERSION , synced to END_OF_FLO_ISP
Copyright (c) 1986-2012 by Cisco Systems, Inc.
Compiled Fri 10-Au
      BOOT =
bootflash:cat4500e-universal.SSA.03.03.00.SGN1.34.151-2.SGN1.34.bin,12;bootflash:cat4500e-
universal.SSA.03.03.00.SGN1.33.151-2.SGN1.33.bin,12;
      Configuration register = 0x2102

```

The following example shows how to use `issu changeversion` with the "at" command option to schedule an ISSU upgrade procedure to automatically start at the specified time. This example specifies that the ISSU upgrade should be started at 16:30 (24 hour format).

```

Switch> enable
Switch# issu changeversion 1 bootflash:y.bin 11 slavebootflash:y at 16:30
% 'issu changeversion' was executed at [ Aug 12 16:27:43 ].
% The planned ISSU changeversion is to occur in (hh:mm:ss) [ 00:03:00 ] at [ Apr 12
16:30:43 ].
% Current system time: [ Aug 12 16:27:43 ]
% Planned upgrade image: bootflash:y.bin
% To cancel the planned upgrade, please execute 'issu abortversion'

Switch# show issu state detail
      Slot = 1
      RP State = Active
      ISSU State = Init
      Changeversion = TRUE
      Operating Mode = Stateful Switchover
      Current Image = bootflash:x.bin
      Pre-ISSU (Original) Image = N/A
      Post-ISSU (Targeted) Image = N/A

      Slot = 11
      RP State = Standby
      ISSU State = Init
      Changeversion = TRUE
      Operating Mode = Stateful Switchover
      Current Image = bootflash:x.bin
      Pre-ISSU (Original) Image = N/A
      Post-ISSU (Targeted) Image = N/A

```

Aborting a Software Upgrade During ISSU

You can abort the ISSU process at any stage manually (prior to entering the `issu commitversion` command) by entering the `issu abortversion` command. The `issu abortversion` command may also be issued after entering the `issu changeversion` command while the automatic ISSU upgrade is still in progress. The ISSU process also aborts on its own if the software detects a failure.



Note

If you enter the **issu abortversion** command before the standby supervisor engine becomes hot, the traffic might be disrupted.

If you abort the process after you issue the **issu loadversion** command, the standby supervisor engine is reset and reloaded with the original software.

If the process is aborted after you enter either the **issu runversion** or **issu acceptversion** command, then a second switchover is performed to the new standby supervisor engine that is still running the original software version. The supervisor engine that had been running the new software is reset and reloaded with the original software version.



Note

Ensure that the standby supervisor is fully booted *before* issuing the **abortversion** command on an active supervisor engine.

The following task describes how to abort the ISSU process before you complete the ISSU process with the **issu commitversion** command.

Perform the following task on the active supervisor engine:

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	Switch# issu abortversion [<i>active slot</i> [<i>active-image-new</i>]]	Cancels the ISSU upgrade or downgrade process in progress and restores the switch to its state before the process had started.

This example shows how to abort the ISSU process on slot number 11, the slot for the current active supervisor engine. In this example, the ISSU upgrade process is in the Runversion state when the **issu abortversion** command is entered:

```
Switch> enable
Switch# show issu state detail
      Slot = 11
      RP State = Active
      ISSU State = Run Version
      Operating Mode = Stateful Switchover
      Current Image = bootflash:x.bin
      Pre-ISSU (Original) Image = bootflash:y.bin
      Post-ISSU (Targeted) Image = bootflash:x.bin

      Slot = 1
      RP State = Standby
      ISSU State = Run Version
      Operating Mode = Stateful Switchover
      Current Image = bootflash:y.bin
      Pre-ISSU (Original) Image = bootflash:y.bin
      Post-ISSU (Targeted) Image = bootflash:x.bin

Switch# issu abortversion 11
% issu abortversion initiated successfully
Switch# show issu state detail

      Slot = 1
      RP State = Active
```

```

ISSU State = Init
Operating Mode = Stateful Switchover
Current Image = bootflash:y.bin
Pre-ISSU (Original) Image = N/A
Post-ISSU (Targeted) Image = N/A

Slot = 11
RP State = Standby
ISSU State = Init
Operating Mode = Stateful Switchover
Current Image = bootflash:y.bin
Pre-ISSU (Original) Image = N/A
Post-ISSU (Targeted) Image = N/A

```

Switch#

Configuring the Rollback Timer to Safeguard Against Upgrade Issues

Cisco IOS XE software maintains an ISSU rollback timer, to safeguard against an upgrade that may leave the new active supervisor engine in a state in which communication with the standby supervisor engine is severed.

You may want to configure the rollback timer to fewer than 45 minutes (the default) so that you need not wait in case the new software is not committed or the connection to the switch was lost while it was in runversion mode. Conversely, you may want to configure the rollback timer to more than 45 minutes in order to have enough time to verify the operation of the new Cisco IOS XE software before committing the new software image.

The ISSU rollback timer kicks in immediately after **issu run version** is entered so that the minimum value configured should be more than the time required for a chassis reload. Else, the process fails.



Note

The valid timer value range is from 0 to 7200 seconds (two hours). A value of 0 seconds disables the rollback timer.

Once you are satisfied that the new image at the active supervisor engine has been successful and you want to remain in the current state, you may indicate acceptance by issuing the **issu acceptversion** command, which stops the rollback timer.

Issuing the **issu commitversion** command at this stage is equal to entering both the **issu acceptversion** and the **issu commitversion** commands. Use the **issu commitversion** command if you do not intend to run in the current state for a period of time and are satisfied with the new software version.



Note

The rollback timer can be configured only in the ISSU Init state.

This task explains how to configure the rollback timer:

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	Switch# configure terminal	Enters global configuration mode.
Step 3	Switch(config)# issu set rollback-timer value	Configures the rollback timer value, which can range from 0 to 7200.

	Command or Action	Purpose
Step 4	Switch(config)# exit	Returns the user to privileged EXEC mode.
Step 5	Switch# show issu rollback-timer	Displays the current setting of the ISSU rollback timer.

This example shows how to set the rollback timer to 3600 seconds:

```
Switch> enable
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# issu set rollback-timer 3600
% Rollback timer value set to [ 3600 ] seconds

Switch(config)# exit

Switch# show issu rollback-timer
Rollback Process State = Not in progress
Configured Rollback Time = 60:00
```

The Rollback Timer cannot be set in loadversion or runversion state, as the following example illustrates:

```
Switch# show issu state detail
Slot = 1
RP State = Active
ISSU State = Load Version
Operating Mode = Stateful Switchover
Current Image = bootflash:old_image
Pre-ISSU (Original) Image = bootflash:old_image
Post-ISSU (Targeted) Image = bootflash:new_image

Slot = 11
RP State = Standby
ISSU State = Load Version
Operating Mode = Stateful Switchover
Current Image = bootflash:new_image
Pre-ISSU (Original) Image = bootflash:old_image
Post-ISSU (Targeted) Image = bootflash:new_image

Switch# show issu rollback-timer
Rollback Process State = Not in progress
Configured Rollback Time = 60:00

Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# issu set rollback-timer 20
% ISSU state should be [ init ] to set the rollback timer
```

The ISSU Compatibility Matrix

The ISSU Compatibility Matrix contains information about the compatibility of the IOS XE software version currently running on the system, and other versions. The Compatibility Matrix deals with two kinds of information:

- [Stored Information, page 5-83](#)
- [Negotiated Information, page 5-83](#)

Stored Information

The stored compatibility matrix contains a list of other IOS XE software releases for the Catalyst 4500 platform that are compatible with this release. This information is precomputed and stored inside the IOS XE image.

When an ISSU upgrade is attempted, the software looks up the Stored Compatibility Matrix on the supervisor engine that is running the higher (that is, later) IOS XE version. In this matrix, the software tries to locate the IOS XE version number that is running on the other supervisor engine (that is, the lower or earlier version number). If this information is missing, the ISSU upgrade cannot proceed.

All current IOS XE releases on the Catalyst 4500 platform support Dynamic Image Version Capability (DIVC). This means that the ISSU compatibility for the specified version is dynamically computed, as illustrated with the following example:

```
Switch# show issu comp-matrix stored
```

```
Number of Matrices in Table = 1
```

```
(1) Matrix for cat4500e-UNIVERSAL-M(182) - cat4500e-UNIVERSAL-M(182)
=====
Start Flag (0xDEADBABE)
```

My Image ver:	03.03.01.SG
Peer Version	Compatibility
-----	-----
03.02.00.SG	Dynamic(0)
03.02.01.SG	Dynamic(0)
03.02.00.XO	Dynamic(0)
03.02.02.SG	Dynamic(0)
03.02.03.SG	Dynamic(0)
03.02.04.SG	Dynamic(0)
03.03.00.SG	Dynamic(0)
03.03.01.SG	Comp(3)

The above Stored Compatibility Matrix is for IOS XE version 03.03.01.SG.

The "Comp(3)" entry shows that IOS XE version 03.03.01.SG is compatible with this version, and the end result is guaranteed to succeed.

The "Dynamic(0)" entry against IOS XE version 03.02.04.SG means that an ISSU upgrade from or to IOS XE version 03.02.04.SG is permitted. However, the end result will depend upon the ability of individual software features comprising the two versions to successfully complete the ISSU negotiation.

IOS XE version 03.01.01.SG is not in the list. This means that an ISSU upgrade from that version to this IOS XE version (03.03.01.SG) is not possible.

Negotiated Information

While the Stored compatibility matrix information is used before an ISSU upgrade is attempted, the Negotiated compatibility matrix information pertains to the ISSU state after or during an ISSU upgrade attempt. It contains information about how the different software components comprising the IOS XE images on the two supervisor engines were able to negotiate their states. So, this data is useful for troubleshooting failed ISSU upgrade operations.

To display information about the ISSU compatibility matrix, perform this task:

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	Switch# show issu comp-matrix { negotiated xml }	Displays the negotiated ISSU compatibility matrix, either in plain text or in XML form. <ul style="list-style-type: none"> • negotiated—Displays negotiated compatibility matrix information in plain text. • xml—Displays negotiated compatibility matrix information in XML form. <p>Note These commands display only the data within IOSd process.</p> <p>Use the show package compatibility command to display the information for the whole system.</p>
Step 3	Switch# show package compatibility	Displays information regarding all client compatibility in the system.

This example shows how to display negotiated information regarding the compatibility matrix:

```
Switch> enable
Switch# show issu comp-matrix negotiated

CardType: WS-C4503-E(182), Uid: 11, Image Ver: 15.1(2)SGN1.34
Image Name: cat4500e-UNIVERSAL-M

Cid      Eid      Sid      pSid     pUid      Compatibility
=====
2        1        131111   4        1        COMPATIBLE
3        1        65617    7        1        COMPATIBLE
4        1        131085   11       1        COMPATIBLE
5        1        131115   13       1        COMPATIBLE
...
...
7200     1        131105   75       1        COMPATIBLE
7201     1        131151   76       1        COMPATIBLE
7203     1        131127   74       1        COMPATIBLE
7301     1        131137   77       1        COMPATIBLE

Message group summary:
Cid      Eid      GrpId     Sid      pSid     pUid      Nego Result
=====
2        1        1         131111   4        1        Y
3        1        1         65617    7        1        Y
4        1        1         131085   11       1        Y
5        1        1         131115   13       1        Y
...
...
7200     1        1         131105   75       1        Y
7201     1        1         131151   76       1        Y
7203     1        1         131127   74       1        Y
7301     1        1         131137   77       1        Y

List of Clients:
Cid      Client Name                      Base/Non-Base
=====
2        ISSU Proto client                Base
```

```

3          ISSU RF                      Base
4          ISSU CF client                Base
5          ISSU Network RF client       Base
...
...
7200       ISSU Archive Client          Non-Base
7201       ISSU Rollback Client         Non-Base
7203       ISSU Shell Client            Non-Base
7301       ISSU ANCP Client             Non-Base

Switch#

```

This example shows how to display negotiated information regarding non-IOSd clients:

```

Switch# show package compatibility

```

PackageName	PeerPackageName	ModuleName	Compatibility
rp_base	rp_base	aaa	COMPATIBLE
rp_base	rp_base	aaacommon	COMPATIBLE
rp_base	rp_base	access_policy	COMPATIBLE
rp_base	rp_base	app_sess	COMPATIBLE
rp_base	rp_base	app_sess_ios	COMPATIBLE
rp_base	rp_base	auth_mgr	COMPATIBLE

```

.....
.....

```

License Upgrade on a VSS

When a current license is about to expire, or a new license is to be installed, perform the following task to update the license on a VSS:

- Step 1** Ensure that the new license is installed on both VSS active (Switch 1) and VSS standby (Switch 2). If your device supports using Right-to-use (RTU) licensing, use the following command to activate the new license on both switches:
Switch# **license right-to-use activate** [add-on {dna-advantage | dna-essentials } { evaluation | subscription} | entservices | internal_service | ipbase | lanbase] [accepteula]
- Step 2** Run the Switch# **write memory** command to save the configuration.
- Step 3** On Switch 2, the VSS standby, shutdown all the non-VSL ports.



Caution

Shutting down the VSL ports on the VSS active detaches the standby, which might transition to the VSS active. If not, reload the VSS standby and allow it to boot as the VSS active. A VSS standby booting as the active does not pose a network problem because all non-VSL ports are shutdown.

- Step 4** On Switch 1, shut down the VSL ports. The VSS standby (Switch 2) becomes the VSS active switch.
- Step 5** Reload Switch 2.
- Step 6** While Switch 2 boots up with the updated license, run the **no shut** command on the VSL ports on Switch 1 and reload Switch1 immediately. Switch 2 boots as the VSS standby.
- Step 7** When Switch 2 achieves SSO-HOT, run the **no shut** command on all ports.
- Step 8** Failover the chassis to bring the VSS pair license to the correct level.

License installation and subsequent VSS formation are now complete.



Programmability

Programmability is supported only on Cisco Catalyst 4500-X Series Switches and Cisco Catalyst 4500E Series Switches with Supervisor Engine 9-E, 8L-E, and 8-E. The feature is supported on all available license levels for these switches. This chapter describes how to set-up and configure the feature. It includes the following major sections:

- [About Programmability, page 6-1](#)
- [Configuring Programmability Components, page 6-4](#)
- [Using NETCONF and RESTCONF Protocols, page 6-14](#)
- [Using ODM Models, page 6-15](#)
- [Monitoring Programmability, page 6-23](#)
- [Troubleshooting Programmability, page 6-25](#)
- [Sample Configuration and Reference Information, page 6-28](#)

About Programmability

- [Overview, page 6-1](#)
- [Programmability Components, page 6-2](#)
- [Protocols and Data Models for Programmatic Device Configuration, page 6-2](#)
- [Default Configuration, page 6-3](#)

Overview

Programmability is the capability to configure and manage networking devices using protocols that are specifically designed to be consumed by software, that is, machine to machine interfaces.

The traditional way of configuring and managing Cisco networking devices, has been manual configuration, through the command line interface (CLI). As deployments become more complex, programmability of devices has enabled a shift from manual to automatic network provisioning and configuration.

Managing device configuration programmatically enables you to:

- **Configure and control at scale**—You can automate network configuration while also overcoming difficulties posed by multiple platforms, multiple operating systems, and multiple vendor devices in your network.

- Check to make sure that dependencies are satisfied before committing a change; and also easily roll-back changes when they are not consistently compatible across the network.

To address configuration and monitoring issues, the Internet Engineering Task Force (IETF) has defined these standards in network management:

- Yet Another Next Generation (YANG) data modeling—[RFC 6020](#).
- Network Configuration Protocol (NETCONF)—[RFC 6241](#)
- Representational State Transfer Configuration Protocol (RESTCONF)—uses the same data models as defined for NETCONF using YANG (<https://tools.ietf.org/html/draft-ietf-netconf-restconf-04>).

On Catalyst 4500 Series Switches, the Programmability feature provides the use of NETCONF and RESTCONF interfaces. They reside in a container on the switch and provide interfaces that enable remote management. The YANG data models available with these interfaces determine the scope of functions or actions that can be performed. See [Figure 6-1](#).

Programmability Components

This section describes the components involved in the setup of the feature. See the [Configuring Programmability Components, page 6-4](#) for information about how to configure individual components.

- The Virtual Services Container—Also referred to as a virtual machine (VM), virtual service, or container, is a virtual environment on the switch.

You can install an application within a virtual services container. The application then runs in the virtual services container of the operating system of a switch. The application is delivered as an open virtual application (OVA), which is a tar file with a .ova extension. The OVA package is installed and enabled on the switch through the device CLI.

- The Data Model Interface (DMI)—A container solution that provides the NETCONF and RESTCONF programmable interfaces. You must install and activate this container on the switch. After you activate it, the YANG models and APIs are available for use.
- The Network Bootloader—Network boot loaders support booting from a network-based source.

On the Catalyst 4500 series switches, the Preboot Execution Environment (PXE) feature, also called PXE boot, enables the switch to retrieve the software image, configuration files, scripts, and ova files from a remote server, without end-user intervention, that is, Zero-Touch Provisioning. The remote server can be an HTTP or a TFTP server.

PXE boot requires the configuration of a DHCP server, and the boot field set to one of the autoboot options in the ROMMON.

Throughout the document, PXE boot is used to refer to the method of booting from a network-based source.

Protocols and Data Models for Programmatic Device Configuration

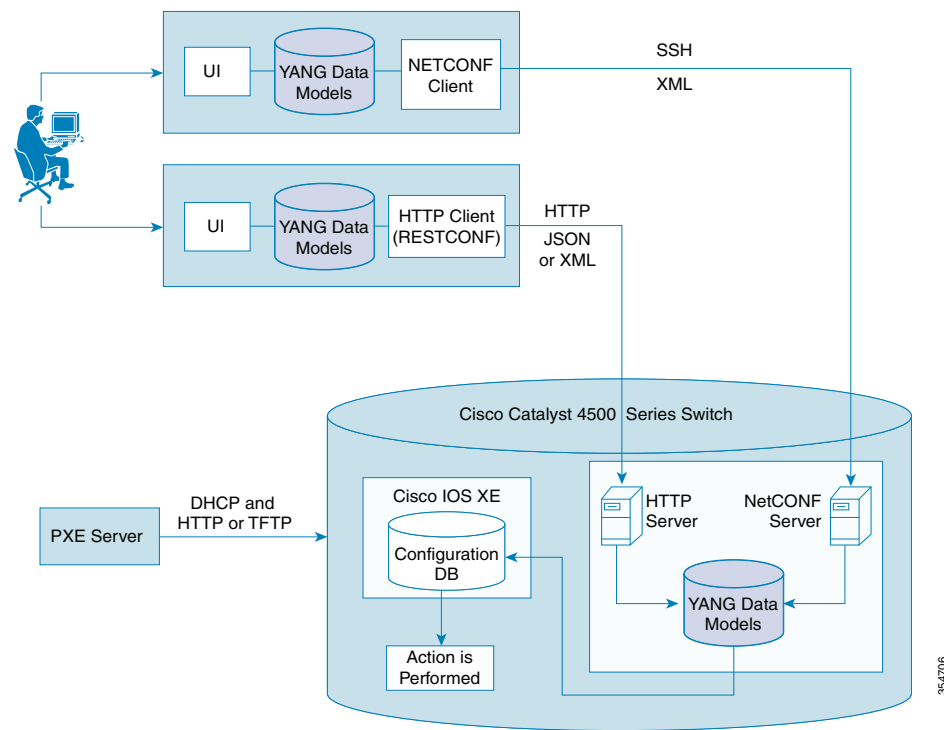
This section describes the protocols and modeling languages that enable a programmatic way of writing configurations to a network device.

- NETCONF—An XML-based protocol that you can use to request information from and make configuration changes to the switch. NETCONF Application Programming Interfaces (APIs) use Secure Shell Version 2 (SSHv2).

- RESTCONF— Uses structured data (XML or JSON) and YANG to provide a REST-like APIs, enabling you to programmatically access different network devices. RESTCONF APIs use HTTPs methods.
- YANG—A data modeling language that is used to model configuration and operational features on the switch. YANG determines the scope and the kind of functions that can be performed by NETCONF and RESTCONF APIs. The following data models are supported:
 - The **ned.yang** model—A Cisco-specific configuration data model that enables to you perform write (SET) operations.
 - The Operational Data Manager (ODM)—Enables you to read operational state data (GET operations) using YANG models.

Figure 6-1 shows how the different components of Programmability come together.

Figure 6-1 Programmability Components



Default Configuration

Programmability is not enabled.

Configuring Programmability Components

You can use Zero-Touch Provisioning to configure the programmability components or follow the standard configuration method (by configuring all required tasks individually).

The following applies to both methods of configuration:

- [Prerequisites for Configuring Programmability, page 6-4](#)
- [Restrictions and Limitations for Configuring Programmability, page 6-5](#)

For zero touch provisioning, you must ensure that you have met:

- [Zero-Touch Provisioning Requirements, page 6-5](#)

For the standard configuration method, you must complete the following:

- [Installing and Activating the DMI Container, page 6-9](#)
- [Configuring One Platform Kit \(OnePK\), page 6-10](#)
- [Providing Privilege Access to Use NETCONF and RESTCONF, page 6-11](#)
- [Enabling the NETCONF Interface, page 6-12](#)
- [Enabling Cisco IOS HTTP Services for RESTCONF, page 6-13](#)

Prerequisites for Configuring Programmability

- Prerequisites for NETCONF and RESTCONF:

Your access to the switch is configured with privilege level 15. This is required to start working with NETCONF and RESTCONF interfaces. See [Providing Privilege Access to Use NETCONF and RESTCONF, page 6-11](#).

- To use the programmability feature, you must use the Universal Crypto Image. See section “Orderable Product Numbers” in the corresponding release notes:

- [Release Notes for the Catalyst 4500-X Series Switches](#)
- [Release Notes for the Catalyst 4500-E Series](#)

- Prerequisites for using PXE boot:

- The boot capability is set to autoboot and the bootfield is set to 04, 05, or 06. PXE boot is supported only if you have enabled autoboot.

**Note**

For PXE boot, the boot capability is set to autoboot by default.

- The required ROMMON version is installed:

On Catalyst 4500-X Series Switches, ROMMON version 15.0(1r)SG14 applies.

On Catalyst 4500-E Series Switches, ROMMON version 15.1(1r)SG8 applies.

With the above ROMMON versions, the factory default setting for the configuration register value is 0x2106 (boot field 06). This is also the recommended setting if you are using an existing device and upgrading to these ROMMON versions.

**Note**

If you are not using PXE boot, you do not have to upgrade the ROMMON version.

Restrictions and Limitations for Configuring Programmability

- Data model related restrictions:
 - Only a subset of the IETF, or common data models are supported.
 - Only the Cisco device-specific **ned.yang** model is supported.
 - When using ODM models, you must stop and restart the ODM control process if the crypto keys are regenerated. See [Activating and Deactivating the ODM, page 6-17](#)
- DMI solution related restrictions:
 - IPv6 is not supported.
 - Switches operating in the VSS mode are not supported.
 - It is not ISSU-capable.
- Only up to 4 simultaneous NETCONF sessions are supported. Further, a session that is idle for more than 180 seconds will timeout.
- Requests coming on an EtherChannel that is part of a Layer 3 Switched Virtual Interface (SVI), and is sharing its IP with the DMI container, is not supported
- AAA remote authentication is not supported.

Zero-Touch Provisioning Requirements

Zero-Touch Provisioning is achieved by using the PXE boot feature. Ensure that you have completed the following:

- Set the boot field value. See [Boot Field, page 6-6](#)
- Configured the DHCP server and an HTTP or TFTP server. See [PXE Boot Requirements—Configuring the DHCP Server, page 6-6](#) and [PXE Boot Process Flow, page 6-7](#)
- Entered the following global configuration commands in the start-up configuration file. This refers to the *<filename>.config* file and is downloaded during the PXE boot process. This is required if you want NETCONF and RESTCONF to be available for use from Day 0.
 - The **virtual-service DMI** command (The virtual service name must be DMI if one opts for Zero-Touch Provisioning).
 - The **activate** command
 - The **ip shared host-interface** *interface-id* command
 - The **onep** command
 - The **service set vty** command
 - The **username** *name* **privilege** level **password** *password* command

To use NETCONF

- The **netconf-yang** command

To use RESTCONF

- The **restconf** command
- The **ip http server** or the **ip http secure-server** command
- The **ip http authentication local** command

The following is a sample output of the **show running-config** command. It displays the commands that are configured as part of start-up configuration

```
Switch #show running-config

Building configuration...

<output truncated>
!
username dmi_admin privilege 15 password 0 dmi_admin
<output truncated>
!
interface GigabitEthernet3/47
no switchport
ip address 198.51.100.20 255.255.255.0
!

<output truncated>

username admin privilege 15 password 7 070E25414707
line vty 0 4
login local
transport input all
ip http authentication local
ip http secure-server
oncp
service set vty
restconf
netconf-yang
!
virtual-service DMI
activate
ip shared host-interface GigabitEthernet6/1
```

Boot Field

To use PXE boot, you must enable autoboot, and set the boot field 04, 05, or 06. This automatically sets the corresponding configuration register value.

The PXE boot feature requires ROMMON version 15.0(1r)SG14 on Catalyst 4500-X Series Switches, and ROMMON version 15.1(1r)SG8 on Catalyst 4500-E Series Switches.

For detailed information about the various boot fields, see table [Explanation of Boot Field \(Configuration Register Bits 00 to 03\)](#) in chapter “Configuring the Switch for the First Time” in this book.

PXE Boot Requirements —Configuring the DHCP Server

To send switch startup configuration files, scripts, and ova file in addition to the bootable image, you must configure the DHCP server.

Depending on your existing DHCP server setup (whether on Microsoft Windows or Linux), ensure that you have made the corresponding, requisite settings. See [DHCP Server Settings on Linux, page 6-28](#) or [Microsoft Windows DHCP Server Configuration, page 6-33](#), whichever applies.



Note

After completing DHCP server configuration, manually assign an IP to the switch and ping to check the switch-to-server connectivity.

Observe the following DHCP server configuration guidelines:

- You must provide the gateway, subnet mask, server IP address, and the client IP address. This information is not permanently stored on switch. They are used only to download files and are deleted when the activity is complete.
- Specifics for the start-up configuration file, script file and ova files (applies to DHCP server configuration for Microsoft Windows and Linux):
 - Complete the DHCP Option 43 list with information about the location of configuration, script and ova file to be downloaded.
 - You can specify an HTTP server or a TFTP server location from which to download. Depending on your requirement, specify one or more options— the boot image name, the start-up configuration file name and path, the script file name and path, and the ova file name and path. Ensure that the configuration, script and ova file extensions are `<config-file>.config,<script-file>.script,<container-file>.ova` respectively.
 - If you are opting for Zero-Touch Provisioning, the ova file name must include `_dmi_`. For example, `example_dmi_container.ova`
 - These files should be saved in the root folder.
 - If you are using HTTP to download, you must configure the DNS server information.
- When the DHCP server responds successfully, the output displays `Received DHCP_ACK`.
- If you receive a TFTP timeout error, increase the DHCP timeout by using the ROMMON variable *DhcpTimeout*. The default DHCP timeout is 5 seconds. You can increase it by a maximum of 30 seconds. For example, if **DhcpTimeout=20**, the DHCP timeout increases by 20 seconds. Enter the set command to verify the change.

```
rommon> DhcpTimeout=20
rommon> set
```

- The DHCP options list that the DHCP server sends should not be greater than 255 bytes. If it is, the following error message is displayed:

```
DHCP ERROR: Received Option length is more than maximum supported (255)
```

- The PXE process ignores network information that you configure on the ROMMON, such as IP, gateway, subnet mask etc.
- You can interrupt the autoboot process at any point, by pressing Control +C (switches to the ROMMON mode).
 - For sample output of the autoboot process, using HTTP, see [Autoboot Process Output—Using HTTP, page 6-37](#).
 - For sample output of the autoboot process using TFTP, see [Autoboot Process Output—Using TFTP, page 6-40](#)

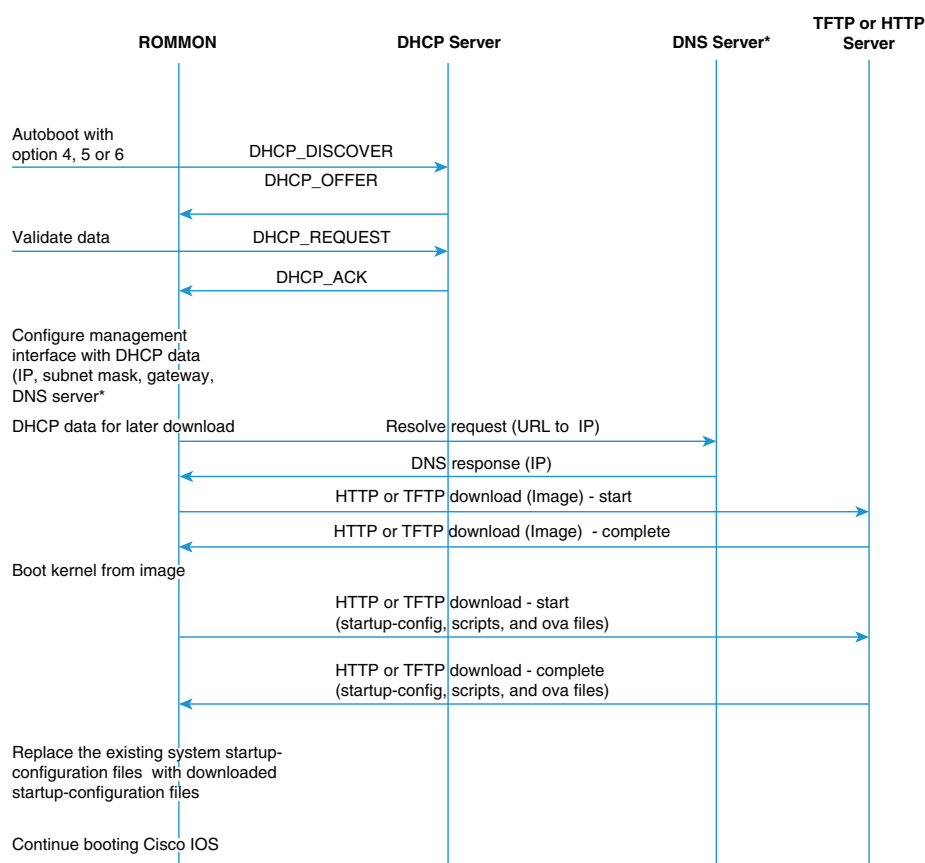
PXE Boot Process Flow

This section outlines the communication process between the DHCP server and the switch and provides the sequence of events followed during the PXE boot (network boot loading) process. This assumes that autoboot is enabled.

1. The switch sends a DHCP discovery packet.
2. The DHCP server responds with an offer containing the TFTP or HTTP server IP address, the offered IP address for the client, the gateway IP address, the boot file name, and the path and names of the OVA, script, and switch configuration files.

3. The switch sends the DHCP request for the IP address.
4. The switch receives the DHCP acknowledgment packet from the server, downloads the image specified in the *filename* variable in the DHCP server, and then boots up with downloaded image.
While booting, the switch receives the Option 43 list from the DHCP server with information about the location of configuration file, script file and ova file to be downloaded.
5. After POST is complete, the switch looks for the startup configuration files, script files, and ova files as mentioned in the Option 43 list received in Step 4. If the files mentioned the Option 43 list are present in the specified location, the switch downloads them.
The script file is downloaded to— bootflash:pxe/scripts folder
The ova file is downloaded to— bootflash:pxe/ova folder.

Figure 6-2 PXE Boot Process Flow



* DNS Server steps do not apply if you are using a TFTP Server

354901

Installing and Activating the DMI Container

This task is mandatory if you have opted for the standard configuration method.

Before you begin, ensure that you have completed the following:

- Downloaded an OVA package that is compatible with the device operating system. The OVA package is available for download in the same location as your system image (.bin) file.
- Ensured that the minimum required disk space - 512 MB, and memory - 256 MB RAM is available on the device for installation and deployment of the DMI container.

To install and activate the DMI by using the virtual services container CLI, perform the following task:

	Command or Action	Purpose
Step 1	enable Example: Switch# enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	virtual-service install name <i>virtual-services-name package file</i> Example: Switch# virtual-service install name dmi package bootflash:/dmi.ova	Installs an OVA package from the specified location onto a device. Ensure that the ova file is located in the root directory of the storage device.
Step 3	configure terminal Example: Switch# configure terminal	Enters the global configuration mode.
Step 4	[no] virtual-service virtual-services-name Example: Switch(config)# virtual-service dmi Switch(config-virt-serv) #	Configures a virtual services container and enters virtual services configuration mode. Observe these guidelines: <ul style="list-style-type: none"> • Use the virtual-services-name defined during installation of the application. • Ensure that installation is complete before proceeding to the next step using the show virtual-service list command.
Step 5	[no] activate Example: Switch(config-virt-serv) # activate	Activates the installed virtual services container.

	Command or Action	Purpose
Step 6	ip shared host-interface <i>interface-id</i> Example: Switch(config-virt-serv)# ip shared host-interface gigabitethernet 3/47	Maps the virtual service container to the interface that you specify. The IP address of the interface you specify here is used for NETCONF and RESTCONF communication. Observe these guidelines: Note You cannot configure a port channel interface as a shared interface. All other interface types are supported. Note If you want to change the shared interface that you have configured, enter the same command with the new interface that you want to use. The no form of this command is not supported.
Step 7	end Example: Switch(config-virt-serv)# exit Switch(config)#	Exits virtual services configuration mode and enters privileged EXEC mode.

Configuring One Platform Kit (OnePK)

OnePK is a software development kit. It enables you to create applications with which to interact directly with Cisco networking devices, and also use a set of controlled API to access networking services.

In the context of Programmability, it is used to enable the VTY service set. The VTY Service allows a onePK application to communicate with a network element, through a virtual terminal.

This task is mandatory if you have opted for the standard configuration method. To enable the requisite, internal OnePK infrastructure, perform the following task:

	Command or Action	Purpose
Step 1	enable Example: Switch> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Switch# configure terminal	Enters the global configuration mode.
Step 3	onep Example: Switch(config)# onep Switch(config-onep)#	Enters the OneP configuration mode.

	Command or Action	Purpose
Step 4	service set vty Example: Switch(config-onep)# service set vty	Enables the VTY service set. The VTY service enables the OneP application to communicate with a network element via a virtual terminal.
Step 5	end Example: Switch# end	Exits the onep configuration mode and enters the privileged EXEC mode.

Providing Privilege Access to Use NETCONF and RESTCONF

This task is mandatory for both zero touch provisioning, and the standard configuration method.

To start working with NETCONF and RESTCONF APIs you must be a user with privilege level 15. To provide this, perform the following task:

	Command or Action	Purpose
Step 1	enable Example: Switch# enable	Enables the privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Switch# configure terminal	Enters the global configuration mode.
Step 3	username name privilege level password password Example: Switch (config)# username example-name privilege 15 password example_password	Establishes a user name-based authentication system. Configure the following keywords: <ul style="list-style-type: none"> privilege level—Sets the privilege level for the user. For the programmability feature, it must be 15. password password—Sets a password to access the CLI view.
Step 4	end Example: Switch# end	Exits global configuration mode and enters privileged EXEC mode.

Enabling the NETCONF Interface

This task is mandatory if you want to use the NETCONF interface and have opted for the standard configuration method:

	Command or Action	Purpose
Step 1	enable Example: Switch# enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Switch# configure terminal	Enters the global configuration mode.
Step 3	netconf-yang Example: Switch(config)# netconf-yang	Enables the NETCONF interface on your network device. After you have completed this step, you can manage network devices through a model based interface. The complete activation of model-based interface processes may require up to 90 seconds.
Step 4	end Example: Switch# end	Exits global configuration mode and enters privileged EXEC mode.

See [Examples for NETCONF RPCs, page 6-14](#).

Enabling Cisco IOS HTTP Services for RESTCONF

This task is mandatory if you want to use the RESTCONF interface and have opted for the standard configuration method.

	Command or Action	Purpose
Step 1	enable Example: Switch# enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Switch# configure terminal	Enters the global configuration mode.
Step 3	restconf Example: Switch(config)# restconf	Enables the RESTCONF interface on your network device.
Step 4	ip http server or ip http secure-server Example: Switch (config)# ip http server OR Switch (config)# ip http secure-server	<p>The ip http server command enables the HTTP server on your system.</p> <p>The ip http secure-server command enables a secure HTTP (HTTPS) server.</p> <p>Note When enabling an HTTPS server, you should always disable the standard HTTP server to prevent unsecured connections to the same services. Disable the standard HTTP server using the no ip http server command in global configuration mode (this step is precautionary; typically, the HTTP server is disabled by default).</p> <p>Configure only one of the commands.</p>
Step 5	ip http authentication local Example: Switch(config)# ip http authentication local	Indicates that the login user name, password and privilege level access combination specified in the local system configuration (with the username global configuration command) should be used for authentication and authorization.
Step 6	end Example: Switch# end	Exits global configuration mode and enters privileged EXEC mode.

See [Examples for RESTCONF RPCs, page 6-15](#).

Using NETCONF and RESTCONF Protocols

NETCONF uses a simple RPC-based (Remote Procedure Call) mechanism to facilitate communication between a client and a server. The client can be a script or an application running as part of a network manager. The server is typically a network device (switch or router).

NETCONF uses Secure Shell Version 2 (SSHv2) as the transport layer across network devices and RESTCONF uses HTTP.

To use NETCONF and RESTCONF you must complete all the required tasks as per the [Configuring Programmability Components, page 6-4](#) section.

NETCONF and RESTCONF also support capability discovery and model downloads. Supported models are discovered using the ietf-netconf-monitoring model. Revision dates for each model are shown in the capabilities response. Data models are available for optional download from a device using the get-schema rpc. You can use these YANG models to understand or export the data model.

The following shows sample RPCs you can send and the kind of action that is performed.

- [Examples for NETCONF RPCs, page 6-14](#)
- [Examples for RESTCONF RPCs, page 6-15](#)

Examples for NETCONF RPCs

Get the running-configuration of the switch by sending the following RPC:

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-config>
    <source>
      <running/>
    </source>
    <filter>
      <native xmlns="http://cisco.com/ns/yang/ned/ios"/>
    </filter>
  </get-config>
</rpc>
```

Change the description of an interface by sending the following RPC

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
      <native xmlns="http://cisco.com/ns/yang/ned/ios">
        <interface>
          <TenGigabitEthernet>
            <name>4/1</name>
            <description>to_distribution</description>
          </TenGigabitEthernet>
        </interface>
      </native>
    </config>
  </edit-config>
</rpc>
```

Remove the description from an interface by sending the following RPC

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
      <native xmlns="http://cisco.com/ns/yang/ned/ios">
        <interface>
          <TenGigabitEthernet>
            <name>4/1</name>
            <description xc:operation="delete"/>
          </TenGigabitEthernet>
        </interface>
      </native>
    </config>
  </edit-config>
</rpc>
```

Examples for RESTCONF RPCs

Get the TFTP source interface by sending the following RPC:

```
GET http://10.106.30.33:80/restconf/api/running/native/ip/tftp/source-interface
```

Configure the TFTP source interface by sending the following RPC:

```
PATCH
http://10.106.30.33:80/restconf/api/running/native/ip/tftp/source-interface/GigabitEthernet
payload = "{\n  \"GigabitEthernet\": \"2/2\"\n}"
```

Enter an HTTP delete request by sending the following RPC:

```
DELETE http://10.106.30.33:55080/api/running/native/ip/tftp/source-interface/
```



Note

For the HTTP delete request do not use:

```
http://10.106.30.33:80/restconf/api/running/native/ip/tftp/source-interface/
```

Using ODM Models

You use ODM models to retrieve read-only operational state data from the system. For this, you must enable Secure Shell Version 2 (SSHv2), activate the ODM, and set the time interval at which the models will collect information from the system.

Each ODM model has a corresponding parser, which polls the specified operational data according to the specified polling interval in milliseconds. See the following sections:

- [Enabling SSHv2, page 6-17](#)
- [Activating and Deactivating the ODM, page 6-17](#)
- [Enabling the Polling Mode, page 6-19](#)
- [Displaying Supported Parsers and Polling Intervals, page 6-20](#)

The following tables lists the parsers, ODM models, and the kind of operational state data that is polled. By default, polling is enabled.

No.	Parsers	ODM Models	Corresponding show Command and Purpose
1	parse.showArchive	cisco-checkpoint-archive.yang	Corresponds to the show archive command, which displays information about the files saved in the Cisco IOS configuration archive.
2	parse.showACL	cisco-acl-oper.yang (confirm if a footnote is required - pratyusha)	Corresponds to the show ip access-lists command, which displays the contents of all current IP access lists.
3	parse.showVirtualService	cisco-virtual-service.yang	Corresponds to the show virtual-service list command, which displays an overview of resources utilized by the applications
4	parse.showProcessesMemory	cisco-process-memory.yang	Corresponds to the show processes memory command, which displays the amount of memory used by each system process in Cisco IOS, Cisco IOS XE, or Cisco IOS Software Modularity images.
5	parse.showProcessesCPU	cisco-process-cpu.yang	Corresponds to the show processes cpu command, which displays CPU utilization to identify the causes of high CPU utilization.
6	parse.showIpRoute	ietf-routing.yang	Corresponds to the show ip route command, which displays the current state of the routing table to verify the configuration.
7	parse.showInterfaces	ietf-interfaces.yang	Corresponds to the show interfaces command, which displays statistics for all interfaces configured on the device or access server.
8	parse.showBFDneighbors	cisco-bfd-state.yang	Corresponds to the show bfd neighbors command, which displays the active BFD neighbor and displays the routing protocols that BFD has registered.
9	parse.showLLDPneighbors	cisco-lldp-state.yang	Corresponds to the show lldp neighbors command, which displays information about neighbors, including device type, interface type and number, holdtime settings, capabilities, and port ID.
10	parse.showMacAddTable	cisco-mac-address-table.yang	Corresponds to the show mac-address-table command, which displays the MAC address table.
11	parse.showPower	cisco-poe-interfaces.yang	Corresponds to the show power inline command, which displays the PoE state for the switch.
12	parse.showModule	cisco-equipment-module.yang	Corresponds to the show module command, which displays module status.
13	parse.showVersion	cisco-cat4k-version.yang	Corresponds to the show version command, which displays hardware and software information for the system.

No.	Parsers	ODM Models	Corresponding show Command and Purpose
14	parse.showInventory	cisco-inventory-entities.yang	Corresponds to the show inventory command, which displays product identification (PID) information for the hardware
15	parse.showIntTransceiver	cisco-interface-transceiver.yang	Corresponds to the show interfaces transceiver detail command, which displays information about the optical transceivers that have digital optical monitoring (DOM) enabled.
16	parse.showIgmpGroup	cisco-igmpsn-group.yang	Corresponds to the show ip igmp snooping groups command, which displays the member port and the IP address.
17	parse.showFlowMonitor	cisco-flow-monitor.yang	Corresponds to the show flow monitor name cache command, which displays the status and statistics for a Flexible NetFlow flow monitor.
18	showIPslaStatistics	cisco-ip-sla-stats.yang	Corresponds to the show ip sla statistics command, which displays the current operational status and statistics of all IP SLAs operations or a specified operation.

Enabling SSHv2

For information about configuring SSHv2, see

http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_usr_ssh/configuration/xe-3e/sec-usr-ssh-xe-3e-book.html

Activating and Deactivating the ODM

This section contains sample RPCs to check the current status of the ODM, to activate it, and to deactivate it.

Example: Checking the Current Status of the ODM.

In the example below, the ODM is active (value set to true).

Input

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get>
    <filter>
      <cisco-ia xmlns="http://cisco.com/yang/cisco-ia">
        <odm-control/>
      </cisco-ia>
    </filter>
  </get>
</rpc>
```

Output

```
NETCONF RETURN
-----
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <data>
```

```

    <cisco-ia xmlns="http://cisco.com/yang/cisco-ia">
      <odm-control>true</odm-control>
    </cisco-ia>
  </data>
</rpc-reply>

```

Example: Activating or Starting the ODM

In the example below, the RPC reply (ok) indicates that the ODM is activated successfully.

Input

```

<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
      <cisco-ia xmlns="http://cisco.com/yang/cisco-ia">
        <odm-control>true</odm-control>
      </cisco-ia>
    </config>
  </edit-config>
</rpc>

```

Output

```

NETCONF RETURN
-----
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <ok/>
</rpc-reply>

```

To deactivate or stop the ODM, send the following RPC:

Input

```

<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
      <cisco-ia xmlns="http://cisco.com/yang/cisco-ia">
        <odm-control>false</odm-control>
      </cisco-ia>
    </config>
  </edit-config>
</rpc>

```

Output

```

NETCONF RETURN
-----
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <ok/>
</rpc-reply>

```

Enabling the Polling Mode

This section contains sample RPCs to check the current polling mode, to enable or change it, and to change the polling interval.

Example: Verifying the Currently Polling Mode of the ODM Models

In the example below, polling is enabled. (In the output section, the polling-enable parameter is set to true).

Input

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get>
    <filter>
      <cisco-odm xmlns="http://cisco.com/yang/cisco-odm">
        <polling-enable/>
      </cisco-odm>
    </filter>
  </get>
</rpc>
```

Output

```
NETCONF RETURN

<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <data>
    <cisco-odm xmlns="http://cisco.com/yang/cisco-odm">
      <polling-enable>true</polling-enable>
    </cisco-odm>
  </data>
</rpc-reply>
```

Example: Enabling or Changing the Polling Mode of the ODM Models

Input

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
      <cisco-odm xmlns="http://cisco.com/yang/cisco-odm">
        <polling-enable>true</polling-enable>
      </cisco-odm>
    </config>
  </edit-config>
</rpc>
```

Output

```
NETCONF RETURN
-----
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <ok/>
</rpc-reply>
```

Example: Changing the Polling Interval of a Parser

In the example below, the polling interval of parser **parse.showArchive** is changed to 110000 milliseconds:

Input

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
      <cisco-odm xmlns="http://cisco.com/yang/cisco-odm">
        <actions>
          <action-name>parse.showArchive</action-name>
          <polling-interval>110000</polling-interval>
        </actions>
      </cisco-odm>
    </config>
  </edit-config>
</rpc>
```

Output

```
NETCONF RETURN
-----
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <ok/>
</rpc-reply>
```

Displaying Supported Parsers and Polling Intervals

To retrieve information about all the supported parsers and their polling intervals, send the following RPC:

Input

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get>
    <filter>
      <cisco-odm xmlns="http://cisco.com/yang/cisco-odm">
        <actions>
          <action-name/>
          <polling-interval/>
          <mode/>
        </actions>
      </cisco-odm>
    </filter>
  </get>
</rpc>
```

Output

```
NETCONF RETURN
-----
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="101">
  <data>
```

```

<cisco-odm xmlns="http://cisco.com/yang/cisco-odm">
  <actions>
    <action-name>parse.showACL</action-name>
    <polling-interval>120000</polling-interval>
    <mode>poll</mode>
  </actions>
  <actions>
    <action-name>parse.showArchive</action-name>
    <polling-interval>120000</polling-interval>
    <mode>poll</mode>
  </actions>
  <actions>
    <action-name>parse.showBFDneighbors</action-name>
    <polling-interval>120000</polling-interval>
    <mode>poll</mode>
  </actions>
  <actions>
    <action-name>parse.showFlowMonitor</action-name>
    <polling-interval>120000</polling-interval>
    <mode>poll</mode>
  </actions>
  <actions>
    <action-name>parse.showIPslaStatistics</action-name>
    <polling-interval>120000</polling-interval>
    <mode>poll</mode>
  </actions>
  <actions>
    <action-name>parse.showIgmpGroup</action-name>
    <polling-interval>120000</polling-interval>
    <mode>poll</mode>
  </actions>
  <actions>
    <action-name>parse.showIntTransceiver</action-name>
    <polling-interval>120000</polling-interval>
    <mode>poll</mode>
  </actions>
  <actions>
    <action-name>parse.showInterfaces</action-name>
    <polling-interval>120000</polling-interval>
    <mode>poll</mode>
  </actions>
  <actions>
    <action-name>parse.showInventory</action-name>
    <polling-interval>120000</polling-interval>
    <mode>poll</mode>
  </actions>
  <actions>
    <action-name>parse.showIpRoute</action-name>
    <polling-interval>120000</polling-interval>
    <mode>poll</mode>
  </actions>
  <actions>
    <action-name>parse.showLLDPneighbors</action-name>
    <polling-interval>120000</polling-interval>
    <mode>poll</mode>
  </actions>
  <actions>
    <action-name>parse.showMacAddTable</action-name>
    <polling-interval>120000</polling-interval>
    <mode>poll</mode>
  </actions>
  <actions>
    <action-name>parse.showModule</action-name>
    <polling-interval>120000</polling-interval>
  </actions>

```

```

        <mode>poll</mode>
    </actions>
    <actions>
        <action-name>parse.showPower</action-name>
        <polling-interval>120000</polling-interval>
        <mode>poll</mode>
    </actions>
    <actions>
        <action-name>parse.showProcessesCPU</action-name>
        <polling-interval>120000</polling-interval>
        <mode>poll</mode>
    </actions>
    <actions>
        <action-name>parse.showProcessesMemory</action-name>
        <polling-interval>120000</polling-interval>
        <mode>poll</mode>
    </actions>
    <actions>
        <action-name>parse.showVersion</action-name>
        <polling-interval>120000</polling-interval>
        <mode>poll</mode>
    </actions>
    <actions>
        <action-name>parse.showVirtualService</action-name>
        <polling-interval>120000</polling-interval>
        <mode>poll</mode>
    </actions>
</cisco-odm>
</data>
</rpc-reply>
.
-----
Ran 1 test in 0.583s

OK

```

Monitoring Programmability

Use these commands in the privileged EXEC mode, to display the Programmability settings you have configured:

Table 6-1 Monitoring Programmability

Show Command	Purpose
debug remotemanagement dmi	<p>Displays the list of applications for which you can debug information. You can choose from the following list of applications</p> <ul style="list-style-type: none"> • ciaauthd—Debug CIA Authorizaition Daemon • ciam—Debug CiaManager • confd—Debug Confd • monit—Debug Monit • nes—Debug Network Element Synchronizer • odm—Debug Operational Data Manager • snmp—Debug SNMP Notification Processor • syncfd—Debug SyncFromDaemon
show onep session all	<p>Displays OneP session information. To verify if NETCONF and RESTCONF interfaces are configured correctly, ensure that these three sessions are listed: NetworkElementSynchronizer, SyncFromDaemon and CiaAuthDaemon. The following is sample output for this command:</p> <p>Switch # show onep session all</p> <pre>ID Username State ReconnectTimer ConnectTime ApplicationName 8145 Connected 0 Thu Jul 28 06:07:05.304 com.cisco.NetworkElementSynchronizer 3234 Connected 0 Thu Jul 28 06:07:06.504 com.cisco.SyncFromDaemon 7249 Connected 0 Thu Jul 28 06:07:07.343 com.cisco.CiaAuthDaemon</pre>
show remotemanagement dmi	<p>Displays the list of applications for which you can display log and status information. You can choose from the following list of applications:</p> <ul style="list-style-type: none"> • ciaauthd—Show CIA Authorization Daemon • ciam—Show CiaManager • confd—Show Confd • genet—Show Mapping Framework • log—Show all DMI logs • monit—Show Monit • nes—Show Network Element Synchronizer • odm—Show Operational Data Manager • snmp—Show SNMP Notification Processor • status—Show status for all DMI applications • syncfd—Show SyncFromDaemon
show virtual-service [global]	Displays available memory, disk space, and CPU allocated for applications.

Table 6-1 Monitoring Programmability

Show Command	Purpose
show virtual-service detail [<i>name virtual-services-name</i>]	Displays a list of resources committed to a specified application, including attached devices.
show virtual-service list	Displays the list of applications installed in the virtual services container. The following is sample output for this command: Switch# show virtual-service list Virtual Service List: Name Status Package Name ----- dmi Activated cat4500e_20160725-212823.ova
show virtual-service storage pool list	Displays an overview of storage locations (pools) used for virtual service containers.
show virtual-service storage volume list	Displays an overview of storage volume information for virtual service containers.
show virtual-service version name <i>virtual-services-name</i> installed	Displays the version of an installed application. For example: Switch# show virtual-service version name dmi installed Virtual service dmi installed version: Name : Netconf-Yang Version : 1.0.0
show virtual-service tech-support	Displays container-based information.
show virtual-service redundancy state	Displays synchronization status
show virtual-service utilization statistics CPU	Displays virtual service CPU utilization statistics.

Troubleshooting Programmability

This section shows sample output for the some of the errors you may encounter while configuring the feature. In some cases a solution is described, and in others, sample configuration output serves as a guideline for correct configuration.

- [File Not Found Errors, page 6-25](#)
- [Startup Configuration Errors, page 6-27](#)
- [Debugging the DMI, page 6-27](#)

File Not Found Errors

If you receive such an error, check the path you have entered for the `filename` field in the DHCP configuration file and make sure that the file exists in your TFTP server. The sample output below shows a successful TFTP session:

```

Filename      : /cat4500e-universalk9.SSA.03.09.00.PR4.46.152-5.0.46.PR4.bin
IP Address    : 192.168.20.16
Loading from TftpServer: 10.106.24.187
TftpBlkSize   : 1468
RxDataPacket  : 130207

Loaded 191143008 bytes successfully.

Checking digital signature....
[/cat4500e-universalk9.SSA.03.09.00.PR4.46.152-5.0.46.PR4.bin]
Digitally Signed Development Software with key version A

Rommon reg: 0x00084F80
Reset2Reg: 0x00004F00

Image load status: 0x00000000
###
Winter 110 controller 0x0468AFAC..0x047F4313 Size:0x002FDB9D
Program Done!
#####
[ 0.058359] pci 0000:00:00.0: ignoring class b20 (doesn't match header type 01)
[ 0.148582] pci 0001:04:00.0: ignoring class b20 (doesn't match header type 01)
[ 0.241172] pci 0002:0c:00.0: ignoring class b20 (doesn't match header type 01)
Starting System Services
devpts /dev/pts devpts rw,nosuid,noexec,relatime,gid=4,mode=600,ptmxmode=000 0 0

diagsk10-post version 5.1.4.1

prod: WS-C4500X-16 part: 73-13860-03 serial: JAE155209ZG

Power-on-self-test for Module 1: WS-C4500X-16

CPU Subsystem Tests ...
seeprom: Pass

Traffic: L3 Loopback ...
Test Results: Pass

Traffic: L2 Loopback ...
Test Results: Pass
post done(56 secs)
Exiting to ios...
Downloading config files from 10.106.24.187 to /bootflash/pxe/user-startup-config

```

```

configs/4500x_start.config
.Received 2201 bytes in 0.0 seconds
Downloading script files from 10.106.24.187 to /bootflash/pxe/scripts
scripts/hello.script
.Received 90 bytes in 0.0 seconds
Downloading ova files from 10.106.24.187 to /bootflash/pxe/ova
container/cat4500e_20160717-183651_33.ova
.....Received 164270080 bytes in 32.0 seconds
Continuing with IOS boot..
Aug 1 06:23:42 %IOSXE-3-PLATFORM: process kernel: [ 124.746012]
mpc85xx_pci_err_probe: Unable to request irq 0 for MPC85xx PCI err
Aug 1 06:23:42 %IOSXE-3-PLATFORM: process kernel: [ 124.756621]
mpc85xx_pcie_err_probe: Unable to request irq 0 for MPC85xx PCIE err
Loading gsbu64atomic as gdb64atomic
Loading pds_helper module
Loading container module
Failed to bring interface "eth1" up
Using 1 for MTS slot
Platform Manager: starting in standalone mode (active)

```

Restricted Rights Legend

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) of the Commercial Computer Software - Restricted Rights clause at FAR sec. 52.227-19 and subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS sec. 252.227-7013.

cisco Systems, Inc.
 170 West Tasman Drive
 San Jose, California 95134-1706

Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3 Switch Software (cat4500e-UNIVERSALK9-M), Version 03.09.00.PR4.46 EARLY DEPLOYMENT [PROD IMAGE] ENGINEERING NOVA_WEEKLY BUILD, synced to V152_5_1_E
 Technical Support: <http://www.cisco.com/techsupport>
 Copyright (c) 1986-2016 by Cisco Systems, Inc.
 Compiled Sun 31-Jul-16 16:31 by sabind

Cisco IOS-XE software, Copyright (c) 2005-2015 by cisco Systems, Inc. All rights reserved. Certain components of Cisco IOS-XE software are licensed under the GNU General Public License ("GPL") Version 2.0. The software code licensed under GPL Version 2.0 is free software that comes with ABSOLUTELY NO WARRANTY. You can redistribute and/or modify such GPL code under the terms of GPL Version 2.0. (<http://www.gnu.org/licenses/gpl-2.0.html>) For more details, see the documentation or "License Notice" file accompanying the IOS-XE software, or the applicable URL provided on the flyer accompanying the IOS-XE software.

This product contains cryptographic features and is subject to United States and local country laws governing import, export, transfer and use. Delivery of Cisco cryptographic products does not imply third-party authority to import, export, distribute or use encryption. Importers, exporters, distributors and users are responsible for compliance with U.S. and local country laws. By using this product you agree to comply with applicable laws and regulations. If you are unable to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at: <http://www.cisco.com/wwl/export/crypto/tool/stqrg.html>

If you require further assistance please contact us by sending email to

```

export@cisco.com.

cisco WS-C4500X-16 (MPC8572) processor (revision 3) with 4194304K bytes of
physical memory.
Processor board ID JAE155209ZG
MPC8572 CPU at 1.5GHz, Cisco Catalyst 4500X
Last reset from Reload
1 Virtual Ethernet interface
16 Ten Gigabit Ethernet interfaces
511K bytes of non-volatile configuration memory.

Press RETURN to get started!
Switch>

```

Startup Configuration Errors

If you encounter errors when you replace the existing startup configuration with a new configuration, the system does not replace the existing startup configuration. You must resolve the errors in the switch start-up configuration file before resuming.

Debugging the DMI

To start debugging the DMI container, perform the following task:

-
- Step 1** Set the logging level to “debug” in cisco-ia.yang model.
 - Step 2** In the privilege EXEC Mode on the switch CLI, enter one of these commands and use NETCONF to read the responses
 - The **debug remotemanagement** command.
 - The **show remotemanagement** command
 - Step 3** To display NETCONF statistical information, such as, the number of sessions, netconf RPCs, packets and so on, use the ietf-netconf-monitoring.yang model.

The following is sample output for the **show remotemanagement command confd log** command:

```

Switch# show remotemanagement dmi confd log
remotemanagement-iosxe-remote-mgmt-8086

***** cia-confd.log *****
<DEBUG> 16-Nov-2016::22:58:44.010 iosxe-remote-mgmt confd[28320]: - Loading file
./mib-fxs/VPN-TC-STD-MIB.fxs
<DEBUG> 16-Nov-2016::22:58:44.018 iosxe-remote-mgmt confd[28320]: - Loading file
./mib-fxs/IANA-ADDRESS-FAMILY-NUMBERS-MIB.fxs
<DEBUG> 16-Nov-2016::22:58:44.019 iosxe-remote-mgmt confd[28320]: - Loading file
./mib-fxs/IANA-RTPROTO-MIB.fxs
<DEBUG> 16-Nov-2016::22:58:44.020 iosxe-remote-mgmt confd[28320]: - Loading file
./mib-fxs/IEEE8021-TC-MIB.fxs

```

Sample Configuration and Reference Information

This chapter provides sample configuration for the following :

- [DHCP Server Settings on Linux, page 6-28](#)
- [Configuring DHCP Option 43 \(for Microsoft Windows\), page 6-32](#)
- [Microsoft Windows DHCP Server Configuration, page 6-33](#)
- [Autoboot Process Output, page 6-37](#)

DHCP Server Settings on Linux

The following is sample configuration that is saved in the *dhcpd.conf* file. Use it as a point of reference when you configure DHCP server settings on Linux.

This sample output covers a scenario where different files are sent to multiple devices of the same vendor specific class, but each one of the devices has a different MAC address.

Comments throughout the sample configuration provide guidelines for important steps (sentence starting with #).



Note

You must restart the DHCP service every time you make a change in the *dhcpd.conf* file.

Depending on whether you are using HTTP or TFTP to download files, refer to the corresponding sample configuration file:

- [Using HTTP, page 6-28](#)
- [Using TFTP, page 6-30](#)

Using HTTP

```
allow booting;
allow bootp;
ddns-update-style none;
#DEFINE AN OPTION SPACE. "EXAMPLE" IS USED HERE. IT IS A VARIABLE YOU CAN SET.
#MAINTAIN code 1,2 AND 3 CONSISTENTLY SINCE THE VALUES CORRESPOND TO CONFIG,SCRIPT AND
#OVA FILES RESEPECTIVELY.
option space EXAMPLE;
option EXAMPLE.startup-config code 1=text;
option EXAMPLE.user-script code 2=text;
option EXAMPLE.user-ova code 3=text;

#ENTER THESE DETAILS AS APPLICABLE TO YOUR NETWORK. TO DOWNLOAD USING HTTP, ENTER THE
#DNS SERVER DETAILS
option domain-name "example-httpserver-url.com";
option domain-name-servers 198.51.100.3;

option subnet-mask 255.255.255.0;
option broadcast-address 192.0.2.255;

#DEFINE A CLASS FOR THE VENDOR-SPECIFIC IDENTIFIER NAME THAT THE DEVICE HAS.
#EXAMPLE:FOR SUP8E/8LE IT IS "WS-X45-SUP8L-E"
#FOR CATALYST 4500-X IT IS "WS-4500X-16"
#ALSO DEFINE THE ROUTER,HTTP SERVER IDENTIFIER,NEXT SERVER IP DETAILS - AS APPLICABLE
#TO YOUR NETWORK
```

```

class "WS-X45-SUP8L-E" {
    match pick-first-value (option dhcp-client-identifier, hardware);
    option routers 198.0.2.254;
    option subnet-mask 255.255.255.0;
    server-identifier 198.51.100.2;
    next-server 198.51.100.2;
}

class "WS-4500X-16" {
    match pick-first-value (option dhcp-client-identifier, hardware);
    option routers 198.0.2.254;
    option subnet-mask 255.255.255.0;
    server-identifier 198.51.100.2;
    next-server 198.51.100.2;
}

#DEFINE A SUBCLASS TO ADD THE DEVICE BASED ON ITS MAC ADDRESS TO RECEIVE
#CONFIGURATION FILES. THIS APPLIES WHEN YOU HAVE MULTIPLE DEVICES WITH SAME
#VENDOR-SPECIFIC IDENTIFIER AND YOU WANT TO PUSH DIFFERENT CONFIGURATIONS BASED ON THE
#MAC ADDRESS

subclass "WS-X45-SUP8L-E" 1:e4:aa:5d:c4:a5:a6 {
# ENTER THE BOOTFILENAME.THIS .BIN IMAGE FILE SHOULD RESIDE IN THE TFTPBOOT FOLDER.
    filename "example-ios-image.bin";
    option routers 198.0.2.254;

#SPECIFY THAT THE OPTION 43 AND ROUTER(3) DETAILS HAVE TO BE SENT TO THE CLIENT SWITCH
    option dhcp-parameter-request-list 43,3;
    option vendor-class-identifier "WS-X45-SUP8L-E";
    vendor-option-space EXAMPLE;

#SPECIFY THE PATH OF THE FILES YOU WANT TO SEND (HTTP).
#MAKE SURE THESE FILES RESIDE IN IDENTICAL FOLDERS (configs/,scripts/,container/) IN
#the HTTP ROOT FOLDER. YOU MUST CREATE THE IDENTICAL FOLDERS WITH THE SAME NAME AND
#CASE.
#ENTER A FILE NAME. MAKE SURE THAT CONFIG, SCRIPT, AND OVA FILE EXTENTIONS ARE
#<config-file>.config,<script-file>.script,<container-file>.ova RESPECTIVELY.

    option EXAMPLE.startup-config
"http://example-httpserver-url.com/configs/example-config.config";
    option EXAMPLE.user-script
"http://example-httpserver-url.com/scripts/example-script.py";
    option EXAMPLE.user-ova
"http://example-httpserver-url.com/container/example_dmi_container.ova";
    option dhcp-parameter-request-list 43,3;
}

subclass "WS-X45-SUP8L-E" 1:e4:aa:5d:c4:a5:a1 {
#WHEN USING HTTP TO DOWNLOAD FILES, PROVIDE THE PATH IN THE FOLLOWING FORMAT:
#filename "http://<http server url>/ios_image.bin"
filename "http://example-httpserver-url.com/example-ios-image.bin"

    option routers 198.0.2.254;
    option dhcp-parameter-request-list 43,3;
    option vendor-class-identifier "WS-X45-SUP8L-E";
    vendor-option-space EXAMPLE;
    option EXAMPLE.startup-config
"http://example-httpserver-url.com/example-config.config";
    option EXAMPLE.user-script
"http://example-httpserver-url.com/example-script.py";
    option EXAMPLE.user-ova
"http://example-httpserver-url.com/example-container.ova";
    option dhcp-parameter-request-list 43,3;
}

```

```

}

subclass "WS-4500X-16" 1:30:e4:db:f8:a4:9f {
    filename "example-ios-image.bin";
    option routers 198.0.2.254;
    option dhcp-parameter-request-list 43,3;
    option vendor-class-identifier "WS-4500X-16";
    vendor-option-space EXAMPLE;
    option EXAMPLE.startup-config
"http://example-httpserver-url.com/example-config.config";
    option EXAMPLE.user-script
"http://example-httpserver-url.com/example-script.py";
    option EXAMPLE.user-ova
"http://example-httpserver-url.com/example-container.ova";
    option dhcp-parameter-request-list 43,3;
}

#ASSIGN A POOL TO GIVE IP ADDRESSES TO THE MEMBERS OF THE VENDOR-SPECIFIC CLASS
subnet 192.0.2.0 netmask 255.255.255.0 {
    pool {
        allow members of "WS-X45-SUP8L-E";
        range 192.0.2.10 192.0.2.50;
    }
    pool {
        allow members of "WS-4500X-16";
        range 192.0.2.51 192.0.2.100;
    }
}

subnet 203.0.113.0 netmask 255.255.255.0 {
    range 203.0.113.12 203.0.113.100;
    option routers 198.51.100.3;
    option subnet-mask 255.255.255.0;
    server-identifier 198.51.100.2;
    next-server 198.51.100.2;
}

```

Using TFTP

```

allow booting;
allow bootp;
ddns-update-style none;
#DEFINE AN OPTION SPACE. "EXAMPLE" IS USED HERE. IT IS A VARIABLE YOU CAN SET.
#MAINTAIN code 1,2 AND 3 CONSISTENTLY SINCE THE VALUES CORRESPOND TO CONFIG,SCRIPT AND
#OVA FILES RESEPTIVELY.
option space EXAMPLE;
option EXAMPLE.startup-config code 1=text;
option EXAMPLE.user-script code 2=text;
option EXAMPLE.user-ova code 3=text;

#ENTER THESE DETAILS AS APPLICABLE TO YOUR NETWORK.

option domain-name "example.com";
option domain-name-servers 198.51.100.3;
option subnet-mask 255.255.255.0;
option broadcast-address 192.0.2.255;

#DEFINE A CLASS FOR THE VENDOR-SPECIFIC IDENTIFIER NAME THAT THE DEVICE HAS.
#EXAMPLE:FOR SUP8E/8LE IT IS "WS-X45-SUP8L-E"
#FOR CATALYST 4500-X IT IS "WS-4500X-16"
#ALSO DEFINE THE ROUTER,TFTP SERVER IDENTIFIER,NEXT SERVER IP DETAILS - AS APPLICABLE
#TO YOUR NETWORK

```

```

class "WS-X45-SUP8L-E" {
    match pick-first-value (option dhcp-client-identifier, hardware);
#THE OPTION ROUTER ADDRESS IS REQUIRED ONLY IF YOU USE A RELAY AGENT BETWEEN THE
#DHCP SERVER AND THE CLIENT.
    option routers 198.0.2.254;
    option subnet-mask 255.255.255.0;
    server-identifier 198.51.100.2;
    next-server 198.51.100.2;
}

class "WS-4500X-16" {
    match pick-first-value (option dhcp-client-identifier, hardware);
    option routers 198.0.2.254;
    option subnet-mask 255.255.255.0;
    server-identifier 198.51.100.2;
    next-server 198.51.100.2;
}

#DEFINE A SUBCLASS TO ADD THE DEVICE BASED ON IT'S MAC ADDRESS TO RECEIVE
#CONFIGURATION FILES. THIS APPLIES WHEN YOU HAVE MULTIPLE DEVICES WITH SAME
#VENDOR-SPECIFIC IDENTIFIER AND YOU WANT TO PUSH DIFFERENT CONFIGURATIONS BASED ON THE
#MAC ADDRESS

subclass "WS-X45-SUP8L-E" 1:e4:aa:5d:c4:a5:a6 {
# ENTER THE BOOTFILENAME.THIS .BIN IMAGE FILE SHOULD RESIDE IN THE TFTP BOOT FOLDER.
    filename "example2-ios-image.bin";
    option routers 198.0.2.254;

#SPECIFY THAT THE OPTION 43 AND ROUTER(3) DETAILS HAVE TO BE SENT TO THE CLIENT SWITCH
    option dhcp-parameter-request-list 43,3;
    option vendor-class-identifier "WS-X45-SUP8L-E";
    vendor-option-space EXAMPLE;

#SPECIFY THE PATH OF THE FILES YOU WANT TO SEND (TFTP).

#MAKE SURE THESE FILES RESIDE IN IDENTICAL FOLDERS (configs/,scripts/,container/) IN
#the TFTP BOOT FOLDER. YOU MUST CREATE THE IDENTICAL FOLDERS WITH THE SAME NAME AND
#CASE.
#ENTER A FILE NAME. MAKE SURE THAT CONFIG, SCRIPT, AND OVA FILE EXTENTIONS ARE
#<config-file>.config,<script-file>.script,<container-file>.ova RESPECTIVELY.

    option EXAMPLE.startup-config
"tftp://198.51.100.2/configs/example2-config.config";
    option EXAMPLE.user-script "tftp://198.51.100.2/scripts/example2-script.py";
    option EXAMPLE.user-ova
"tftp://198.51.100.2/container/example2_dmi_container.ova";
    option dhcp-parameter-request-list 43,3;
}

subclass "WS-X45-SUP8L-E" 1:e4:aa:5d:c4:a5:a1 {
#WHEN USING TFTP TO DOWNLOAD FILES, PROVIDE THE PATH IN THE FOLLOWING FORMAT:
#filename "tftp://<next-server ip address>/<ios_image.bin>";
filename "tftp://198.51.100.2/example2-ios-image.bin"

    option routers 198.0.2.254;
    option dhcp-parameter-request-list 43,3;
    option vendor-class-identifier "WS-X45-SUP8L-E";
    vendor-option-space EXAMPLE;
    option EXAMPLE.startup-config
"tftp://198.51.100.2/configs/example2-config.config";
    option EXAMPLE.user-script "tftp://198.51.100.2/scripts/example2-script.py";
    option EXAMPLE.user-ova
"tftp://198.51.100.2/container/example2_dmi_container.ova";
}

```

```

        option dhcp-parameter-request-list 43,3;
    }

    subclass "WS-4500X-16" 1:30:e4:db:f8:a4:9f {
        filename "tftp://198.51.100.2/example2-ios-image.bin";
        option routers 198.0.2.254;
        option dhcp-parameter-request-list 43,3;
        option vendor-class-identifier "WS-4500X-16";
        vendor-option-space EXAMPLE;
        option EXAMPLE.startup-config
        "tftp://198.51.100.2/configs/example2-config.config";
        option EXAMPLE.user-script "tftp://198.51.100.2/scripts/example2-script.py";
        option EXAMPLE.user-ova
        "tftp://198.51.100.2/container/example2_dmi_container.ova";
        option dhcp-parameter-request-list 43,3;}

#ASSIGN A POOL TO GIVE IP ADDRESSES TO THE MEMBERS OF THE VENDOR-SPECIFIC CLASS
subnet 192.0.2.0 netmask 255.255.255.0 {
    pool {
        allow members of "WS-X45-SUP8L-E";
        range 192.0.2.10 192.0.2.50;
    }
    pool {
        allow members of "WS-4500X-16";
        range 192.0.2.51 192.0.2.100;
    }
}

subnet 203.0.113.0 netmask 255.255.255.0 {
    range 203.0.113.12 203.0.113.100;
    option routers 198.51.100.3;
    option subnet-mask 255.255.255.0;
    server-identifier 198.51.100.2;
    next-server 198.51.100.2;
}

```

Configuring DHCP Option 43 (for Microsoft Windows)

DHCP Option 43 is used by clients and servers to exchange vendor-specific information. (RFC 2132).

This section describes the DHCP Option 43 configuration information that pertains to sending device configuration files, script files, and .ova files to the switch. It is applicable only if you use OpenDhcpServer as the DHCP server, with Microsoft Windows. Other DHCP servers have their own methods to configure this option and the information is available on the Internet.

To send any file, you must convert the file name along with the extension, to a hexadecimal format and the files must be stored in the TFTP root directory.

<File code><length of filename.ext in hexadecimal value><hex value of the filename.ext>

Use the relevant codes to specify the type of file you want to send

- code 01—A configuration file. For example, to send file `example-config.config`:
43=01:15:65:78:61:6d:70:6c:65:2d:63:6f:6e:66:69:67:2e:63:6f:6e:66:69:67:
- code 02—A script file. For example to send file `example-script.py`:
43=02:11:65:78:61:6d:70:6c:65:2d:73:63:72:69:70:74:2e:70:79
- code 03—An ova file. For example, to send file `example_dmi_container.ova`:
43=03:19:65:78:61:6d:70:6c:65:5f:64:6d:69:5f:63:6f:6e:74:61:69:6e:65:72:2e:6f:76:61

Concatenating all three file names


```
43=01:15:65:78:61:6d:70:6c:65:2d:63:6f:6e:66:69:67:2e:63:6f:6e:66:69:67:02:11:65:78:61:
6d:70:6c:65:2d:73:63:72:69:70:74:2e:70:79:03:19:65:78:61:6d:70:6c:65:5f:64:6d:69:5f:63:
6f:6e:74:61:69:6e:65:72:2e:6f:76:61:ff
```

Microsoft Windows DHCP Server Configuration

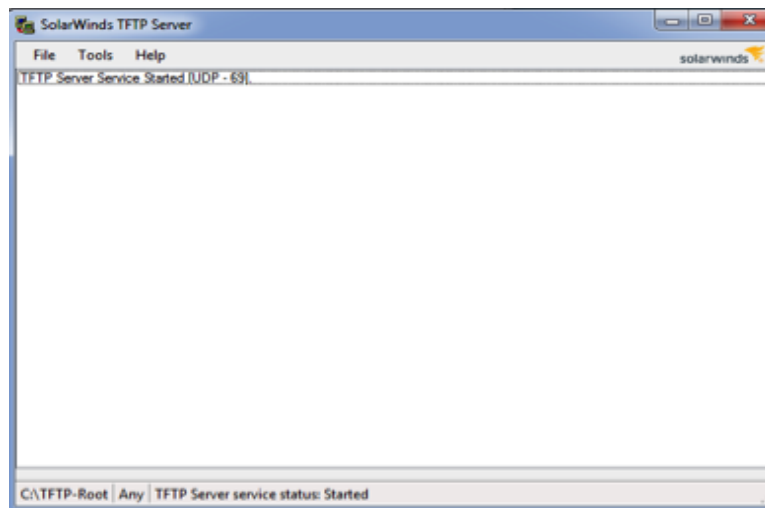
The following example shows how to configure the DHCP Server on Microsoft Windows.



Note

The example uses OpenDhcpServer and Solarwinds TFTP server. Information about configuring both is available on the Internet. The use of both applications here is only meant to serve as an example for configuration, and are not product recommendations.

Figure 6-3 Solarwinds TFTP Server



The important sections of this sample configuration are highlighted **bold**:

```
#This is a configuration file.
#Lines starting with punctuation marks are comments.
#This file should be saved in the same folder as the exe file.
#Remove; and replace the sample value with your own to change a setting

[LISTEN_ON]
#Specify the interfaces that the server should listen to. If you have more than one NIC
#card on your server, always specify the cards that will listen to DHCP/DNS requests.
#Requests from different interfaces look for matching DHCP ranges.
#Requests from relay agents look for a matching range to relay the agent IP.
#You can specify up to 125 interfaces
#By default this includes all static interfaces
;198.51.100.2

[LOGGING]
#You can set the LogLevel as None, Errors or All
#We recommend that you set the logging level to Normal, Normal include errors and DHCP
#renewal messages. The default logging level is Normal.
```

```
;LogLevel=None
;LogLevel=Normal
;LogLevel=All
;LogLevel=Debug
```

[REPLICATION_SERVERS]

```
#You can have 2 instances of Open DHCP Servers in a network. Open DHCP Server sends
#replicated inform messages to the other instance of the Open DHCP server and replicates
#leases. The IP address allotted by one server is not realotted by the other server to
#another host. Further when one server goes down, the other can renew the leases, without
#NAK and DISCOVER. You must specify the primary and secondary servers for replication to
#work.Ensure that the primary & secondary server entries are identical on both servers.
#You may copy the entire ini file on both servers and change the
#LISTEN_ON on individual servers, if needed.
;Primary=192.0.2.253
;Secondary=192.0.2.254
```

[HTTP_INTERFACE]

```
#This is http interface to display the lease status. By default, this is the first
#interface, port 6789. You can change it to any network interface.
;HTTPServer=192.51.100.1:6789
#To limit client access, you can specify up to 8 HTTP client IPs here. If no client IPs
#are specified then clients can access the HTTP interface.
#You can also change the title of the HTML page
;HTTPTitle=example-httpserver-url.com
```

```
#Sections below are other DHCP Sections. You can allot clients addresses dynamically from
#the DHCP Range or statically. For static addresses, client section must be created for
#each static client against its MAC Address. BOOTP clients are always static. DHCP Ranges
#are grouped into [RANGE_SET]s, so that range specific options can be specified for a
#group of ranges in one place. The total ranges in all [RANGE_SET]s is also 125 and you
#can also have a maximum of 125 [RANGE_SET]s.
#You can specify one or more ranges in each [RANGE_SET] section, in the specified format.
#Open DHCP Server allots addresses from these ranges. Static Hosts and BootP clients do
#not require ranges. You do not have to specify a [RANGE_SET] or a DHCP_Range if all
#clients are Static.
```

```
#The dynamic address allocation policy is -
```

```
#1)Look to see if a MAC address is specified as static DHCP Client and use that IP
#2)If not found, look for an old, expired or active address of the same host
#3)If not found, look at the requested IP address and check if available
#4)If not found, allot an unused IP address, if available
#5)If not found, allot the expired IP address of the other host.
#From 2) to 6), requests from different interfaces look for matching DHCP ranges
#of interface IP, and requests from relay agents look for matching range to
#relay agent IP.
```

```
#All the ranges in a [RANGE_SET] section can be further restricted with Filter_Mac_Range,
#Filter_Vender_Class and Filter_User_Class. For example, if a MAC range is specified, then
#this section's ranges will only be available to hosts, whose MAC address falls in this
#range. Also if any host has a matching Filter_Mac_Range in any DHCP_RANGE section, then
#other DHCP range sections without a Filter_Mac_Range or without a matching MAC range will
#not be available to it. Each Manufacturer has a fixed MAC Range. The same Mac ranges can
#repeat in many DHCP_RANGE sections.
#For Filter_Vendor_Class (option 60) and Filter_User_Class filter (option 77),
#the range is available only to a matching value of Filter_Vender_Class
#and Filter_User_Class sent in the client request. If the Filter_Vender_Class and
#the Filter_User_Class values do match in one or more ranges, other ranges with missing
#or non-matching values are not available to such clients. You can specify upto 32
#Filter_Mac_Range, Filter_Vender_Class and Filter_User_Class in each [RANGE_SET].
```

```
#Generally you do not have to specify any filters for the relay agent. The range is
#automatically selected based on the relay agent IP and subnet mask of the range. The
#Relay agent always sends it's subnet side IP. This server uses only the DHCP Range, which
#matches this IP. This ensures that correct range is used. This feature
#eliminate the need of additional configuration. For matching purpose, range is
#recalculated using Subnet Mask of range and Relay Agent IP. However if you want
#to manually configure the subnet selection, you can use FilterSubnetSelection in
#a RANGE_SET. If this filter is specified it will be first matched with SubnetSelection
#Option 118 sent by client. If client sends no such option, it will be matched
#with relay Agent IP. If not relay agent IP is sent, Listening Interface's IP
#will be matched. You can also override the Target Relay Agent using TargetRelayAgent
#option.
```

```
[RANGE_SET]
#This is first and simple DHCP range section example,
#If you need range filters, look at example below
DHCPRange=192.0.2.3-192.0.2.250
VendorClass="Example Server"
43=01:15:65:78:61:6d:70:6c:65:2d:63:6f:6e:66:69:67:2e:63:6f:6e:66:69:67:02:11:65:78:61:6d:
70:6c:65:2d:73:63:72:69:70:74:2e:70:79:03:19:65:78:61:6d:70:6c:65:5f:64:6d:69:5f:63:6f:6e:
74:61:69:6e:65:72:2e:6f:76:61:ff
;43="example-config.config"65:78:61:6d:70:6c:65:2d:63:6f:6e:66:69:67:2e:63:6f:6e:66:69:67;
;"example-script.py"65:78:61:6d:70:6c:65:2d:73:63:72:69:70:74:2e:70:79
;"example_dmi_container.ova"05:78:61:6d:70:6c:65:5f:64:6d:69:5f:63:6f:6e:74:61:69:6e:65:72
;2e:6f:76:61
#The following are the range specific DHCP options.
#You can copy more options names from [GLOBAL_OPTIONS]
SubnetMask=255.255.255.0
DomainServer=198.51.100.3
Router=198.0.2.254
#Lease Time can be different for this Range
;AddressTime=360
```

```
[RANGE_SET]
#This section is also simple [RANGE_SET] section
#Here the options are specified as flat options.
;DHCPRange=192.0.2.3-192.0.2.250
#The following are the flat range specific DHCP options.
#SubnetMask below
;1=255.255.255.0
#DomainServers below
;6=198.51.100.3
#Router
;3=198.0.2.254
#AddressTime
;51=11000
```

```
[RANGE_SET]
#This is filtered [RANGE_SET] section.
#The first eight entries in this example are filters.
#Currently, only the following types of filters are supported
#However 32 filters of each type can be specified
;FilterMacRange=00:0d:60:c5:4e:00-00:0d:60:c5:4e:ff
;FilterMacRange=00:0e:12:c5:4e:00-00:0e:12:c5:4e:ff
;FilterMacRange=00:0f:60:c5:4e:a1-00:0f:60:c5:4e:a1
;FilterVendorClass="EXAMPLE 5.0"
;FilterVendorClass="EXAMPLE 5.1"
;FilterVendorClass="EXAMPLE 5.2"
;FilterUserClass="My User Class 4.0"
;FilterUserClass=123,56,87,123,109,0,23,56,156,209,234,56
;FilterUserClass=00:0d:60:c5:4e:0d:60:c5:4e
#You can select RANGE_SET based on FilterSubnetSelection
```

```

;FilterSubnetSelection=198.51.100.1
;FilterSubnetSelection=192.0.2.1
;Ethernet=no

[GLOBAL_OPTIONS]
#These are global DHCP Options and they supplement client specific options and [RANGE_SET]
#options. Options tags start with 1 and go up to 254, you can specify and option like
#1=255.255.255.0, but it may be difficult to remember option tags. Try using Option names
#instead. If a matching name is not found, you can use tag=value (flat options)
#You can also specify the value as byte array or even hex array. Some options that have
#sub-options can be specified only as hex/byte array. If options have client specific
#values, move/copy them to specific static client sections. If any option has a DHCP range
#specific value, move or copy them to [RANGE_SET] sections.
#You may quote string values (must quote if string contain chars like comma, dot or
#colon). For example NDS_Tree_Name="my.NDS.Tree" or 43="this is return string" or use the
#byte array in value like 43=123,56,87,123,109,0,23,56,156,209,234,56 or use the hex
#array in value 43=a6:87:b6:c9:ae:eb:89:09:a4:67:d5

;DomainName="example-httpserver-url.com"
;SubNetMask=255.255.255.0
;DomainServer=198.51.100.3
;Router=192.0.2.254
#AddressTime is default lease time for server
#specify 0 for infinity lease time
;AddressTime=36000
;RenewalTime=0
;RebindingTime=0
#NextServer is PXEBoot TFTP Server
NextServer=198.51.100.2
;Trailers=yes
;ARPTIMEout=3453
;Ethernet=yes
;DefaultTCPTTL=21
;KeepaliveTime=120
;KeepaliveData=yes
;TFTPServerName=MyTFTPServer
BootFileName=example-ios-image.bin
;AutoConfig=yes
;NameServiceSearch=23,0,235,4,2,0,236,7,94,34,87,4,127,254,23
;SubnetSelectionOption=255.255.255.240
#Option TFTPServerIPAddress is for phone use only, for PXEBoot use NextServer option
;TFTPServerIPAddress=198.51.100.2

#Following sections are Static Client DHCP entries/options
#If no IP is given, then that host will never be allotted any IP
#More option Names can be copied from DHCP-OPTIONS to clients.
#For BOOTP requests, only these options would be sent.
#For DHCP requests. Missing Options will be supplemented from
#first [DHCP-RANGE] options (if IP falls in any range), other
#options will be supplemented from [DHCP-OPTIONS].

[00:41:42:41:42:00]
#This is a client with MAC addr 00:41:42:41:42:00
IP=192.0.2.200
#No other options specified for this client
#For non BOOTP requests, Missing Options will be supplemented from first [RANGE_SET]
#options, if IP falls in any range. and other missing would be added from
[GLOBAL_OPTIONS].

[00:41:42:41:42:05]
#This is a client with MAC addr 00:41:42:41:42:05
IP=192.0.2.201
#DHCP will offer following hostname to this client

```

```
;HostName=TestHost
[00:ff:a4:0e:ef:99]
#This host has no IP
#This host will not get an IP, even from Dynamic Ranges
#You can use such entries to prevent a host from getting an IP from this Server.
```

Autoboot Process Output

[Autoboot Process Output—Using HTTP, page 6-37](#)

[Autoboot Process Output—Using TFTP, page 6-40](#)

Autoboot Process Output—Using HTTP

The following is sample output of the autoboot process on Cisco Catalyst 4500-X Series Switches. .

- The HTTP server from which the files are being downloaded is example-httpserver-url.com.
- The image, configuration, script, and ova files being downloaded are example-ios-image.bin, example-config.config, example-script.py, and example_dmi_container.ova respectively.

```
rommon 1 >reset
|
Resetting .....
|
rommon 2 >
Rommon (G) Signature verification PASSED
Rommon (P) Signature verification PASSED
FPGA (P) Signature verification PASSED
|

*****
*
* Welcome to Rom Monitor for WS-C4500X-16 System.
* Copyright (c) 2008-2014 by Cisco Systems, Inc.
* All rights reserved.
*
*****

Rom Monitor (P) Version 15.0(1r)SG14
CPU Rev: 2.2, Board Rev: 3, Board Type: 108
CPLD Mobat Rev: 2.0x4377.0xb277
Chassis: WS-C4500X-16
|
MAC Address : 30-e4-db-f8-a4-7f
Ip Address : Not set.
Netmask : Not set.
Gateway : Not set.
TftpServer : Not set.
|
Non-Redundant system or peer not running IOS
System Uplinks & Linecards have been reset!!
|
|
***** The system will autoboot in 5 seconds *****
|
|
Type control-C to prevent autobooting.
. . .
Management Ethernet Link Up: 1Gb Full Duplex
```

```

Received DHCP_ACK .
Extending autoboot timeout ...
. . . . .
DHCP Bootfile:http://example-httpserver-url.com/example-ios-image.bin
|
|
HTTP Session Details are ...
|
Filename      : /example-ios-image.bin
IP Address    : 192.0.2.1
HttpServer    : 198.51.100.1
|
|
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!
Loaded 197495364 bytes successfully.
|
Checking digital signature...
[http://example-httpserver-url.com/example-ios-image.bin]
Digitally Signed Development Software with key version A
|
Rommon reg: 0x00084F80
Reset2Reg: 0x00004F00
|
Image load status: 0x00000000
###
Winter 110 controller 0x0468AFAC..0x047F4313 Size:0x002FDB9D
Program Done!
#####
[ 0.091269] pci 0000:00:00.0: ignoring class b20 (doesn't match header type 01)
[ 0.181505] pci 0001:04:00.0: ignoring class b20 (doesn't match header type 01)
[ 0.274669] pci 0002:0c:00.0: ignoring class b20 (doesn't match header type 01)
Starting System Services
devpts /dev/pts devpts rw,nosuid,noexec,relatime,gid=4,mode=600,ptmxmode=000 0 0
|
diagsk10-post version 5.1.4.1
|
prod: WS-C4500X-16 part: 73-13860-03 serial: JAE155209ZD
|
|
Power-on-self-test for Module 1: WS-C4500X-16
|
CPU Subsystem Tests ...
seeprom: Pass
|
Traffic: L3 Loopback ...
Test Results: Pass
|
Traffic: L2 Loopback ...
Test Results: Pass
post done(57 secs)
Exiting to ios...
|
Downloading http://example-httpserver-url.com/example-config.config
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total   Spent    Left     Speed
100  2267  100  2267    0     0   222k      0  --:--:-- --:--:-- --:--:-- 1106k
|
Downloading http://example-httpserver-url.com/example-script.py
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total   Spent    Left     Speed
100  2391  100  2391    0     0   712k      0  --:--:-- --:--:-- --:--:-- 2334k
|

```

```

Downloading http://example-httpserver-url.com/example_dmi_container.ova
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 125M  100 125M    0     0  24.3M      0  0:00:05  0:00:05 --:--:-- 19.1M
|
Continuing with IOS boot..
Nov 21 09:06:06 %IOSXE-3-PLATFORM: process kernel: [ 93.350890] mpc85xx_pci_err_probe:
Unable to request irq 0 for MPC85xx PCI err
Nov 21 09:06:06 %IOSXE-3-PLATFORM: process kernel: [ 93.361062]
mpc85xx_pcie_err_probe: Unable to request irq 0 for MPC85xx PCIE err
Loading gsbu64atomic as gdb64atomic
Loading pds_kc_flowcntl for kstack
loading kstack module
Loading container module
Failed to bring interface "eth1" up
Using 1 for MTS slot
Platform Manager: starting in standalone mode (active)
|
|               Restricted Rights Legend
|
| Use, duplication, or disclosure by the Government is
| subject to restrictions as set forth in subparagraph
| (c) of the Commercial Computer Software - Restricted
| Rights clause at FAR sec. 52.227-19 and subparagraph
| (c) (1) (ii) of the Rights in Technical Data and Computer
| Software clause at DFARS sec. 252.227-7013.
|
|               cisco Systems, Inc.
|               170 West Tasman Drive
|               San Jose, California 95134-1706
|
Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3 Switch Software
(cat4500e-UNIVERSALK9-M), Version 03.09.01.E.179 EARLY DEPLOYMENT [PROD IMAGE]
ENGINEERING NOVA_WEEKLY BUILD, synced to V152_5_1_68_E1
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2016 by Cisco Systems, Inc.
Compiled Sat 12-Nov-16 19:26 by sdcunha
|
Cisco IOS-XE software, Copyright (c) 2005-2015 by cisco Systems, Inc.
All rights reserved. Certain components of Cisco IOS-XE software are
licensed under the GNU General Public License ("GPL") Version 2.0. The
software code licensed under GPL Version 2.0 is free software that comes
with ABSOLUTELY NO WARRANTY. You can redistribute and/or modify such
GPL code under the terms of GPL Version 2.0.
(http://www.gnu.org/licenses/gpl-2.0.html) For more details, see the
documentation or "License Notice" file accompanying the IOS-XE software,
or the applicable URL provided on the flyer accompanying the IOS-XE
software.
|
This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.
|
A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html
|
If you require further assistance please contact us by sending email to
export@cisco.com.
|

```

```

cisco WS-C4500X-16 (MPC8572) processor (revision 3) with 4194304K bytes of physical
memory.
Processor board ID JAE155209ZD
MPC8572 CPU at 1.5GHz, Cisco Catalyst 4500X
Last reset from Reload
1 Virtual Ethernet interface
24 Ten Gigabit Ethernet interfaces
511K bytes of non-volatile configuration memory.
Service config is not supported
|
Press RETURN to get started!
|
Switch>

```

Autoboot Process Output—Using TFTP

The following is sample output of the autoboot process on Cisco Catalyst 4500E Series Switches with Supervisor Engine 8-E.

- The TFTP server from which the files are being downloaded is 198.51.100.2.
- The image, configuration, script and, ova files being downloaded are example2-ios-image.bin, example2-config.config, example2-script.py, and example2_dmi_container.ova respectively.

```

rommon 1 >reset
|
Resetting .....
|
Verifying FPGA (P) Signature ..... PASSED
Verifying ROMMON (P) Signature ..... PASSED
|
*****
*
* Rom Monitor
* Copyright (c) 2012-2015 by cisco Systems, Inc.
* All rights reserved.
*
*****
|
Rom Monitor (P) Version 15.1(1r)SG8
Compiled Wed 26-Oct-16 12:13 [RLS]

System      : WS-X45-SUP8L-E  Slot [3]    Peer [4]
Chassis     : WS-C4507R*E    Mod  [1]
Revision    : CPU 2.1    BOARD 3.0    FPGA 4.3571.7DC7
Memory      : 4096 MB
Date        : Mon Nov 21 09:14:09 2016
|
**** The system will autoboot in 5 seconds ****
|
Type Control-C to prevent autobooting....
Sending DHCP_DISCOVERLink Speed : 1Gb Full Duplex
Received DHCP_ACK .
DHCP state: DHCP_BOUND
|
DHCP Bootfile:tftp://198.51.100.2/example2-ios-image.bin
Link Speed : 1Gb Full Duplex
Filename : /example2-ios-image.bin
IpAddress : 192.0.2.2
TftpServer : 198.51.100.2
TftpBlkSize : 1468

```



```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!*
File Size      : 518480204
MD5            : c5aba4a3946eb54017e49c10e34dacd0
Loaded 518480204 bytes successfully.
|
Checking digital signature....
[mem:/cat4500es8-firmware]
Digitally Signed Development Software with key version A
|
|
|
Rommon reg: 0x00084F80
Reset2Reg: 0x0CB00000
#####
ConanLite controller 0x381D7988..0x38488CC0 Size: 0x0074D07C @
####
Radtrooper controller 0x37AEB588..0x37C87122 Size: 0x00661EDC @
Link: 0x00000080-0x16000000
  Program Done!
|
Checking digital signature....
[mem:/cat4500es8-base]
Digitally Signed Development Software with key version A
|
|
#####
pci 0000:00:00.0: ignoring class b20 (doesn't match header type 01)
pci 0001:02:00.0: ignoring class b20 (doesn't match header type 01)
pci 0002:04:00.0: ignoring class b20 (doesn't match header type 01)
audit: cannot initialize inotify handle
All packages are Digitally Signed
Starting System Services
devpts /dev/pts devpts rw,nosuid,noexec,relatime,gid=4,mode=600,ptmxmode=000 0 0
|
diagsk10-post version 6.2.0.0
|
|
prod: WS-X45-SUP8L-E part: 73-16780-03 serial: CAT1940L26Y
|
|
|
Power-on-self-test for Module 3: WS-X45-SUP8L-E
|
CPU Subsystem Tests ...
  seeprom: Pass
|
Traffic: L3 Loopback ...
  Test Results: Pass
|
Traffic: L2 Loopback ...
  Test Results: Pass
post done(64 secs)
Exiting to ios...
|
Downloading tftp://198.51.100.2/configs/example2-config.config
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
             %                   Dload  Upload  Total  Spent    Left  Speed
100  5848  100  5848    0     0   441k      0  --:--:-- --:--:-- --:--:--   441k
|
Downloading tftp://198.51.100.2/scripts/example2-script.py
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
             %                   Dload  Upload  Total  Spent    Left  Speed
100  2391  100  2391    0     0   603k      0  --:--:-- --:--:-- --:--:--   603k
|

```

```

Downloading tftp://198.51.100.2/container/example2_dmi_container.ova
  % Total    % Received % Xferd  Average Speed   Time    Time     Current
                                 Dload  Upload   Total     Spent    Left     Speed
100 161M 100 161M    0      0 2077k      0  0:01:19  0:01:19 --:--:-- 2077k
|
Continuing with IOS boot..
Loading gsbu64atomic as gdb64atomic
Loading pds_kc_flowcntl for kstack
loading kstack module
Loading container module
Using 3 for MTS slot
Platform Manager: starting in standalone mode (active)
|
|           Restricted Rights Legend
|
| Use, duplication, or disclosure by the Government is
| subject to restrictions as set forth in subparagraph
| (c) of the Commercial Computer Software - Restricted
| Rights clause at FAR sec. 52.227-19 and subparagraph
| (c) (1) (ii) of the Rights in Technical Data and Computer
| Software clause at DFARS sec. 252.227-7013.
|
|           cisco Systems, Inc.
|           170 West Tasman Drive
|           San Jose, California 95134-1706
|
Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3 Switch Software
(cat4500es8-UNIVERSALK9-M), Version 03.09.01.E.179 EARLY DEPLOYMENT [PROD IMAGE]
ENGINEERING NOVA_WEEKLY BUILD, synced to V152_5_1_68_E1
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2016 by Cisco Systems, Inc.
Compiled Sat 12-Nov-16 13:49 by sdcunha
|
Cisco IOS-XE software, Copyright (c) 2005-2015 by cisco Systems, Inc.
All rights reserved. Certain components of Cisco IOS-XE software are
licensed under the GNU General Public License ("GPL") Version 2.0. The
software code licensed under GPL Version 2.0 is free software that comes
with ABSOLUTELY NO WARRANTY. You can redistribute and/or modify such
GPL code under the terms of GPL Version 2.0.
(http://www.gnu.org/licenses/gpl-2.0.html) For more details, see the
documentation or "License Notice" file accompanying the IOS-XE software,
or the applicable URL provided on the flyer accompanying the IOS-XE
software.
|
This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.
|
A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html
|
If you require further assistance please contact us by sending email to
export@cisco.com.
|
cisco WS-C4507R+E (P5040) processor (revision 2) with 4194304K bytes of physical
memory.
Processor board ID FXS1851Q140
P5040 CPU at 1.8GHz, Supervisor 8L-E
Last reset from Reload

```

```
1 Virtual Ethernet interface
48 Gigabit Ethernet interfaces
8 Ten Gigabit Ethernet interfaces
511K bytes of non-volatile configuration memory.
|
|
*****
* WARNING WARNING WARNING !!!!!!! *
* * *
* The ISSU compatibility matrix check has been disabled. *
* No image version compatibility checking will be done. *
* Please be sure this is your intention. *
*****
|
|
Press RETURN to get started!
|
|
Switch>
```




Configuring the Cisco IOS In-Service Software Upgrade Process



Note

Starting with Cisco IOS 12.2(31)SGA, ISSU is supported on the Catalyst 4500. All line cards are supported.

Operating on redundant systems, the In-Service Software Upgrade (ISSU) process allows Cisco IOS software to be updated or otherwise modified while packet forwarding continues. In most networks, planned software upgrades are a significant cause of downtime. ISSU allows Cisco IOS software to be modified while packet forwarding continues. This increases network availability and reduces downtime caused by planned software upgrades. This document provides information about ISSU concepts and describes the steps taken to perform ISSU in a system.

This section includes these topics:

- [Prerequisites to Performing ISSU, page 7-1](#)
- [About ISSU, page 7-2](#)
- [Performing the ISSU Process, page 7-15](#)
- [Related Documents, page 7-42](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

Prerequisites to Performing ISSU

Before performing ISSU, you need to meet these prerequisites:

- Image type of the existing and target image must match. For example, you cannot upgrade from an IP Base image to an Enterprise Services image (and vice versa) without experiencing several minutes of traffic loss.



Note

A similar limitation applies between crypto and non-crypto images.

- The active and the standby supervisor engines must have the same supervisor engine hardware (same model, same memory, NFL daughter card and so on).
- The new and old Cisco IOS software images must be loaded into the file systems (bootflash or compact flash) of both the active and the standby supervisor engines before you begin the ISSU process.

The old image should be available either in bootflash or compact flash and the system should have been booted from one of these locations because the boot variable should not be changed before the ISSU process unfolds.



Note **auto-boot** must be enabled for ISSU to succeed.

- Stateful Switchover (SSO) must be configured and the standby supervisor engine should be in standby hot state.
These commands indicate whether SSO is enabled: **show module**, **show running-config**, **show redundancy state**.
If you do not have SSO enabled, see the *Stateful Switchover* document for further information on how to enable and configure SSO.
- Nonstop Forwarding (NSF) must be configured and working properly. If you do not have NSF enabled, see the *Cisco Nonstop Forwarding* document for further information on how to enable and configure NSF.
- Before you perform ISSU, ensure that the system is configured for redundancy mode SSO and that the file system for both the active and the standby supervisor engines contains the new ISSU-compatible image. The current Cisco IOS version running in the system must also support ISSU.

You can enter various commands on the Catalyst 4500 series switch or the ISSU application on Cisco Feature Navigator to determine supervisor engine versioning and Cisco IOS compatibility.

- If you enter the **no ip routing** command, ISSU falls back from SSO to RPR mode, resulting in traffic loss.
- Autoboot is turned on and the current booted image matches the one specified in the BOOT environmental variable. For details on how to configure and verify these, please refer to "[Modifying the Boot Field and Using the boot Command](#), page 3-28.
- If you enter the **no ip routing** command, ISSU falls back from SSO to RPR mode, resulting in traffic loss.

About ISSU



Note

Do not make any hardware changes while performing ISSU.

Before you perform ISSU, you should understand the following concepts:

- [Stateful Switchover Overview](#), page 7-3
- [NSF Overview](#), page 7-5
- [ISSU Process Overview](#), page 7-6
- [Performing an ISSU Upgrade: 2 Methods](#), page 7-11

- [Changeversion Process, page 7-12](#)
- [Guidelines for Performing ISSU, page 7-13](#)
- [Versioning Capability in Cisco IOS Software to Support ISSU, page 7-13](#)
- [SNMP Support for ISSU, page 7-15](#)
- [Compatibility Verification Using Cisco Feature Navigator, page 7-15](#)

Stateful Switchover Overview

Development of the SSO feature is an incremental step within an overall program to improve the availability of networks constructed with Cisco IOS switches.

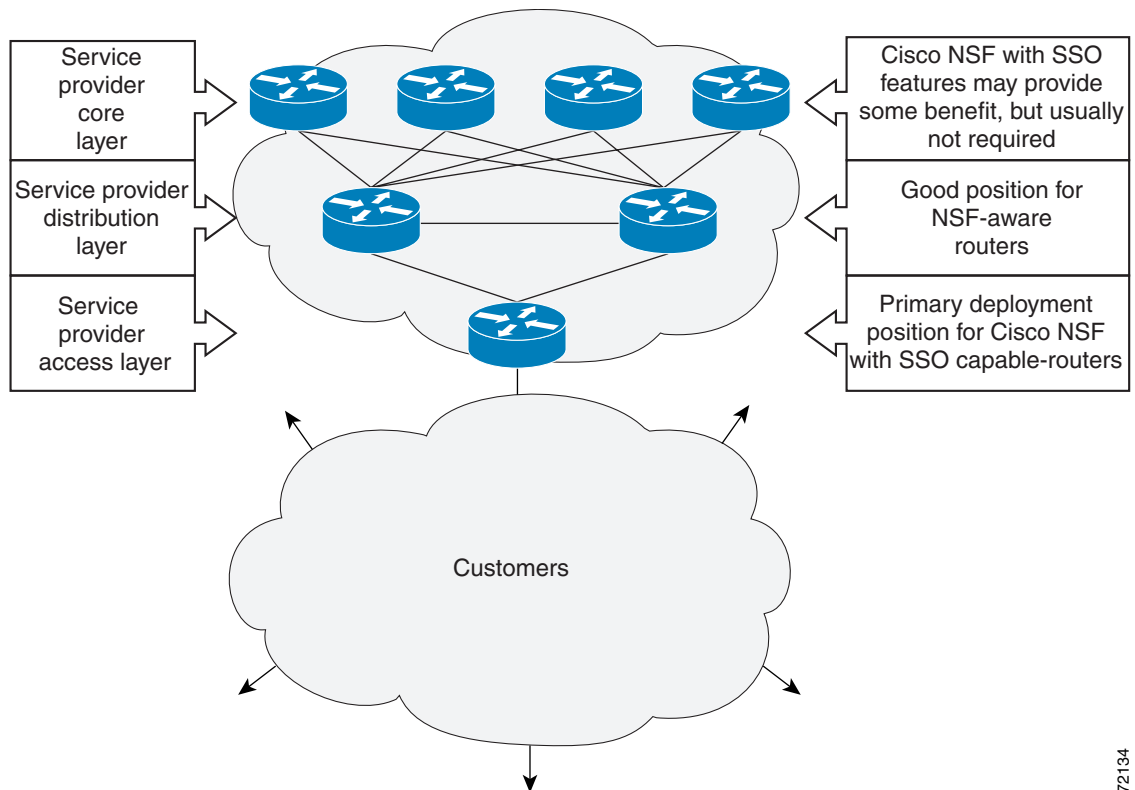
In specific Cisco networking devices that support dual supervisor engines, SSO takes advantage of supervisor engine redundancy to increase network availability. SSO achieves this by establishing one of the supervisor engines as the active processor while the other supervisor engine is designated as the standby processor. Following an initial synchronization between the two supervisor engines, SSO dynamically synchronizes supervisor engine state information between them in real-time.

A switchover from the active to the standby processor occurs when the active supervisor engine fails or is removed from the networking device.

Cisco NSF is used with SSO. Cisco NSF allows the forwarding of data packets to continue along known routes while the routing protocol information is being restored following a switchover. With Cisco NSF, peer networking devices do not experience routing flaps, which reduce loss of service outages for customers.

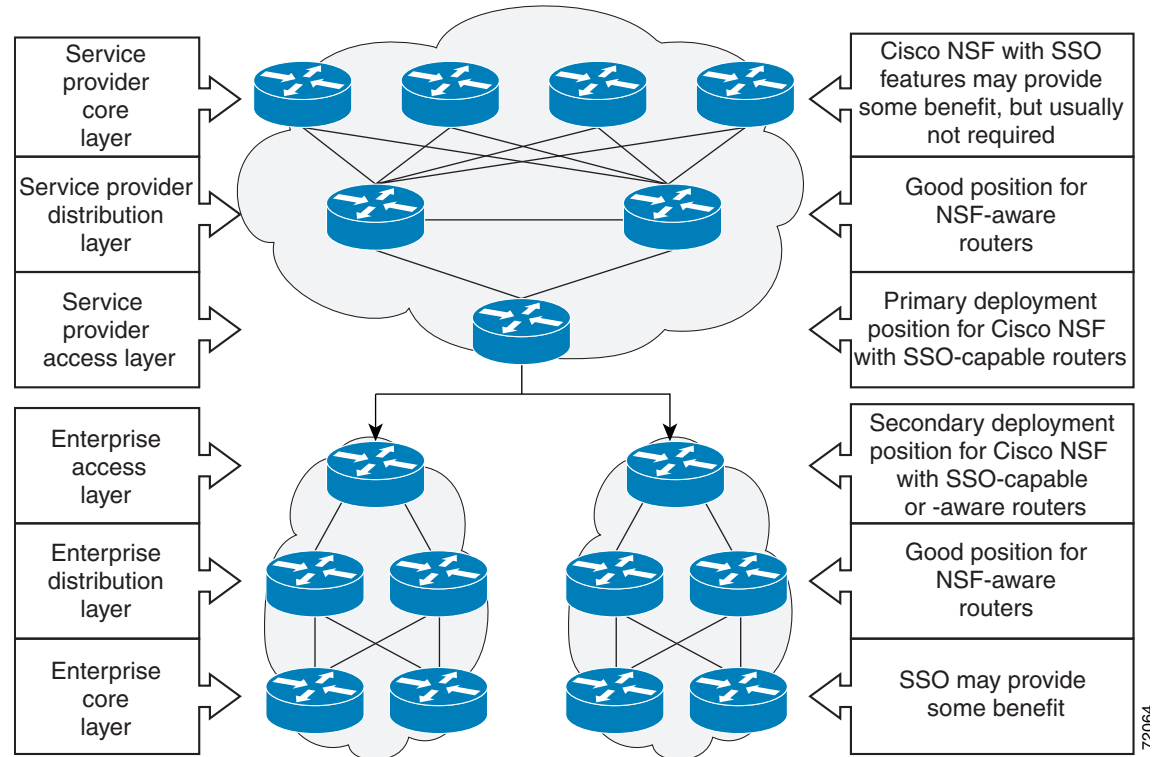
[Figure 7-1](#) illustrates how SSO is typically deployed in service provider networks. In this example, Cisco NSF with SSO is enabled at the access layer (edge) of the service provider network. A fault at this point could result in loss of service for enterprise customers requiring access to the service provider network.

For Cisco NSF protocols that require neighboring devices to participate in Cisco NSF, Cisco NSF-aware software images must be installed on those neighboring distribution layer devices. Depending on your objectives, you may decide to deploy Cisco NSF and SSO features at the core layer of your network. Doing this can help reduce the time to restore network capacity and service for certain failures, which leads to additional availability.

Figure 7-1 Cisco NSF with SSO Network Deployment: Service Provider Networks

72134

Additional levels of availability may be gained by deploying Cisco NSF with SSO at other points in the network where a single point of failure exists. [Figure 7-2](#) illustrates an optional deployment strategy that applies Cisco NSF with SSO at the enterprise network access layer. In this example, each access point in the enterprise network represents another single point of failure in the network design. In the event of a switchover or a planned software upgrade, enterprise customer sessions continue uninterrupted through the network in this example.

Figure 7-2 Cisco NSF with SSO Network Deployment: Enterprise Networks

NSF Overview

Cisco NSF works with the SSO feature in Cisco IOS software. SSO is a prerequisite of Cisco NSF. NSF works with SSO to minimize the amount of time a network is unavailable to its users following a switchover. The main objective of Cisco NSF is to continue forwarding IP packets following a supervisor engine switchover.

Usually, when a networking device restarts, all routing peers of that device detect that the device went down and then came back up. This transition results in what is called a routing flap, which could spread across multiple routing domains. Routing flaps caused by routing restarts create routing instabilities, which are detrimental to the overall network performance. Cisco NSF helps to suppress routing flaps in SSO-enabled devices, thus reducing network instability.

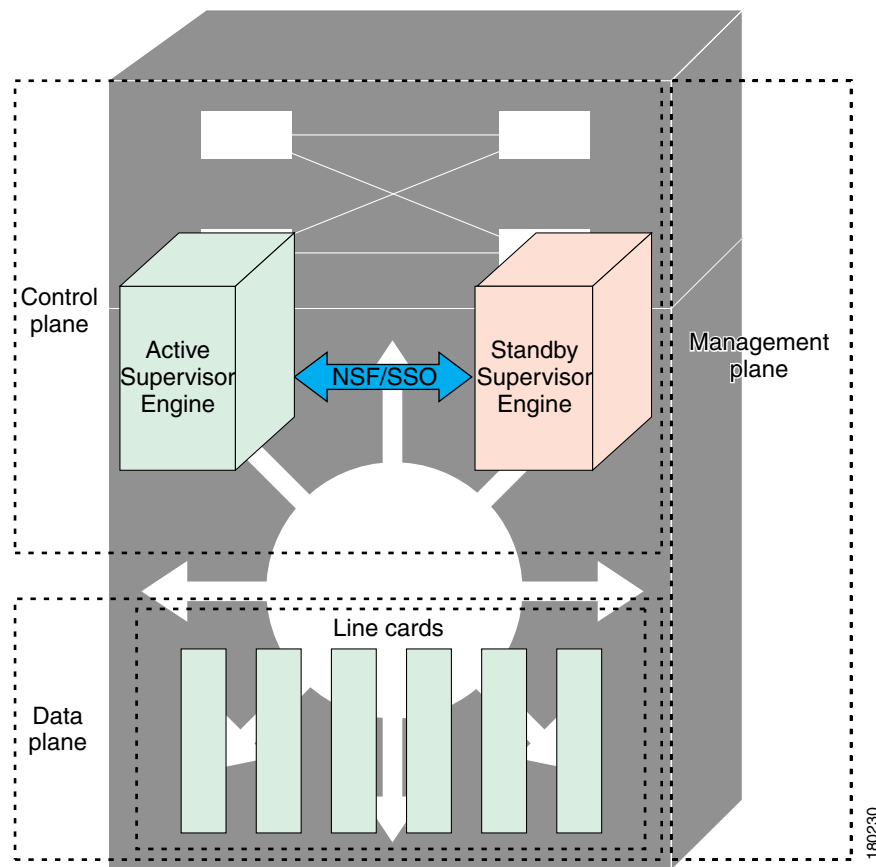
Cisco NSF allows for the forwarding of data packets to continue along known routes while the routing protocol information is being restored following a switchover. With Cisco NSF, peer networking devices do not experience routing flaps. Data traffic is forwarded while the standby supervisor engine assumes control from the failed active supervisor engine during a switchover. The ability of physical links to remain up through a switchover and to be kept current with the Forwarding Information Base (FIB) on the active supervisor engine is key to Cisco NSF operation.

ISSU Process Overview

The ISSU process allows you to perform a Cisco IOS software upgrade or downgrade while the system continues to forward packets. (For an illustration of the commands used during the ISSU process, refer to [Figure 7-8 on page 7-11](#).) Cisco IOS ISSU takes advantage of the Cisco IOS high availability infrastructure—Cisco NSF with SSO and hardware redundancy—and eliminates downtime associated with software upgrades or version changes by allowing changes while the system remains in service (see [Figure 7-3](#)).

SSO and NSF mode support configuration and runtime state synchronization from the active to the standby supervisor engine. For this process to happen, the images on both the active and the standby supervisor engines must be the same. When images on active and standby supervisor engines are different ISSU allows the two supervisor engines to be kept in synchronization even when these two versions of Cisco IOS support different sets of features and commands.

Figure 7-3 High Availability Features and Hardware Redundancy in the ISSU Process

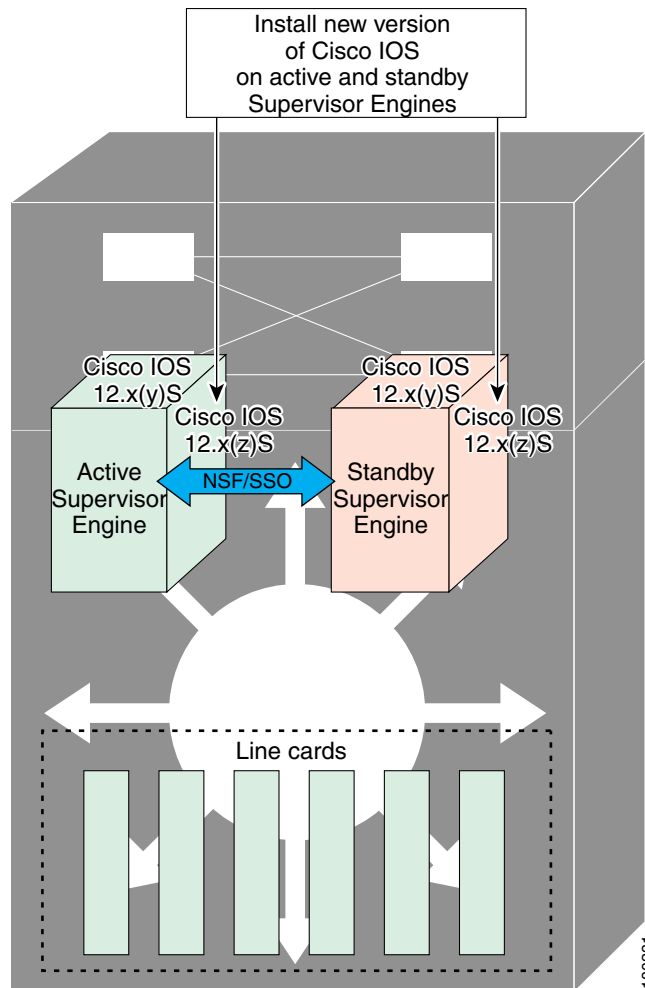


An ISSU-capable switch consists of two supervisor engines (active and standby) and one or more line cards. Before initiating the ISSU process, copy the Cisco IOS software into the file systems of both supervisor engines (see [Figure 7-4](#)).

**Note**

In the following figure, Cisco IOS 12.x(y)S represents the *current* version of Cisco IOS.

Figure 7-4 Install/Copy New Version of Cisco IOS Software on Both Supervisor Engines

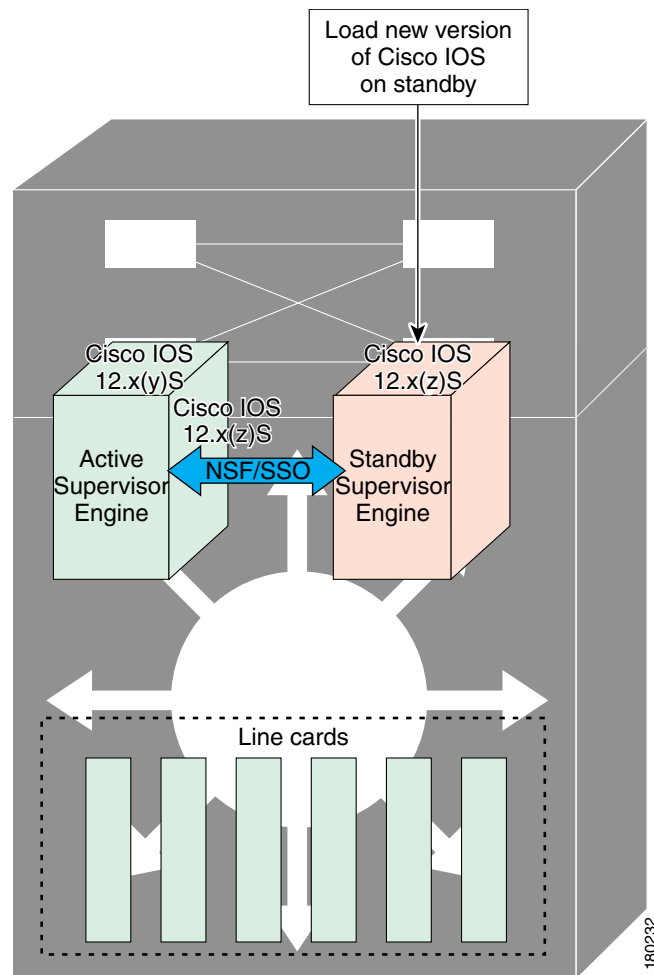


After you have copied the Cisco IOS software to both file systems, load the new version of Cisco IOS software onto the standby supervisor engine (see [Figure 7-5](#)).

**Note**

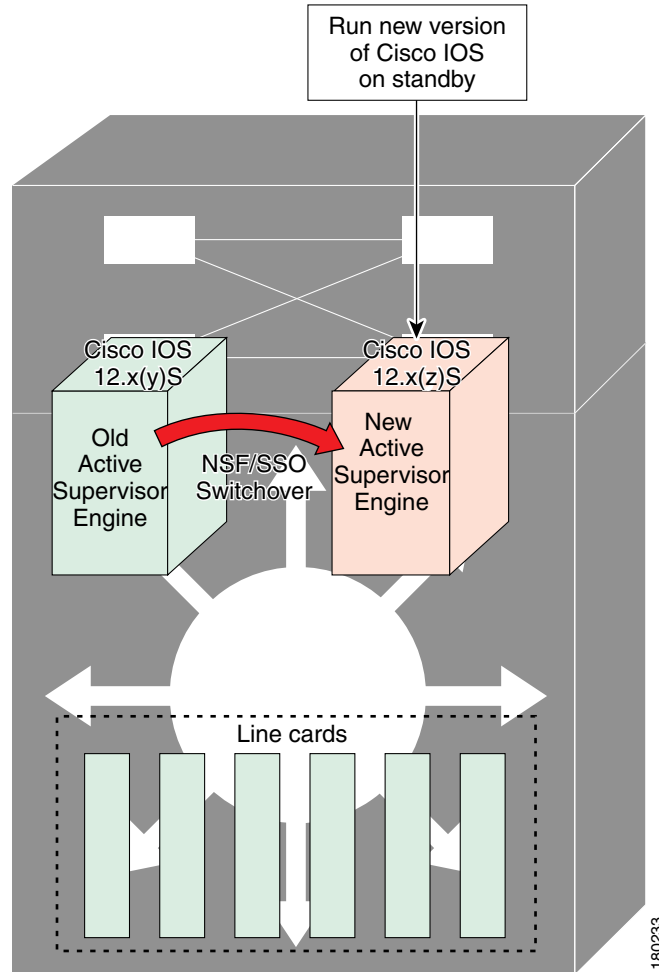
Without the ISSU feature, you cannot have SSO or NSF functioning between the active and standby supervisor engines when they are running two different versions of Cisco IOS image.

Figure 7-5 Load New Version of Cisco IOS Software on the Standby Supervisor Engine



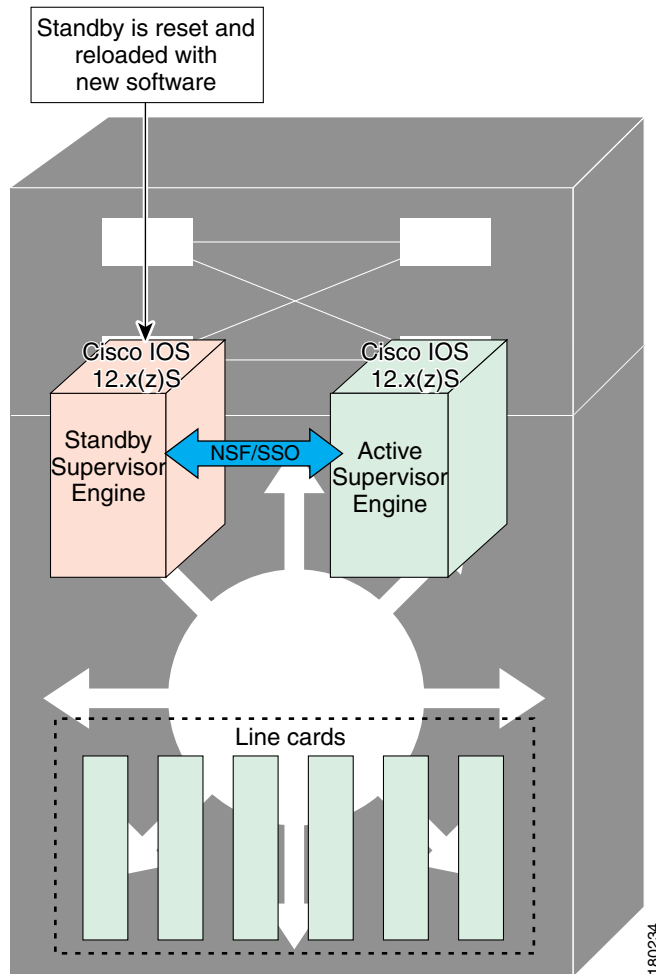
After a switchover (NSF or SSO, not RPR), the standby supervisor engine takes over as the new active supervisor engine (see [Figure 7-6](#)).

Figure 7-6 **Switch Over to Standby Supervisor Engine**



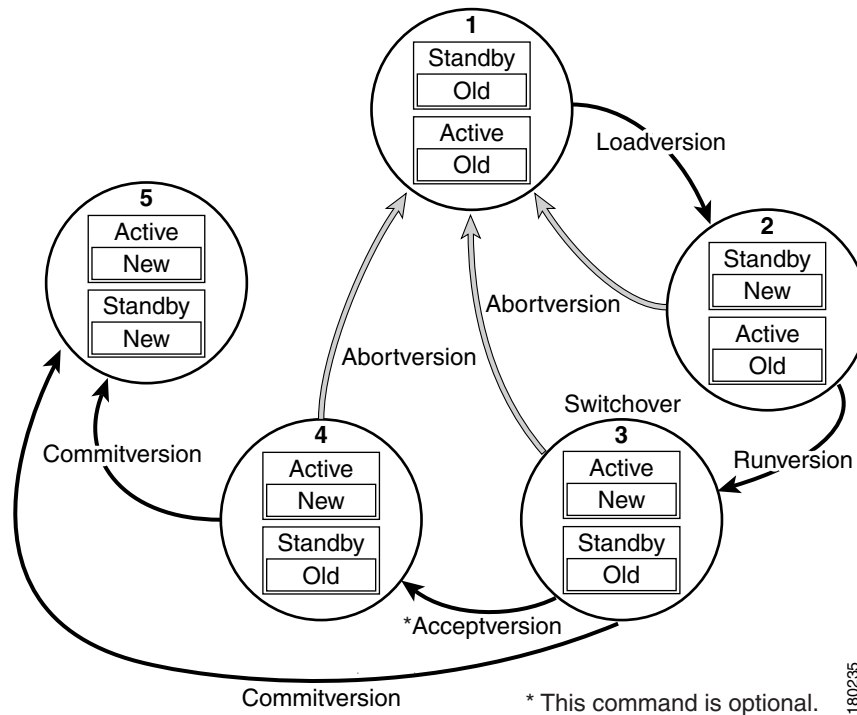
The former active supervisor engine is loaded with an old Cisco IOS image so that if the new active supervisor engine experiences problems, you can abort and conduct a switchover to the former active, which is already running the old image. Next, the former active supervisor engine is loaded with the new version of Cisco IOS software and becomes the new standby supervisor engine (see [Figure 7-7](#)).

Figure 7-7 Load New Standby Supervisor Engine with New Cisco IOS Software



[Figure 7-8](#) shows the steps during the ISSU process.

Figure 7-8 Steps During the ISSU Process



Performing an ISSU Upgrade: 2 Methods

There are two ways to perform an ISSU upgrade: manually, with four commands; or automatically, with one command.

The normal ISSU upgrade process involves issuing four separate ISSU exec commands (**issu loadversion**, **issu runversion**, **issu acceptversion**, **issu commitversion**) along with additional show command invocations to evaluate the success of each command before proceeding. Although the ISSU process is complicated, you should not expect disruption of service. The use of multiple ISSU commands dictates an additional level of care to ensure no service disruption. However, in some scenarios, this upgrade procedure might be cumbersome and of minimal value. A typical example is during a network upgrade that involves performing an ISSU upgrade on a large number of Catalyst 4500 switches. In these cases, we recommend that you first perform the normal (four command) ISSU upgrade procedure on one switch (possibly in a lab environment) to verify successful upgrade. Then, use a single **issu changeversion** command to perform an automatic ISSU on the rest of the Catalyst 4500 switches in the network.



Note

To use the **issu changeversion** command, both old and new IOS versions must support **issu changeversion** functionality.

Changeversion Process

The **issu changeversion** command launches a single-step complete ISSU upgrade cycle. It performs the logic for all four of the standard commands (**issu loadversion**, **issu runversion**, **issu acceptversion**, and **issu commitversion**) without user intervention, streamlining the upgrade through a single CLI step.

Additionally, **issu changeversion** allows the upgrade process to be scheduled for a future time. This enables you to stage a number of systems to perform upgrades sequentially when a potential disruption would be least harmful.

After the standby supervisor engine initializes and the system reaches a terminal state (RPR/SSO), the upgrade process is complete and the BOOT variable is permanently written with the new IOS software software image. Hence, a reset on any RP will keep the system booting the new software image. Console and syslog messages will be generated to notify anyone monitoring the upgrade that the state transition has occurred.

Similar to the normal ISSU upgrade procedure, the in-progress upgrade procedure initiated by the **issu changeversion** command can be aborted with the **issu abortversion** command. If the system detects any problems or detects an unhealthy system during an upgrade, the upgrade might be automatically aborted.

When the **issu runversion** command is entered during the four step manual upgrade process, if any incompatible ISSU clients exist, the upgrade process reports them and their side effects, and allows the user to abort the upgrade. While performing a single-step upgrade process, when the process reaches the runversion state, it will either automatically continue with the upgrade provided the base clients are compatible, or automatically abort because of client incompatibility. If the user wants to continue the upgrade procedure in RPR mode, the user must use the normal ISSU command set and specify the **force** option when entering the **issu loadversion** command.

Changeversion: Quick Option

The **issu changeversion** command provides an optional quick command option that can reduce the time required to perform the automatic ISSU upgrade. When the **quick** command option is applied, the ISSU upgrade state transition differs from that described previously. With this option, the software logic up the loadversion stage remains the same as previously described, and the logic that performs runversion and commitversion is combined. This logic skips the step in the upgrade procedure that loads the old software version on the new standby (old active) supervisor, reducing the time required for the automatic ISSU upgrade by about a third.

Scheduled Changeversion: “in” and “at” Options

issu changeversion provides **in** and **at** command options that enable you to schedule a future automatic ISSU upgrade.

The **at** command option schedules an automatic ISSU upgrade to begin at a specific time. This option specifies an exact time (*hh:mm*, 24 hour format) in the next 24 hours at which the upgrade will occur.

The **in** command option schedules an automatic ISSU upgrade to begin after a certain amount of time has elapsed. This option specifies the number of hours and minutes (*hh:mm* format) that must elapse before an upgrade will occur, with a maximum value of 99:59.

Changeversion Deployment Scenario

The typical **issu changeversion** command usage scenario is for experienced users with a large installed base. These users typically validate a new image using a topology and configuration similar to their production network. The validation process should be done using both the existing multi-command process and the new **issu changeversion** command process. Once users certify an IOS software image and want to roll it out broadly, they can use the single command process to perform an efficient upgrade of their network.

Aborting an In-Progress Changeversion Procedure

The **issu changeversion** command functionality is designed to perform an ISSU software upgrade without user intervention. However, status messages are displayed to the console as the upgrade transitions through the various states. If any anomalies are noticed during the automatic upgrade, perhaps with peers or other parts of the network, you can use the **issu abortversion** command to manually abort the upgrade at any point in the process prior to the commitversion operation.

Guidelines for Performing ISSU

Be aware of the following guidelines while performing the ISSU process:

- Even with ISSU, it is recommended that upgrades be performed during a maintenance window.
- The new features should not be enabled (if they require change of configuration) during the ISSU process.



Note Enabling them will cause the system to enter RPR mode because commands are only supported on the new version.

- In a downgrade scenario, if any feature is not available in the downgrade revision of the Cisco IOS software handle, that feature should be disabled prior to initiating the ISSU process.

Versioning Capability in Cisco IOS Software to Support ISSU

Before the introduction of ISSU, the SSO mode of operation required each supervisor engine to be running the same versions of Cisco IOS software.



Note The operating mode of the system in a redundant HA configuration is determined by exchanging version strings when the standby supervisor engine registers with the active supervisor engine.

The system entered SSO mode only if the versions running on the both supervisor engines were the same. If not, the redundancy mode changes to RPR. With ISSU capability, the implementation allows two different but compatible release levels of Cisco IOS images to interoperate in SSO mode and enables software upgrades while packet forwarding continues. Version checking done before ISSU capability was introduced is no longer sufficient to allow the system to determine the operating mode.

ISSU requires additional information to determine compatibility between software versions. A compatibility matrix is defined, containing information about other images relative to the one in question. This compatibility matrix represents the compatibility of two software versions, one running on the active and the other on the standby supervisor engine, and to allow the system to determine the highest operating mode it can achieve. Incompatible versions cannot progress to SSO operational mode.

Compatibility Matrix

You can perform the ISSU process when the Cisco IOS software on both the active and the standby supervisor engine is capable of ISSU and the old and new images are compatible. The compatibility matrix information stores the compatibility among releases as follows:

- **Compatible**—The base-level system infrastructure and all optional HA-aware subsystems are compatible. An in-service upgrade or downgrade between these versions succeeds with minimal service impact. The matrix entry designates the images to be compatible (C).
- **Base-level compatible**—One or more of the optional HA-aware subsystems is not compatible. An in-service upgrade or downgrade between these versions succeeds; however, some subsystems cannot always maintain state during the transition from the old to the new version of Cisco IOS. The matrix entry designates the images to be base-level compatible (B).

However, you should be able to perform an ISSU upgrade without any functionality loss even if the matrix entry is B. The downgrade may experience some functionality loss if the newer image had additional functionality.

- **Incompatible**—A core set of system infrastructure exists in Cisco IOS that must be able to interoperate in a stateful manner for SSO to function correctly. If any of these required features or subsystems is not interoperable, then the two versions of the Cisco IOS software images are declared to be incompatible. An in-service upgrade or downgrade between these versions is not possible. The matrix entry designates the images to be incompatible (I). The system operates in RPR mode during the period when the versions of Cisco IOS at the active and standby supervisor engines are incompatible.

If you attempt to perform ISSU with a peer that does not support ISSU, the system automatically uses RPR instead.

The compatibility matrix represents the compatibility relationship a Cisco IOS software image has with all of the other Cisco IOS software versions within the designated support window (for example, all of those software versions the image “knows” about) and is populated and released with every image. The matrix stores compatibility information between its own release and prior releases. It is always the newest release that contains the latest information about compatibility with existing releases in the field. The compatibility matrix is available within the Cisco IOS software image and on Cisco.com so that users can determine in advance whether an upgrade can be done using the ISSU process.

To display the compatibility matrix data between two software versions on a given system, enter the **show issu comp-matrix stored** command.



Note

This command is useful *only for verification purposes* because it is available *only after* the ISSU process has started. You might want to check the compatibility matrix prior to starting ISSU. Use the Feature Navigator to obtain the needed information:

<http://tools.cisco.com/ITDIT/CFN/jsp/index.jsp>

SNMP Support for ISSU

SNMP for SSO provides a mechanism for synchronizing the SNMP configurations and the MIBs that support SSO from the active supervisor engine to the standby supervisor engine, assuming that both supervisor engines are running the same version of Cisco IOS software. This assumption is not valid for ISSU.

With ISSU, an SNMP client can handle transformations for the MIBs across two different versions of Cisco IOS, if needed. An SNMP client handles transformation for all MIBs and handles the transmit and receive functionality across the active and standby supervisor engines. During SNMP, a MIB is completely synchronized from the active supervisor engine to the standby supervisor engine only if the versions of the MIB on both Cisco IOS releases are the same.

Compatibility Verification Using Cisco Feature Navigator

The ISSU application on Cisco Feature Navigator allows you to:

- Select an ISSU-capable image
- Identify which images are compatible with that image
- Compare two images and understand the compatibility level of the images (that is, compatible, base-level compatible, and incompatible)
- Compare two images and see the client compatibility for each ISSU client
- Provide links to release notes for the image

Performing the ISSU Process

Unlike SSO, which is a mode of operation for the device and a prerequisite for performing ISSU, the ISSU process is a series of steps performed while the switch is in operation. The steps result in an upgrade to a new or modified Cisco IOS software, and have a minimal impact to traffic.

**Note**

For an illustration of the process flow for ISSU, refer to [Figure 7-8 on page 7-11](#).

This section includes the following topics:

- [Upgrading ISSU to Cisco IOS XE 3.4.0SG/15.1\(2\)SG from a Prior Release, page 7-16](#)
- [Downgrading ISSU from Cisco IOS XE 3.4.0SG/15.1\(2\)SG to a Prior Release, page 7-17](#)
- [Verifying the ISSU Software Installation, page 7-18](#)
- [Loading New Cisco IOS Software on the Standby Supervisor Engine, page 7-21 \(required\)](#)
- [Switching to the Standby Supervisor Engine, page 7-24 \(required\)](#)
- [Stopping the ISSU Rollback Timer \(Optional\), page 7-26 \(optional\)](#)
- [Loading New Cisco IOS Software on the New Standby Supervisor Engine, page 7-27](#)
- [Aborting a Software Upgrade During ISSU, page 7-34](#)
- [Configuring the Rollback Timer to Safeguard Against Upgrade Issues, page 7-35](#)
- [Displaying ISSU Compatibility Matrix Information, page 7-36](#)

Upgrading ISSU to Cisco IOS XE 3.4.0SG/15.1(2)SG from a Prior Release

Because images prior to Cisco IOS XE 3.4.0SG/15.1(2)SG use the earlier CLI format and Cisco IOS XE 3.4.0SG and 15.1(2)SG images use a newer CLI format, your upgrade consists of the following:

- Upgrading the image on your switch to Cisco IOS XE 3.4.0SG/15.1(2)SG.
- Upgrading mgmtVrf from the earlier CLI format to the later format, removing any IPv6 addresses on the interface.
- Enabling IPv6 address family under mgmtVrf, and reconfigure IPv6 addresses on fa1.

A configuration like the following should exist on pre-Cisco IOS XE 3.4.0SG/15.1(2)SG image:

```
ip vrf mgmtVrf
!
interface FastEthernet1
 ip vrf forwarding mgmtVrf
 ip address 10.1.1.1 255.255.255.0
 speed auto
 duplex auto
 ipv6 address 2000::1/64
!
```

Step 1 Perform an ISSU upgrade to a Cisco IOS XE 3.4.0SG/15.1(2)SG image.

Step 2 Run the VRF upgrade command.

```
Switch# config t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# vrf upgrade-cli multi-af-mode common-policies vrf mgmtVrf
You are about to upgrade to the multi-AF VRF syntax commands.
You will lose any IPv6 address configured on interfaces
belonging to upgraded VRFs.

Are you sure ? [yes]:
Number of VRFs upgraded: 1
Switch(config)# exit
```

Your configuration will appear as follows:

```
vrf definition mgmtVrf
!
 address-family ipv4
 exit-address-family
!
interface FastEthernet1
 vrf forwarding mgmtVrf
 ip address 10.1.1.1 255.255.255.0
 speed auto
 duplex auto
!
```

Step 3 Configure the switch to enable the IPv6 address family and add the IPv6 address.

```
Switch# config t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# vrf definition mgmtVrf
Switch(config-vrf)# address-family ipv6
Switch(config-vrf-af)# exit
Switch(config-vrf)# exit
Switch(config)# interface fa1
Switch(config-if)# ipv6 address 2000::1/64
```

```
Switch(config-if)# end
```

Your configuration will appear as follows.

```
vrf definition mgmtVrf
!
address-family ipv4
exit-address-family
!
address-family ipv6
exit-address-family
!
interface FastEthernet1
vrf forwarding mgmtVrf
ip address 10.1.1.1 255.255.255.0
speed auto
duplex auto
ipv6 address 2000::1/64
```

Downgrading ISSU from Cisco IOS XE 3.4.0SG/15.1(2)SG to a Prior Release

Because a Cisco IOS XE 3.4.0SG/15.1(2)SG image uses a new CLI format and prior images use earlier CLI formats, the downgrade procedure include the following:

- Downgrading mgmtVrf from new CLI format to older CLI format, removing any IPv6 addresses on the interface.
- Downgrading the image on your switch to a prior release.
- Reconfiguring the IPv6 addresses on fa1.

A configuration like the following will appear on a switch running a Cisco IOS XE 3.4.0SG/15.1(2)SG image:

```
vrf definition mgmtVrf
!
address-family ipv4
exit-address-family
!
address-family ipv6
exit-address-family
!
interface FastEthernet1
vrf forwarding mgmtVrf
ip address 10.1.1.1 255.255.255.0
speed auto
duplex auto
ipv6 address 2000::1/64
!
```

Step 1 Perform a downgrade to a release prior to Cisco IOS XE 3.4.0SG/15.1(2)SG.

```
Switch# config t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# no vrf upgrade-cli multi-af-mode common-policies vrf mgmtVrf
You are about to downgrade to the single-AF VRF syntax commands.
You will lose any IPv6 address configured on interfaces
belonging to downgraded VRFs.

Are you sure ? [yes]:
% ipv6 addresses from all interfaces in VRF mgmtVrf have been removed
```

```
Number of VRFs downgraded: 1
Switch(config)#
```

Your configuration will appear as follows:

```
ip vrf mgmtVrf
!
interface FastEthernet1
 ip vrf forwarding mgmtVrf
 ip address 10.1.1.1 255.255.255.0
 speed auto
 duplex auto
!
```

Step 2 Perform an ISSU downgrade to a pre-Cisco IOS XE 3.4.0SG/15.1(2)SGn image.

Step 3 Reconfigure the IPv6 address.

```
Switch# config t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface fa1
Switch(config-if)# ipv6 address 2000::1/64
Switch(config-if)# end
Switch#
```

Your configuration will appear as follows.

```
ip vrf mgmtVrf
!
interface FastEthernet1
 ip vrf forwarding mgmtVrf
 ip address 10.1.1.1 255.255.255.0
 speed auto
 duplex auto
 ipv6 address 2000::1/64
```

Verifying the ISSU Software Installation

During the ISSU process, five valid states exist: disabled, init, load version, run version, and system reset. Use the **show issu state** command to obtain the current ISSU state:

- Disabled state—The state for the standby supervisor engine while this engine is resetting.
- Init state—The initial state is two supervisor engines, one active and one standby, before the ISSU process is started. It is also the final state after the ISSU process completes.
- Load version (LV) state—The standby supervisor engine is loaded with the new version of Cisco IOS software.
- Run version (RV) state—The **issu runversion** command forces the switchover of the supervisor engines. The newly active supervisor engine now runs the new Cisco IOS software image.
- System reset (SR) state—This state occurs either when you enter the **issu abortversion** command before the Init state is reached, or if the rollback timer expires before you execute the **issu acceptversion** command.

You can verify the ISSU software installation by entering **show** commands, as follows:

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	Switch# show issu state [detail]	Displays the state of the during the ISSU process.
Step 3	Switch# show redundancy	Displays current or historical status, mode, and related redundancy information about the device.

This example shows how to display the state and the current status of the supervisor engine during the ISSU process:

```
Switch> enable
Switch# show issu state
Switch# show redundancy
```

Verifying Redundancy Mode Before Beginning the ISSU Process

Before you begin the ISSU process, verify the redundancy mode for the system and be sure to configure NSF and SSO.

The following example displays verification that the system is in SSO mode, that slot 1 is the active supervisor engine, and that slot 2 is the standby supervisor engine. Both supervisor engines are running the same Cisco IOS software image.

```
Switch# show redundancy states
    my state = 13 -ACTIVE
    peer state = 8  -STANDBY HOT
        Mode = Duplex
        Unit = Primary
        Unit ID = 1

Redundancy Mode (Operational) = Stateful Switchover
Redundancy Mode (Configured)  = Stateful Switchover
Redundancy State               = Stateful Switchover
Maintenance Mode = Disabled
    Manual Swact = enabled
    Communications = Up

    client count = 39
    client_notification_TMR = 240000 milliseconds
        keep_alive TMR = 9000 milliseconds
        keep_alive count = 0
        keep_alive threshold = 18
        RF debug mask = 0x0

Switch# show redundancy
Redundant System Information :
-----
    Available system uptime = 1 minute
    Switchovers system experienced = 0
        Standby failures = 0
        Last switchover reason = none

        Hardware Mode = Duplex
    Configured Redundancy Mode = Stateful Switchover
```

```

Operating Redundancy Mode = Stateful Switchover
Maintenance Mode = Disabled
Communications = Up

Current Processor Information :
-----
      Active Location = slot 1
      Current Software state = ACTIVE
      Uptime in current state = 0 minutes
      Image Version = Cisco IOS Software, Catalyst 4500 L3 Switch Software
(cat4500-ENTSERVICES-M), Version 12.2(31)SGA, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Tue 05-Sep-06 16:16 by sanjdas
      BOOT = bootflash:old_image,1;
      Configuration register = 0x822

Peer Processor Information :
-----
      Standby Location = slot 2
      Current Software state = STANDBY HOT
      Uptime in current state = 1 minute
      Image Version = Cisco IOS Software, Catalyst 4500 L3 Switch Software
(cat4500-ENTSERVICES-M), Version 12.2(31)SGA, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Tue 05-Sep-06 16:16 by sanjdas
      BOOT = bootflash:old_image,1;
      Configuration register = 0x822

```

Verifying the ISSU State Before Beginning the ISSU Process

Ensure that the active and standby supervisor engines are up and in ISSU Init state and that the boot variables are set and pointing to valid files.

The following example displays the ISSU state before the process begins:

```

Switch# show issu state detail
      Slot = 1
      RP State = Active
      ISSU State = Init
      Boot Variable = bootflash:old_image,1;
      Operating Mode = Stateful Switchover
      Primary Version = N/A
      Secondary Version = N/A
      Current Version = bootflash:old_image

      Slot = 2
      RP State = Standby
      ISSU State = Init
      Boot Variable = bootflash:old_image,1;
      Operating Mode = Stateful Switchover
      Primary Version = N/A
      Secondary Version = N/A
      Current Version = bootflash:old_image

```

The new version of the Cisco IOS software must be present on both of the supervisor engines. The directory information displayed for each of the supervisor engines (or supervisor engines) shows that the new version is present.

```

Switch# dir bootflash:
Directory of bootflash:/

```



```

5 -rwx      13636500   Sep 6 2006 09:32:33 +00:00  old_image
6 -rwx      13636500   Sep 6 2006 09:34:07 +00:00  new_image

61341696 bytes total (1111388 bytes free)

Switch# dir slavebootflash:
Directory of slavebootflash:/

4 -rwx      13636500   Sep 6 2006 09:40:10 +00:00  old_image
5 -rwx      13636500   Sep 6 2006 09:42:13 +00:00  new_image

61341696 bytes total (1116224 bytes free)

```

Loading New Cisco IOS Software on the Standby Supervisor Engine

This task describes how to use ISSU to load a new version of Cisco IOS software to the standby supervisor engine.

Prerequisites

- Ensure that the new version of Cisco IOS software image is already present in the file system of both the active and standby supervisor engines. Also ensure that appropriate boot parameters (BOOT string and config-register) are set for the standby supervisor engine.



Note The switch must boot with the BOOT string setting before the ISSU procedure is attempted.



Note **auto-boot** must be enabled for ISSU to succeed.

- Optionally, perform additional tests and commands to determine the current state of peers and interfaces for later comparison.
- Ensure the system (both active and standby supervisor engines) is in SSO redundancy mode. If the system is in RPR mode rather than SSO mode, you can still upgrade the system using the ISSU CLI commands, but the system experiences extended packet loss during the upgrade.

Refer to the *Stateful Switchover* document for more details on how to configure SSO mode on supervisor engines.

- For ISSU to function, the image names on the active and standby supervisor engines must match.

Perform this task at the active supervisor engine:

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	Switch# issu loadversion <i>active-slot</i> <i>active-image-new standby-slot standby-image-new</i> [forced]	Starts the ISSU process and (optionally) overrides the automatic rollback when the new Cisco IOS software version is detected to be incompatible. It may take several seconds after the issu loadversion command is entered for Cisco IOS software to load onto the standby supervisor engine and for the standby supervisor engine to transition to SSO mode. This causes the standby supervisor engine to reload with the new image. If you use the forced option, the standby supervisor engine is booted with the new image. After the image is loaded on the standby supervisor engine, if the image is incompatible, the system is forced to the RPR mode. Otherwise the system continues in the SSO mode.
Step 3	Switch# show issu state [detail]	Displays the state of the during the ISSU process. At this point in the ISSU process, use this command to check that the standby supervisor engine is loaded and is in SSO mode. It may take several seconds after entering the issu loadversion command for Cisco IOS software to load onto the standby supervisor engine and the standby supervisor engine to transition to SSO mode. If you enter the show issu state command too quickly, you may not see the information you need.
Step 4	Switch# show redundancy [states]	Displays redundancy facility state information.

This example shows how to start the ISSU process, boot the standby supervisor engine in the Standby Hot state, and load the standby supervisor engine slot (2) with the new image:

```
Switch> enable
Switch# issu loadversion 1 bootflash:new_image 2 slavebootflash:new_image
Switch# show issu state detail
      Slot = 1
      RP State = Active
      ISSU State = Load Version
      Boot Variable = bootflash:old_image,12
      Operating Mode = Stateful Switchover
      Primary Version = bootflash:old_image
      Secondary Version = bootflash:new_image
      Current Version = bootflash:old_image

      Slot = 2
      RP State = Standby
      ISSU State = Load Version
      Boot Variable = bootflash:new_image,12;bootflash:old_image,12
      Operating Mode = Stateful Switchover
      Primary Version = bootflash:old_image
      Secondary Version = bootflash:new_image
      Current Version = bootflash:new_image
```

```

Switch# show redundancy states
    my state = 13 -ACTIVE
    peer state = 8  -STANDBY HOT
        Mode = Duplex
        Unit = Primary
        Unit ID = 1

Redundancy Mode (Operational) = Stateful Switchover
Redundancy Mode (Configured)  = Stateful Switchover
Redundancy State               = Stateful Switchover
Maintenance Mode = Disabled
    Manual Swact = enabled
    Communications = Up

    client count = 39
    client_notification_TMR = 240000 milliseconds
        keep_alive TMR = 9000 milliseconds
        keep_alive count = 1
        keep_alive threshold = 18
        RF debug mask = 0x0

```

The following example shows how the forced option places the system in RPR mode:

```

Switch> enable
Switch# issu loadversion 1 bootflash:new_image 2 slavebootflash:new_image forced
Switch# show issu state detail
    Slot = 1
        RP State = Active
        ISSU State = Load Version
        Boot Variable = bootflash:old_image,12
        Operating Mode = RPR
        Primary Version = bootflash:old_image
        Secondary Version = bootflash:new_image
        Current Version = bootflash:old_image

    Slot = 2
        RP State = Standby
        ISSU State = Load Version
        Boot Variable = bootflash:new_image,12;bootflash:old_image,12
        Operating Mode = RPR
        Primary Version = bootflash:old_image
        Secondary Version = bootflash:new_image
        Current Version = bootflash:new_image

```

The following example shows the redundancy mode as RPR:

```

Switch# show redundancy states
    my state = 13 -ACTIVE
    peer state = 4  -STANDBY COLD
        Mode = Duplex
        Unit = Primary
        Unit ID = 1

Redundancy Mode (Operational) = RPR
Redundancy Mode (Configured)  = Stateful Switchover
Redundancy State               = RPR
Maintenance Mode = Disabled
    Manual Swact = enabled
    Communications = Up

```

```

client count = 39
client_notification_TMR = 240000 milliseconds
keep_alive TMR = 9000 milliseconds
keep_alive count = 1
keep_alive threshold = 18
RF debug mask = 0x0

```

Switching to the Standby Supervisor Engine

This task describes how to switchover to the standby supervisor engine, which is running the new Cisco IOS software image.

Perform this task at the active supervisor engine:

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	Switch# issu runversion <i>standby-slot</i> <i>[standby-image-new]</i>	Forces a switchover from the active to the standby supervisor engine and reloads the former active (current standby) supervisor engines with the old image. When you enter the issu runversion command, an SSO switchover is performed, and NSF procedures are invoked if configured.
Step 3	Switch# show issu state <i>[detail]</i>	Displays the state of the during the ISSU process. At this point in the ISSU process, use this command to check that a switchover occurs to slot 2.
Step 4	Switch# show redundancy <i>[states]</i>	Displays redundancy facility state information.

This example shows how to cause a switchover to the former standby supervisor engine (slot 2), reset the former active supervisor engine and reload it with the old image so it becomes the standby supervisor engine:

```

Switch> enable
Switch# issu runversion 2 slavebootflash:new_image
This command will reload the Active unit. Proceed ? [confirm]

```

A switchover occurs at this point. At the new active supervisor engine, after old active supervisor engine comes up as the standby engine, do the following:

```

Switch# show issu state detail
      Slot = 2
      RP State = Active
      ISSU State = Run Version
      Boot Variable = bootflash:new_image,12;bootflash:old_image,12
      Operating Mode = Stateful Switchover
      Primary Version = bootflash:new_image
      Secondary Version = bootflash:old_image
      Current Version = bootflash:new_image

      Slot = 1
      RP State = Standby
      ISSU State = Run Version

```

```

Boot Variable = bootflash:old_image,12
Operating Mode = Stateful Switchover
Primary Version = bootflash:new_image
Secondary Version = bootflash:old_image
Current Version = bootflash:old_image

```

**Note**

The new active supervisor engine is now running the new version of software, and the standby supervisor engine is running the old version of software and is in the standby hot state.

```
Switch# show redundancy states
```

```

my state = 13 -ACTIVE
peer state = 8 -STANDBY HOT
Mode = Duplex
Unit = Secondary
Unit ID = 2

```

```

Redundancy Mode (Operational) = Stateful Switchover
Redundancy Mode (Configured) = Stateful Switchover
Redundancy State = Stateful Switchover
Maintenance Mode = Disabled
Manual Swact = enabled
Communications = Up

```

```

client count = 39
client_notification_TMR = 240000 milliseconds
keep_alive TMR = 9000 milliseconds
keep_alive count = 1
keep_alive threshold = 18
RF debug mask = 0x0

```

Once **runversion** command completes, the new active supervisor engine is running the new version of software and the previously active supervisor engine now becomes the standby supervisor engine. The standby is reset and reloaded, but remains on the previous version of software and come back online in standbyhot status. The following example shows how to verify these conditions:

```
Switch# show redundancy
```

```
Redundant System Information :
```

```

-----
Available system uptime = 23 minutes
Switchovers system experienced = 1
Standby failures = 0
Last switchover reason = user forced

```

```

Hardware Mode = Duplex
Configured Redundancy Mode = Stateful Switchover
Operating Redundancy Mode = Stateful Switchover
Maintenance Mode = Disabled
Communications = Up

```

```
Current Processor Information :
```

```

-----
Active Location = slot 2
Current Software state = ACTIVE
Uptime in current state = 11 minutes
Image Version = Cisco IOS Software, Catalyst 4500 L3 Switch Software
(cat4500-ENTSERVICES-M), Version 12.2(31)SGA, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Tue 05-Sep-06 16:16 by sanjdas
BOOT = bootflash:new_image,12;bootflash:old_image,12
Configuration register = 0x822

```

Peer Processor Information :

```

-----
Standby Location = slot 1
Current Software state = STANDBY HOT
Uptime in current state = 4 minutes
Image Version = Cisco IOS Software, Catalyst 4500 L3 Switch Software
(cat4500-ENTSERVICES-M), Version 12.2(31)SGA, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Tue 05-Sep-06 16:16 by sanjdas
BOOT = bootflash:old_image,12
Configuration register = 0x822

```

Stopping the ISSU Rollback Timer (Optional)

This optional task describes how to stop the rollback timer.

If you do not run the following procedure before the rollback timer “timeout,” the system automatically aborts the ISSU process and reverts to the original Cisco IOS software version. By default the rollback timer is 45 minutes.

Use the following information to decide what action you should take:

- If you want to retain your switch in this state for an extended period, you need to stop the rollback timer (then validate and run the **acceptversion** command directly).
- If you want to proceed to the following step (running “commitversion”) within the rollback timer window of 45 minutes, you do not need to stop the rollback timer.

**Note**

The **issu acceptversion** command can be optionally executed after the **issu runversion** command.

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	Switch# issu acceptversion active-slot [active-image-new]	Halts the rollback timer and ensures the new Cisco IOS ISSU process is not automatically aborted during the ISSU process. Enter the issu acceptversion command within the time period specified by the rollback timer to acknowledge that the supervisor engine has achieved connectivity to the outside world; otherwise, the ISSU process is terminated, and the system reverts to the previous version of Cisco IOS software by switching to the standby supervisor engine.
Step 3	Switch# show issu rollback-timer	Displays the amount of time left before an automatic rollback occurs.

This example displays the timer before you stop it. In the following example, the Automatic Rollback Time information indicates the amount of time remaining before an automatic rollback occurs.

```

Switch> enable
Switch# show issu rollback-timer
Rollback Process State = In progress

```

```
Configured Rollback Time = 45:00
Automatic Rollback Time = 38:30

Switch# issu acceptversion 2 bootflash:new_image
% Rollback timer stopped. Please issue the commitversion command.
Switch# show issu rollback-timer
Rollback Process State = Not in progress
Configured Rollback Time = 45:00
```

Loading New Cisco IOS Software on the New Standby Supervisor Engine

This task explains how to load new version of Cisco IOS software to the new standby supervisor engine. Perform this task at the active supervisor engine:

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	Switch# issu commitversion <i>standby-slot-number</i> [<i>standby-image-new</i>]	Allows the new Cisco IOS software image to be loaded into the standby supervisor engine.
Step 3	Switch# show redundancy [states]	Displays redundancy facility state information.
Step 4	Switch# show issu state [detail]	Displays the state of the during the ISSU process. At this point in the ISSU process, use this command to check that a switchover occurs to slot 2.

This example shows how to reset and reload the current standby supervisor engine (slot 1) with the new Cisco IOS software version. After entering the **commitversion** command, the standby supervisor engine boots in the Standby Hot state.

```
Switch> enable
Switch# issu commitversion 1 slavebootflash:new_image
```

Wait till standby supervisor is reloaded with the new image. Then apply the following:

```
Switch# show redundancy states
00:17:12: %RF-5-RF_TERMINAL_STATE: Terminal state reached for (SSO)
    my state = 13 -ACTIVE
    peer state = 8  -STANDBY HOT
        Mode = Duplex
        Unit = Secondary
        Unit ID = 2

Redundancy Mode (Operational) = Stateful Switchover
Redundancy Mode (Configured)  = Stateful Switchover
Redundancy State               = Stateful Switchover
Maintenance Mode = Disabled
    Manual Swact = enabled
    Communications = Up

    client count = 39
    client_notification_TMR = 240000 milliseconds
        keep_alive TMR = 9000 milliseconds
            keep_alive count = 0
            keep_alive threshold = 18
            RF debug mask = 0x0
```

```

Switch# show redundancy
Redundant System Information :
-----
    Available system uptime = 41 minutes
Switchovers system experienced = 1
    Standby failures = 1
    Last switchover reason = user forced

    Hardware Mode = Duplex
Configured Redundancy Mode = Stateful Switchover
Operating Redundancy Mode = Stateful Switchover
    Maintenance Mode = Disabled
    Communications = Up

Current Processor Information :
-----
    Active Location = slot 2
    Current Software state = ACTIVE
    Uptime in current state = 29 minutes
    Image Version = Cisco IOS Software, Catalyst 4500 L3 Switch Software
(cat4500-ENTSERVICES-M), Version 12.2(31)SGA, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Tue 05-Sep-06 16:16 by sanjdas
    BOOT = bootflash:new_image,12;bootflash:old_image,1;
    Configuration register = 0x822

Peer Processor Information :
-----
    Standby Location = slot 1
    Current Software state = STANDBY HOT
    Uptime in current state = 12 minutes
    Image Version = Cisco IOS Software, Catalyst 4500 L3 Switch Software
(cat4500-ENTSERVICES-M), Version 12.2(31)SGA, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Tue 05-Sep-06 16:16 by sanjdas
    BOOT = bootflash:new_image,12;bootflash:old_image,1;
    Configuration register = 0x822

Switch# show issu state detail
    Slot = 2
    RP State = Active
    ISSU State = Init
    Boot Variable = bootflash:new_image,12;bootflash:old_image,1;
    Operating Mode = Stateful Switchover
    Primary Version = N/A
    Secondary Version = N/A
    Current Version = bootflash:new_image

    Slot = 1
    RP State = Standby
    ISSU State = Init
    Boot Variable = bootflash:new_image,12;bootflash:old_image,1;
    Operating Mode = Stateful Switchover
    Primary Version = N/A
    Secondary Version = N/A
    Current Version = bootflash:new_image

```

The ISSU process has been completed. At this stage, any further Cisco IOS software version upgrade or downgrade requires that a new ISSU process be invoked.

Using changeversion to Automate an ISSU Upgrade

This task describes how to use the **issu changeversion** command to perform a one step ISSU upgrade.

Prerequisites

- Ensure that the new version of Cisco IOS software image is already present in the file system of both the active and standby supervisor engines. Also ensure that appropriate boot parameters (BOOT string and config-register) are set for the active and standby supervisor engines
- Optionally, perform additional tests and commands to determine the current state of peers and interfaces for later comparison.
- Ensure the system (both active and standby supervisor engines) is in SSO redundancy mode. If the system is in RPR mode, you can still upgrade the system using the ISSU CLI commands, but the system will experience extended packet loss during the upgrade.'

Refer to the Stateful Switchover document for more details on how to configure SSO mode on supervisor engines (refer to [Chapter 11, “Configuring Supervisor Engine Redundancy Using RPR and SSO on Supervisor Engine 6-E and Supervisor Engine 6L-E”](#)).

- For ISSU to function, the IOS XE software image file names on the active and standby supervisor engines must match.

Perform the following steps at the active supervisor engine:

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	Switch# issu changeversion [<i>active-slot</i> <i>active-image-new</i>] [<i>standby-slot</i> <i>standby-image-new</i>] [at <i>hh:mm</i> in <i>hh:mm</i>] [quick]	Initiates a single-step complete upgrade process cycle. Performs the logic of the four standard commands (issu loadversion, issu runversion, issu acceptversion, and issu commitversion) without user intervention. <i>active-slot</i> —Defines the active slot number. <i>new-image</i> —Specifies IOS XE image URL to be upgraded to. <i>standby-slot</i> —Defines the standby slot number. <i>standby-image</i> —Specifies the standby IOS XE image URL. at <i>hh:mm</i> —Schedules an ISSU upgrade to begin in the future. Provides an exact time (<i>hh:mm</i> , 24 hour format) in the next 24 hours when the upgrade will occur. in <i>hh:mm</i> —Schedules an ISSU upgrade to begin in the future. Provides the number of hours and minutes (<i>hh:mm</i> format) that will elapse before an upgrade will occur (99:59 max). quick —Upon switchover, boots the standby supervisor engine with the new, rather than old, image for faster upgrade.

	Command or Action	Purpose
Step 3	Switch# show issu state [detail]	Displays the state of the during the ISSU process. At this point in the ISSU process, use this command to check that the standby supervisor engine is loaded and is in SSO mode.
Step 4	Switch# show redundancy [states]	Displays redundancy facility state information.

This example shows how to initiate an ISSU upgrade process using the `issu changeversion` command on slot number 5, the slot for the current active supervisor engine. The `show issu state detail` and `show redundancy` command output is included to show the supervisor state before and after the upgrade procedure.

**Note**

The success messages included in the output below is displayed after some delay because the ISSU upgrade procedure progresses through the ISSU states.

```
Switch> enable
Switch# show issu state detail
                Slot = 5
                RP State = Active
                ISSU State = Init
                Operating Mode = Stateful Switchover
                Current Image = bootflash:x.bin
                Pre-ISSU (Original) Image = N/A
                Post-ISSU (Targeted) Image = N/A

                Slot = 6
                RP State = Standby
                ISSU State = Init
                Operating Mode = Stateful Switchover
                Current Image = bootflash:x.bin
                Pre-ISSU (Original) Image = N/A
                Post-ISSU (Targeted) Image = N/A

Switch# show redundancy
Redundant System Information :

-----
        Available system uptime = 12 minutes
Switchovers system experienced = 0
        Standby failures = 0
        Last switchover reason = none

        Hardware Mode = Duplex
        Configured Redundancy Mode = Stateful Switchover
        Operating Redundancy Mode = Stateful Switchover
        Maintenance Mode = Disabled
        Communications = Up

Current Processor Information :
-----
        Active Location = slot 5
        Current Software state = ACTIVE
        Uptime in current state = 9 minutes
        Image Version = Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3
Switch Software (cat4500e-UNIVERSALK9-M), Version 03.00.00.1.68 CISCO UNIVERSAL
DEVELOPMENT K10 IOSD TEST VERSION
Copyright (c) 1986-2010 by Cisco Systems, Inc.
Compiled Sun 29-Aug-10 03:57 by gsbuprod
Configuration register = 0x2920
```

Peer Processor Information :

```
-----
      Standby Location = slot 6
      Current Software state = STANDBY HOT
      Uptime in current state = 2 minutes
      Image Version = Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3
Switch Software (cat4500e-UNIVERSALK9-M), Version 03.00.00.1.68 CISCO UNIVERSAL
DEVELOPMENT K10 IOSD TEST VERSION
Copyright (c) 1986-2010 by Cisco Systems, Inc.
Compiled Sun 29-Aug-10 03:57 by gsbuprod
      Configuration register = 0x2920
```

Switch# **issu changeversion bootflash:y.bin**

% 'issu changeversion' is now executing 'issu loadversion'

% issu loadversion executed successfully, Standby is being reloaded

% changeversion finished executing loadversion, waiting for standby to reload and reach SSO ...



Note Standby reloads with target image.

.....
.....

*Feb 25 20:41:00.479: %INSTALLER-7-ISSU_OP_SUCC: issu changeversion is now executing 'issu runversion'

*Feb 25 20:41:03.639: %INSTALLER-7-ISSU_OP_SUCC: issu changeversion successfully executed 'issu runversion'



Note Switchover occurs.

.....
.....

Look at the console of new active supervisor engine.

*Feb 25 20:47:39.859: %RF-5-RF_TERMINAL_STATE: Terminal state reached for (SSO)

*Feb 25 20:47:39.971: %INSTALLER-7-ISSU_OP_SUCC: issu changeversion is now executing 'issu commitversion'

....
....



Note The new standby supervisor reloads with target image; changeversion is successful upon SSO terminal state is reached.

*Feb 25 20:54:16.092: %HA_CONFIG_SYNC-6-BULK_CFGSYNC_SUCCEED: Bulk Sync succeeded

*Feb 25 20:54:16.094: %RF-5-RF_TERMINAL_STATE: Terminal state reached for (SSO)

Switch#

Switch# **show issu state detail**

```

      Slot = 6
      RP State = Active
      ISSU State = Init
      Operating Mode = Stateful Switchover
      Current Image = bootflash:y.bin
      Pre-ISSU (Original) Image = N/A
```

```

Post-ISSU (Targeted) Image = N/A

Slot = 5
RP State = Standby
ISSU State = Init
Operating Mode = Stateful Switchover
Current Image = bootflash:y.bin
Pre-ISSU (Original) Image = N/A
Post-ISSU (Targeted) Image = N/A

Switch# show redundancy
Redundant System Information :

-----
Available system uptime = 12 minutes
Switchovers system experienced = 0
Standby failures = 0
Last switchover reason = none

Hardware Mode = Duplex
Configured Redundancy Mode = Stateful Switchover
Operating Redundancy Mode = Stateful Switchover
Maintenance Mode = Disabled
Communications = Up

Current Processor Information :
-----
Active Location = slot 6
Current Software state = ACTIVE
Uptime in current state = 9 minutes
Image Version = Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3
Switch Software (cat4500e-UNIVERSALK9-M), Version 03.00.00.1.68 CISCO UNIVERSAL
DEVELOPMENT K10 IOSD TEST VERSION
Copyright (c) 1986-2010 by Cisco Systems, Inc.
Compiled Sun 29-Aug-10 03:57 by gsbuprod
Configuration register = 0x2920

Peer Processor Information :
-----
Standby Location = slot 5
Current Software state = STANDBY HOT
Uptime in current state = 2 minutes
Image Version = Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3
Switch Software (cat4500e-UNIVERSALK9-M), Version 03.00.00.1.68 CISCO UNIVERSAL
DEVELOPMENT K10 IOSD TEST VERSION
Copyright (c) 1986-2010 by Cisco Systems, Inc.
Compiled Sun 29-Aug-10 03:57 by gsbuprod
Configuration register = 0x2920

```

This example shows how to use `issu changeversion` with the `at` command option to schedule an ISSU upgrade procedure to automatically start at the specified time. This example specifies that the ISSU upgrade should be started at 16:30 (24 hour format). The **show issu state detail** and **show redundancy** command output is included to show the supervisor state before and after the **issu changeversion** command was entered.

```

Switch> enable
Switch# show issu state detail

Slot = 5
RP State = Active
ISSU State = Init
Operating Mode = Stateful Switchover
Current Image = bootflash:x.bin
Pre-ISSU (Original) Image = N/A

```

```

Post-ISSU (Targeted) Image = N/A

Slot = 6
RP State = Standby
ISSU State = Init
Operating Mode = Stateful Switchover
Current Image = bootflash:x.bin
Pre-ISSU (Original) Image = N/A
Post-ISSU (Targeted) Image = N/A

Switch# show redundancy
Redundant System Information :

-----
Available system uptime = 12 minutes
Switchovers system experienced = 0
Standby failures = 0
Last switchover reason = none

Hardware Mode = Duplex
Configured Redundancy Mode = Stateful Switchover
Operating Redundancy Mode = Stateful Switchover
Maintenance Mode = Disabled
Communications = Up

Current Processor Information :
-----
Active Location = slot 5
Current Software state = ACTIVE
Uptime in current state = 9 minutes
Image Version = Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3
Switch Software (cat4500e-UNIVERSALK9-M), Version 03.00.00.1.68 CISCO UNIVERSAL
DEVELOPMENT K10 IOSD TEST VERSION
Copyright (c) 1986-2010 by Cisco Systems, Inc.
Compiled Sun 29-Aug-10 03:57 by gsbuprod
Configuration register = 0x2920

Peer Processor Information :
-----
Standby Location = slot 6
Current Software state = STANDBY HOT
Uptime in current state = 2 minutes
Image Version = Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3
Switch Software (cat4500e-UNIVERSALK9-M), Version 03.00.00.1.68 CISCO UNIVERSAL
DEVELOPMENT K10 IOSD TEST VERSION
Copyright (c) 1986-2010 by Cisco Systems, Inc.
Compiled Sun 29-Aug-10 03:57 by gsbuprod
Configuration register = 0x2920

Switch# issu changeversion 5 bootflash:y.bin 6 slavebootflash:y at 16:30
% 'issu changeversion' was executed at [ Apr 12 16:27:43 ].
% The planned ISSU changeversion is to occur in (hh:mm:ss) [ 00:03:00 ] at [ Apr 12
16:30:43 ].
% Current system time: [ Apr 12 16:27:43 ]
% Planned upgrade image: bootflash:y.bin
% To cancel the planned upgrade, please execute 'issu abortversion'

Switch# show issu state detail

Slot = 5
RP State = Active
ISSU State = Init
Changeversion = TRUE
Operating Mode = Stateful Switchover
Current Image = bootflash:x.bin

```

```

Pre-ISSU (Original) Image = N/A
Post-ISSU (Targeted) Image = N/A

Slot = 6
RP State = Standby
ISSU State = Init
Changeversion = TRUE
Operating Mode = Stateful Switchover
Current Image = bootflash:x.bin
Pre-ISSU (Original) Image = N/A
Post-ISSU (Targeted) Image = N/A

```

Aborting a Software Upgrade During ISSU

You can abort the ISSU process at any stage manually (prior to entering the **issu commitversion** command) by entering the **issu abortversion** command. The ISSU process also aborts on its own if the software detects a failure.



Note

If you enter the **issu abortversion** command before the standby supervisor engine becomes hot, the traffic might be disrupted.

If you abort the process after you enter the **issu loadversion** command, the standby supervisor engine is reset and reloaded with the original software.

If the process is aborted after you enter either the **issu runversion** or **issu acceptversion** command, then a second switchover is performed to the new standby supervisor engine that is still running the original software version. The supervisor engine that had been running the new software is reset and reloaded with the original software version.



Note

Ensure that the standby supervisor engine is fully booted *before* entering the **abortversion** command on an active supervisor engine.

The following task describes how to abort the ISSU process before you complete the ISSU process with the **issu commitversion** command.

Perform the following task on the active supervisor engine:

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	Switch# issu abortversion <i>active slot</i> <i>[active-image-new]</i>	Cancels the ISSU upgrade or downgrade process in progress and restores the router to its state before the process had started.

This example shows how to abort the ISSU process on slot number 2, the slot for the current active supervisor engine:

```

Switch> enable
Switch# issu abortversion 2

```

Configuring the Rollback Timer to Safeguard Against Upgrade Issues

Cisco IOS software maintains an ISSU rollback timer, to safeguard against an upgrade that may leave the new active supervisor engine in a state in which communication with the standby supervisor engine is severed.

You may want to configure the rollback timer to fewer than 45 minutes (the default) so that the user need not wait in case the new software is not committed or the connection to the switch was lost while it was in runversion mode. A user may want to configure the rollback timer to more than 45 minutes in order to have enough time to verify the operation of the new Cisco IOS software before committing the new image.



Note

The valid timer value range is from 0 to 7200 seconds (two hours). A value of 0 seconds disables the rollback timer.

Once you are satisfied that the ISSU process has been successful and you want to remain in the current state, you must indicate acceptance by entering the **issu acceptversion** command, which stops the rollback timer. Entering the **issu acceptversion** command is extremely important in advancing the ISSU process.

Entering the **issu commitversion** command at this stage is equal to entering both the **issu acceptversion** and the **issu commitversion** commands. Use the **issu commitversion** command if you do not intend to run in the current state now and are satisfied with the new software version.



Note

The rollback timer can be configured only in the ISSU Init state.

Perform this task to configure the rollback timer:

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	Switch# configure terminal	Enters global configuration mode.
Step 3	Switch(config)# issu set rollback-timer <i>hh:mm:ss</i>	Configures the rollback timer value.
Step 4	Switch(config)# exit	Returns the user to privileged EXEC mode.
Step 5	Switch# show issu rollback-timer	Displays the current setting of the ISSU rollback timer.

This example shows how to set the rollback timer to 3600 seconds:

```
Switch> enable
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# issu set rollback-timer 3600
% Rollback timer value set to [ 3600 ] seconds

Switch(config)# exit

Switch# show issu rollback-timer
Rollback Process State = Not in progress
Configured Rollback Time = 60:00
```

The rollback timer cannot be set in LV state, as the following example illustrates:

```
Switch# show issu state detail
      Slot = 1
      RP State = Active
      ISSU State = Load Version
      Boot Variable = bootflash:old_image,12
      Operating Mode = RPR
      Primary Version = bootflash:old_image
      Secondary Version = bootflash:new_image
      Current Version = bootflash:old_image

      Slot = 2
      RP State = Standby
      ISSU State = Load Version
      Boot Variable = bootflash:new_image,12;bootflash:old_image,12
      Operating Mode = RPR
      Primary Version = bootflash:old_image
      Secondary Version = bootflash:new_image
      Current Version = bootflash:new_image

Switch# show issu rollback-timer
      Rollback Process State = Not in progress
      Configured Rollback Time = 60:00

Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# issu set rollback-timer 20
% ISSU state should be [ init ] to set the rollback timer
```

Displaying ISSU Compatibility Matrix Information

The ISSU compatibility matrix contains information about other software images about the version in question. This compatibility matrix represents the compatibility of the two software versions, one running on the active and the other on the standby supervisor engine, and the matrix allows the system to determine the highest operating mode it can achieve. This information helps the user identify whether to use ISSU.

Perform this task to display information about the ISSU compatibility matrix:

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	Switch# show issu comp-matrix { negotiated stored xml }	Displays information regarding the ISSU compatibility matrix. <ul style="list-style-type: none"> negotiated—Displays negotiated compatibility matrix information. stored—Displays negotiated compatibility matrix information. xml—Displays negotiated compatibility matrix information in XML format.

This example shows how to display negotiated information regarding the compatibility matrix:


```
Switch> enable
Switch# show issu comp-matrix negotiated
```

```
CardType: WS-C4507R(112), Uid: 2, Image Ver: 12.2(31)SGA
Image Name: cat4500-ENTSERVICES-M
```

Cid	Eid	Sid	pSid	pUid	Compatibility
2	1	262151	3	1	COMPATIBLE
3	1	262160	5	1	COMPATIBLE
4	1	262163	9	1	COMPATIBLE
5	1	262186	25	1	COMPATIBLE
7	1	262156	10	1	COMPATIBLE
8	1	262148	7	1	COMPATIBLE
9	1	262155	1	1	COMPATIBLE
10	1	262158	2	1	COMPATIBLE
11	1	262172	6	1	COMPATIBLE
100	1	262166	13	1	COMPATIBLE
110	113	262159	14	1	COMPATIBLE
200	1	262167	24	1	COMPATIBLE
2002	1	-	-	-	UNAVAILABLE
2003	1	262185	23	1	COMPATIBLE
2004	1	262175	16	1	COMPATIBLE
2008	1	262147	26	1	COMPATIBLE
2008	1	262168	27	1	COMPATIBLE
2010	1	262171	32	1	COMPATIBLE
2012	1	262180	31	1	COMPATIBLE
2021	1	262170	41	1	COMPATIBLE
2022	1	262152	42	1	COMPATIBLE
2023	1	-	-	-	UNAVAILABLE
2024	1	-	-	-	UNAVAILABLE
2025	1	-	-	-	UNAVAILABLE
2026	1	-	-	-	UNAVAILABLE
2027	1	-	-	-	UNAVAILABLE
2028	1	-	-	-	UNAVAILABLE
2054	1	262169	8	1	COMPATIBLE
2058	1	262154	29	1	COMPATIBLE
2059	1	262179	30	1	COMPATIBLE
2067	1	262153	12	1	COMPATIBLE
2068	1	196638	40	1	COMPATIBLE
2070	1	262145	21	1	COMPATIBLE
2071	1	262178	11	1	COMPATIBLE
2072	1	262162	28	1	COMPATIBLE
2073	1	262177	33	1	COMPATIBLE
2077	1	262165	35	1	COMPATIBLE
2078	1	196637	34	1	COMPATIBLE
2079	1	262176	36	1	COMPATIBLE
2081	1	262150	37	1	COMPATIBLE
2082	1	262161	39	1	COMPATIBLE
2083	1	262184	20	1	COMPATIBLE
2084	1	262183	38	1	COMPATIBLE
4001	101	262181	17	1	COMPATIBLE
4002	201	262164	18	1	COMPATIBLE
4003	301	262182	19	1	COMPATIBLE
4004	401	262146	22	1	COMPATIBLE
4005	1	262149	4	1	COMPATIBLE

Message group summary:

Cid	Eid	GrpId	Sid	pSid	pUid	Nego Result
2	1	1	262151	3	1	Y
3	1	1	262160	5	1	Y
4	1	1	262163	9	1	Y
5	1	1	262186	25	1	Y

7	1	1	262156	10	1	Y
8	1	1	262148	7	1	Y
9	1	1	262155	1	1	Y
10	1	1	262158	2	1	Y
11	1	1	262172	6	1	Y
100	1	1	262166	13	1	Y
110	113	115	262159	14	1	Y
200	1	1	262167	24	1	Y
2002	1	2	-	-	-	N - did not negotiate
2003	1	1	262185	23	1	Y
2004	1	1	262175	16	1	Y
2008	1	1	262147	26	1	Y
2008	1	2	262168	27	1	Y
2010	1	1	262171	32	1	Y
2012	1	1	262180	31	1	Y
2021	1	1	262170	41	1	Y
2022	1	1	262152	42	1	Y
2023	1	1	-	-	-	N - did not negotiate
2024	1	1	-	-	-	N - did not negotiate
2025	1	1	-	-	-	N - did not negotiate
2026	1	1	-	-	-	N - did not negotiate
2027	1	1	-	-	-	N - did not negotiate
2028	1	1	-	-	-	N - did not negotiate
2054	1	1	262169	8	1	Y
2058	1	1	262154	29	1	Y
2059	1	1	262179	30	1	Y
2067	1	1	262153	12	1	Y
2068	1	1	196638	40	1	Y
2070	1	1	262145	21	1	Y
2071	1	1	262178	11	1	Y
2072	1	1	262162	28	1	Y
2073	1	1	262177	33	1	Y
2077	1	1	262165	35	1	Y
2078	1	1	196637	34	1	Y
2079	1	1	262176	36	1	Y
2081	1	1	262150	37	1	Y
2082	1	1	262161	39	1	Y
2083	1	1	262184	20	1	Y
2084	1	1	262183	38	1	Y
4001	101	1	262181	17	1	Y
4002	201	1	262164	18	1	Y
4003	301	1	262182	19	1	Y
4004	401	1	262146	22	1	Y
4005	1	1	262149	4	1	Y

List of Clients:

Cid	Client Name	Base/Non-Base
2	ISSU Proto client	Base
3	ISSU RF	Base
4	ISSU CF client	Base
5	ISSU Network RF client	Base
7	ISSU CONFIG SYNC	Base
8	ISSU ifIndex sync	Base
9	ISSU IPC client	Base
10	ISSU IPC Server client	Base
11	ISSU Red Mode Client	Base
100	ISSU rfs client	Base
110	ISSU ifs client	Base
200	ISSU Event Manager client	Base
2002	CEF Push ISSU client	Base
2003	ISSU XDR client	Base
2004	ISSU SNMP client	Non-Base
2008	ISSU Tableid Client	Base

```

2010      ARP HA                      Base
2012      ISSU HSRP Client            Non-Base
2021      XDR Int Priority ISSU cliBase
2022      XDR Proc Priority ISSU clBase
2023      FIB HWIDB ISSU client      Base
2024      FIB IDB ISSU client        Base
2025      FIB HW subblock ISSU clieBase
2026      FIB SW subblock ISSU clieBase
2027      Adjacency ISSU client      Base
2028      FIB IPV4 ISSU client        Base
2054      ISSU process client        Base
2058      ISIS ISSU RTR client        Non-Base
2059      ISIS ISSU UPD client        Non-Base
2067      ISSU PM Client             Base
2068      ISSU PAGP_SWITCH Client    Non-Base
2070      ISSU Port Security clientNon-Base
2071      ISSU Switch VLAN client    Non-Base
2072      ISSU dot1x client          Non-Base
2073      ISSU STP                   Non-Base
2077      ISSU STP MSTP              Non-Base
2078      ISSU STP IEEE              Non-Base
2079      ISSU STP RSTP              Non-Base
2081      ISSU DHCP Snooping clientNon-Base
2082      ISSU IP Host client         Non-Base
2083      ISSU Inline Power clientNon-Base
2084      ISSU IGMP Snooping clientNon-Base
4001      ISSU C4K Chassis client    Base
4002      ISSU C4K Port client        Base
4003      ISSU C4K Rkios client      Base
4004      ISSU C4K HostMan client    Base
4005      ISSU C4k GaliosRedundancyBase

```

This example shows how to display stored information regarding the compatibility matrix:

```
Switch# show issu comp-matrix stored
```

```
Number of Matrices in Table = 1
```

```
(1) Matrix for cat4500-ENTSERVICES-M(112) - cat4500-ENTSERVICES-M(112)
```

```
=====
```

```
Start Flag (0xDEADBABE)
```

```

My Image ver: 12.2(53)SG
Peer Version  Compatibility
-----
12.2(31)SGA5      Base(2)
12.2(44)SG        Base(2)
12.2(31)SGA6      Base(2)
12.2(31)SGA7      Base(2)
12.2(46)SG        Base(2)
12.2(44)SG1       Base(2)
12.2(31)SGA8      Base(2)
12.2(50)SG        Dynamic(0)
12.2(31)SGA9      Base(2)
12.2(50)SG1       Dynamic(0)
12.2(50)SG2       Dynamic(0)
12.2(52)SG        Dynamic(0)
12.2(31)SGA10     Base(2)
12.2(50)SG3       Dynamic(0)
12.2(53)SG        Comp(3)

```

Dynamic(0) was introduced in Cisco IOS Release 12.2(50)SG with the Dynamic Image Version Compatibility (DIVC) feature. With DIVC, Dynamic(0) is stored instead of Incomp(1), Base(2), or Comp(3). Compatibility is determined during runtime when two different DIVC-capable images are running in the active and standby supervisor engines during ISSU.

For Catalyst 4500 switches, a value of Dynamic(0) in the stored compatibility-matrix normally results in Base(2) or Comp(3) upon rollback negotiation between the two images. You never observe Incomp(1) as long as the other image name is present in the stored compatibility matrix.

Displaying ISSU Compatibility Matrix Information

The ISSU compatibility matrix contains information about other IOS XE software releases and the version in question. This compatibility matrix represents the compatibility of the two software versions, one running on the active and the other on the standby supervisor engine, and the matrix allows the system to determine the highest operating mode it can achieve. This information helps the user identify whether to use ISSU.

This task shows how to display information about the ISSU compatibility matrix:

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	Switch# show issu comp-matrix { negotiated stored xml }	Displays information regarding the ISSU compatibility matrix. <ul style="list-style-type: none"> • negotiated—Displays negotiated compatibility matrix information. • stored—Displays negotiated compatibility matrix information. • xml—Displays negotiated compatibility matrix information in XML format. <p>Note These commands display only the data within IOSd process. Use the show package compatibility to display the information for the whole system.</p>
Step 3	Switch# show package compatibility	Displays information regarding all client compatibility in the system.

This example shows how to display negotiated information regarding the compatibility matrix:

```
Switch> enable
Switch# show issu comp-matrix negotiated

CardType: WS-C4507R-E(182), Uid: 4, Image Ver: 03.00.00.1.68
Image Name: cat4500e-UNIVERSALK9-M

Cid      Eid      Sid      pSid     pUid      Compatibility
=====
2        1        131078   3        3         COMPATIBLE
3        1        131100   5        3         COMPATIBLE
4        1        131123   9        3         COMPATIBLE
.....
.....
```

```

Message group summary:
Cid      Eid      GrpId      Sid      pSid      pUId      Nego Result
=====
2        1        1          131078   3         3         Y
3        1        1          131100   5         3         Y
4        1        1          131123   9         3         Y
.....
.....

List of Clients:
Cid      Client Name      Base/Non-Base
=====
2        ISSU Proto client Base
3        ISSU RF          Base
4        ISSU CF client   Base
.....
.....

```

This example shows how to display stored information regarding the compatibility matrix:

```
Switch# show issu comp-matrix stored
```

```
Number of Matrices in Table = 1
```

```

(1) Matrix for cat4500e-ENTSERVICESK9-M(182) - cat4500ex-ENTSERVICESK9-M(182)
=====
Start Flag (0xDEADBABE)

My Image ver: 03.01.00.SG
Peer Version  Compatibility
-----
03.01.00.SG   Comp(3)

```

```
Switch#
```

With Dynamic Image Version Compatibility (DIVC), Dynamic(0) is stored instead of Incomp(1), Base(2), or Comp(3). Compatibility is determined during runtime when two different DIVC-capable images are running in the active and standby supervisor engines during ISSU.

For Catalyst 4500 switches, a value of Dynamic(0) in the stored compatibility-matrix normally results in Base(2) or Comp(3) upon run-time negotiation between the two software images. You never observe Incomp(1) as long as the other image name is present in the stored compatibility matrix.

This example shows how to display negotiated information regarding non-IOSd clients:

```

Switch# show package compatibility
PackageName      PeerPackageName      ModuleName      Compatibility
-----
rp_base          rp_base              aaa             COMPATIBLE
rp_base          rp_base              aaaccommon      COMPATIBLE
rp_base          rp_base              access_policy   COMPATIBLE
rp_base          rp_base              app_sess        COMPATIBLE
rp_base          rp_base              app_sess_ios    COMPATIBLE
rp_base          rp_base              auth_mgr        COMPATIBLE
.....
.....

```

Related Documents

Related Topic	Document Title
Performing ISSU	<i>Cisco IOS Software: Guide to Performing In Service Software Upgrades</i>
Information about Cisco Nonstop Forwarding	<i>Cisco Nonstop Forwarding</i> http://www.cisco.com/en/US/docs/ios/12_2s/feature/guide/fsnsf20s.html
Information about Stateful Switchover	<i>Stateful Switchover</i> http://www.cisco.com/en/US/docs/ios/12_0s/feature/guide/sso120s.html
ISSU and MPLS clients	ISSU MPLS Clients



Configuring the Cisco IOS XE In Service Software Upgrade Process



Note

An ISSU upgrade from any release prior to IOS XE 3.6.0E to 3.6.0E or later is unsupported for IOS XE supervisor engines (Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, and 7-E).



Note

Refer to [Chapter 5, “Configuring Virtual Switching Systems”](#) for details on VSS ISSU.



Note

ISSU is available in Cisco IOS XE Release 3.1.0.SG and later releases.

Operating on redundant systems, the In Service Software Upgrade (ISSU) process allows Cisco IOS XE software to be updated or otherwise modified while packet forwarding continues. In most networks, planned software upgrades are a significant cause of downtime. ISSU allows Cisco IOS XE software to be upgraded while packet forwarding continues. This increases network availability and reduces downtime caused by planned software upgrades. This document provides information about ISSU concepts and describes the steps taken to perform ISSU in a system.

Topics include:

- [Prerequisites to Performing ISSU, page 8-2](#)
- [About Performing ISSU, page 8-3](#)
- [How to Perform the ISSU Process, page 8-16](#)
- [Cisco High Availability Features in Cisco IOS XE 3.1.0SG, page 8-41](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

Related Documents

Related Topic	Document Title
Performing ISSU	<i>Cisco IOS Software: Guide to Performing In Service Software Upgrades</i>
Information about Cisco Nonstop Forwarding	<i>Cisco Nonstop Forwarding</i> http://www.cisco.com/en/US/docs/ios/12_2s/feature/guide/fsnsf20s.html
Information about Stateful Switchover	<i>Stateful Switchover</i> http://www.cisco.com/en/US/docs/ios/12_0s/feature/guide/sso120s.html

Prerequisites to Performing ISSU

Before performing ISSU, you must meet these prerequisites:

- A permanent “ISSU barrier” exists between pre-IOS XE 3.6.0E and IOS XE 3.6.0 releases: ISSU is supported between versions on the same side of the barrier but it is not supported between versions on opposite sides.



Note

This restriction applies to Catalyst 4500X in a VSS, as well as to Supervisor Engine 7E, Supervisor Engine 7LE, and Supervisor Engine 8E in a VSS or in a redundant chassis.

Four scenarios will illustrate the restriction:

If you are running a release prior to IOS XE 3.6.0E (3.5.1E, for example), you cannot perform an ISSU upgrade to IOS XE 3.6.0E.

If you are running IOS XE 3.6.0E, you cannot perform an ISSU downgrade to IOS XE 3.5.0E.

If you are running IOS XE 3.6.0E, you can perform an ISSU upgrade to IOS XE 3.6.1E (when released).

If you are running a release after IOS XE 3.6.0E (for example, 3.7.0, when released), you cannot perform an ISSU downgrade to IOS XE 3.5.0E.

- The type of the existing and target image must match. You cannot upgrade from a Universal Lite image to a Universal image (and vice versa) without experiencing several minutes of traffic loss. The same restriction applies between crypto and non-crypto images.
- The active and the standby supervisor engines must have the same supervisor engine hardware (same model, same memory, and so on).
- The new and old Cisco IOS XE software images must be loaded into the file systems (bootflash, SD card, or USB) of both the active and the standby supervisor engines before you begin the ISSU process.

The old software image should be available either in bootflash, SD card, or USB and the system should have been booted from one of these locations because the boot variable must be changed before the ISSU process starts.



Note **auto-boot** must be enabled for ISSU to succeed.

- Stateful Switchover (SSO) must be configured and the standby supervisor engine should be in STANDBY HOT state.

These commands indicate whether SSO is enabled: **show module**, **show running-config**, **show redundancy state**.

If you do not have SSO enabled, see the *Stateful Switchover* document for further information on how to enable and configure SSO.

- Nonstop Forwarding (NSF) must be configured and working properly. If you do not have NSF enabled, see the *Cisco Nonstop Forwarding* document for further information on how to enable and configure NSF.
- Before you perform ISSU, ensure that the file system for both the active and the standby supervisor engines contains the new ISSU-compatible IOS XE software. The current Cisco IOS XE version running in the system must also support ISSU.

You can enter various commands on the Catalyst 4500 series switch to determine supervisor engine versioning and Cisco IOS XE software compatibility. Alternatively, you can use the ISSU application on Cisco Feature Navigator to determine this.

- If you enter the **no ip routing** command, ISSU falls back from SSO to RPR mode, resulting in traffic loss.
- Autoboot is turned on and the current booted image matches the one specified in the BOOT environmental variable. For details on how to configure and verify these, please refer to "[Modifying the Boot Field and Using the boot Command](#), page 3-28.
- If you enter the **no ip routing** command, ISSU falls back from SSO to RPR mode, resulting in traffic loss.

About Performing ISSU



Note Do not make any hardware changes while performing ISSU.

Before you perform ISSU, you should understand the following concepts:

- [Stateful Switchover](#), page 8-4
- [NSF](#), page 8-6
- [ISSU Process](#), page 8-7
- [Performing an ISSU Upgrade: 2 Methods](#), page 8-12
- [Changeversion Process](#), page 8-13
- [Guidelines for Performing ISSU](#), page 8-14
- [Compatibility Matrix](#), page 8-14
- [SNMP Support for ISSU](#), page 8-15
- [Compatibility Verification Using Cisco Feature Navigator](#), page 8-15

Stateful Switchover

Development of the SSO feature is an incremental step within an overall program to improve the availability of networks constructed with Cisco IOS XE switches.

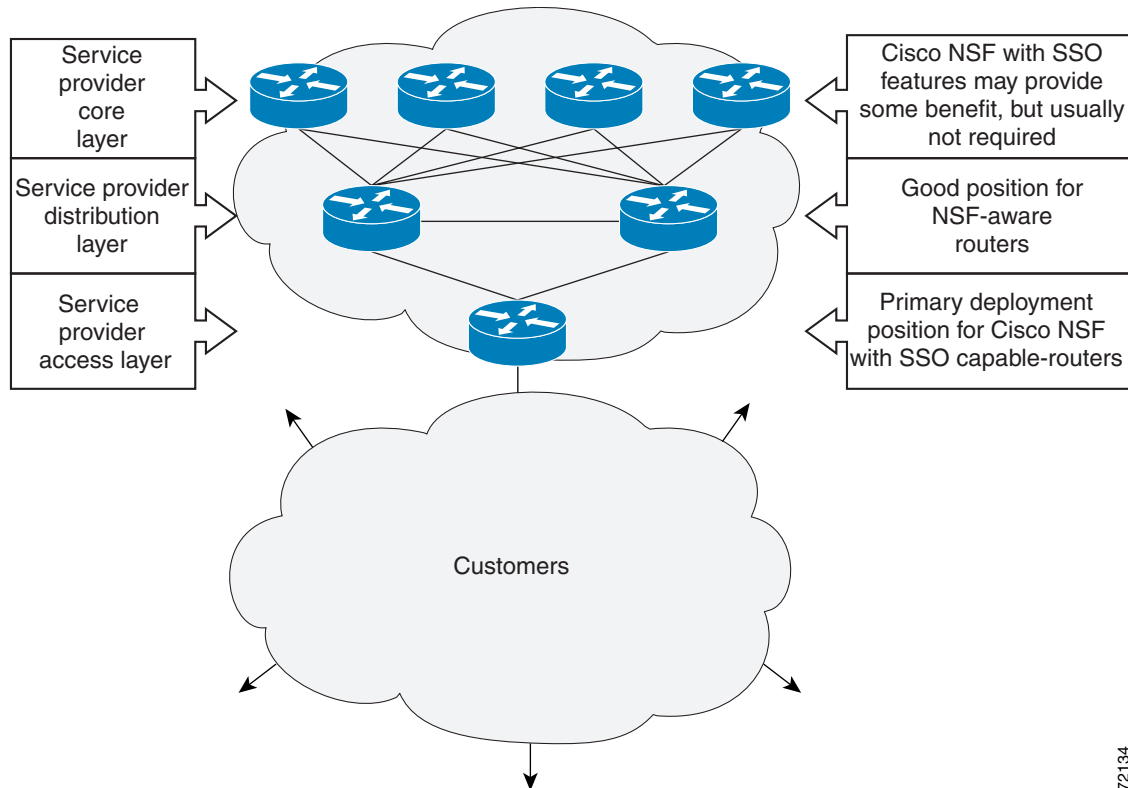
In specific Cisco networking devices that support dual supervisor engines, SSO takes advantage of supervisor engine redundancy to increase network availability. SSO achieves this by establishing one of the supervisor engines as the active processor while the other supervisor engine is designated as the standby processor. Following an initial synchronization between the two supervisor engines, SSO dynamically synchronizes supervisor engine state information between them in real-time.

A switchover from the active to the standby supervisor engine occurs when the active supervisor engine fails or is removed from the networking device.

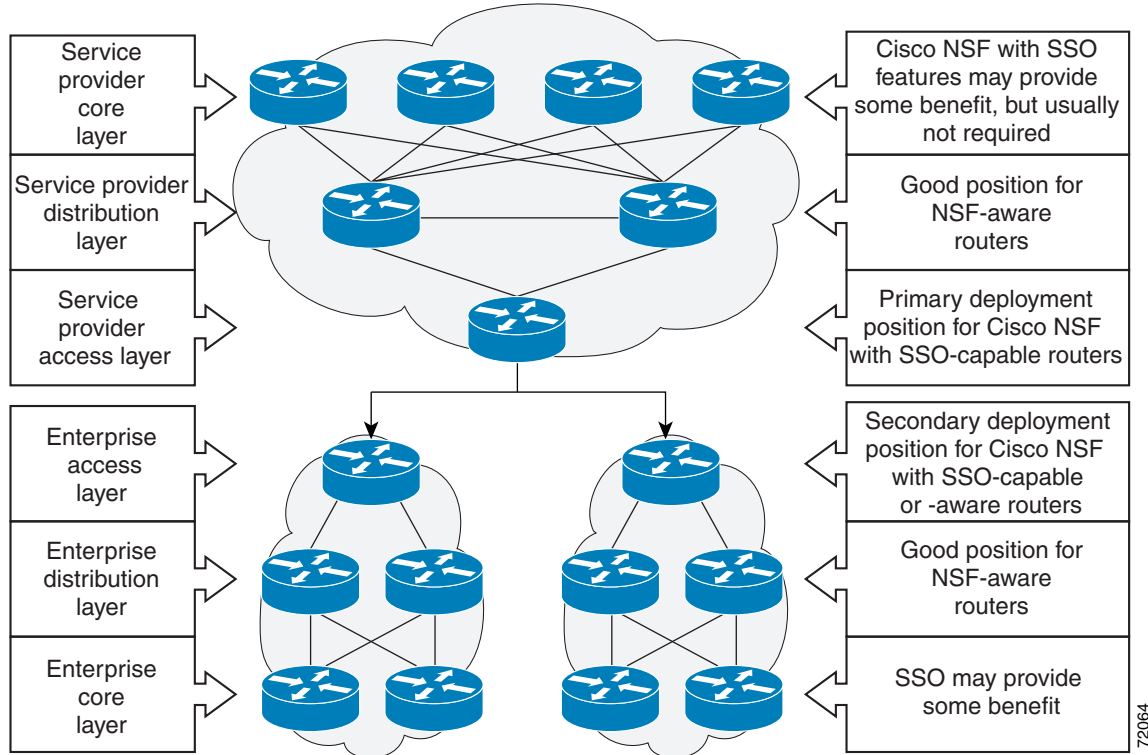
Cisco NSF is used with SSO. Cisco NSF allows the forwarding of data packets to continue along known routes while the routing protocol information is being restored following a switchover. With Cisco NSF, peer networking devices do not experience routing flaps, which reduce loss of service outages for customers.

[Figure 8-1](#) illustrates how SSO is typically deployed in service provider networks. In this example, Cisco NSF with SSO is enabled at the access layer (edge) of the service provider network. A fault at this point could result in loss of service for enterprise customers requiring access to the service provider network.

For Cisco NSF protocols that require neighboring devices to participate in Cisco NSF, Cisco NSF-aware software images must be installed on those neighboring distribution layer devices. Depending on your objectives, you may decide to deploy Cisco NSF and SSO features at the core layer of your network. Doing this can help reduce the time required to restore network capacity and service for certain failures, which leads to additional availability.

Figure 8-1 Cisco NSF with SSO Network Deployment: Service Provider Networks

Additional levels of availability may be gained by deploying Cisco NSF with SSO at other points in the network where a single point of failure exists. [Figure 8-2](#) illustrates an optional deployment strategy that applies Cisco NSF with SSO at the enterprise network access layer. In this example, each access point in the enterprise network represents another single point of failure in the network design. In the event of a switchover or a planned software upgrade, enterprise customer sessions would continue uninterrupted through the network in this example.

Figure 8-2 Cisco NSF with SSO Network Deployment: Enterprise Networks

For further information on SSO, see the *Stateful Switchover* document.

NSF

Cisco NSF works with the SSO feature in Cisco IOS XE software. SSO is a prerequisite of Cisco NSF. NSF works with SSO to minimize the amount of time a network is unavailable to its users following a switchover. The main objective of Cisco NSF is to continue forwarding IP packets following a supervisor engine switchover.

Usually, when a networking device restarts, all routing peers of that device detect that the device went down and then came back up. This transition results in what is called a routing flap, which could spread across multiple routing domains. Routing flaps caused by routing restarts create routing instabilities, which are detrimental to the overall network performance. Cisco NSF helps to suppress routing flaps in SSO-enabled devices, thus reducing network instability.

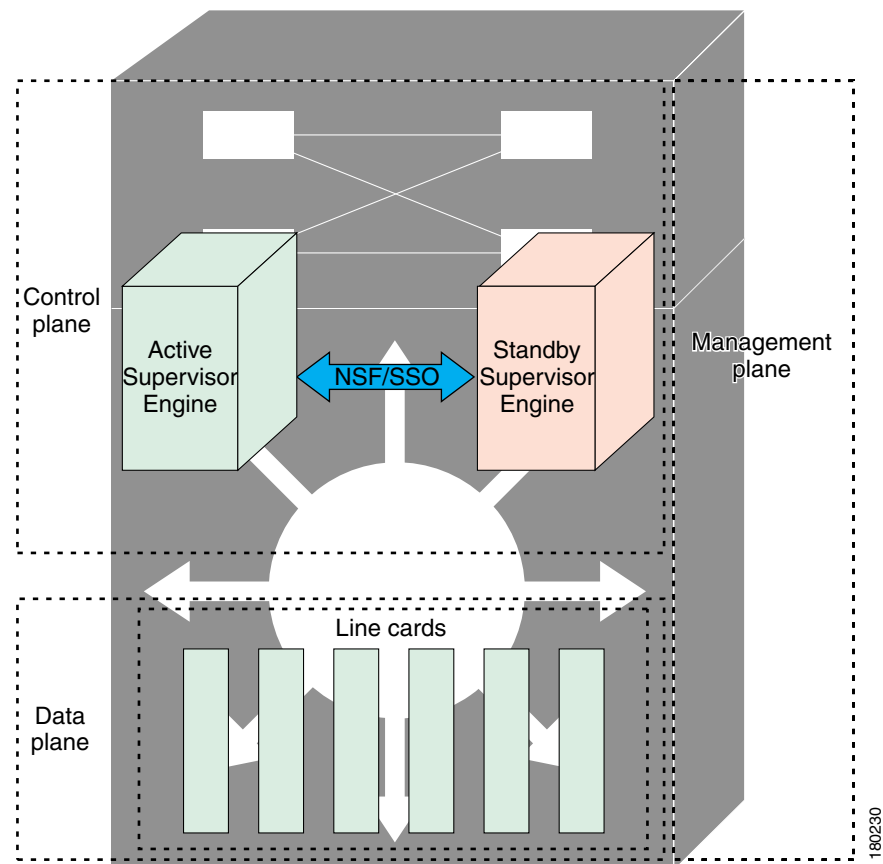
Cisco NSF allows for the forwarding of data packets to continue along known routes while the routing protocol information is being restored following a switchover. With Cisco NSF, peer networking devices do not experience routing flaps. Data traffic is forwarded while the standby supervisor engine assumes control from the failed active supervisor engine during a switchover. The ability of physical links to remain up through a switchover and to be kept current with the Forwarding Information Base (FIB) on the active supervisor engine is key to Cisco NSF operation.

ISSU Process

The ISSU process allows you to perform a Cisco IOS XE software upgrade or downgrade while the system continues to forward packets. (For an illustration of the commands used during the ISSU process, refer to [Figure 8-8](#).) Cisco IOS XE ISSU takes advantage of the Cisco IOS XE high availability infrastructure—Cisco NSF with SSO and hardware redundancy—and eliminates downtime associated with software upgrades by allowing changes while the system remains in service (see [Figure 8-3](#)).

SSO and NSF mode support configuration and runtime state synchronization from the active to the standby supervisor engine. For this process, the IOS XE software image on both the active and the standby supervisor engines must be the same. When images on active and standby supervisor engines are different, ISSU allows the two supervisor engines to be kept in synchronization even when these two versions of Cisco IOS XE support different sets of features and commands.

Figure 8-3 High Availability Features and Hardware Redundancy in the ISSU Process

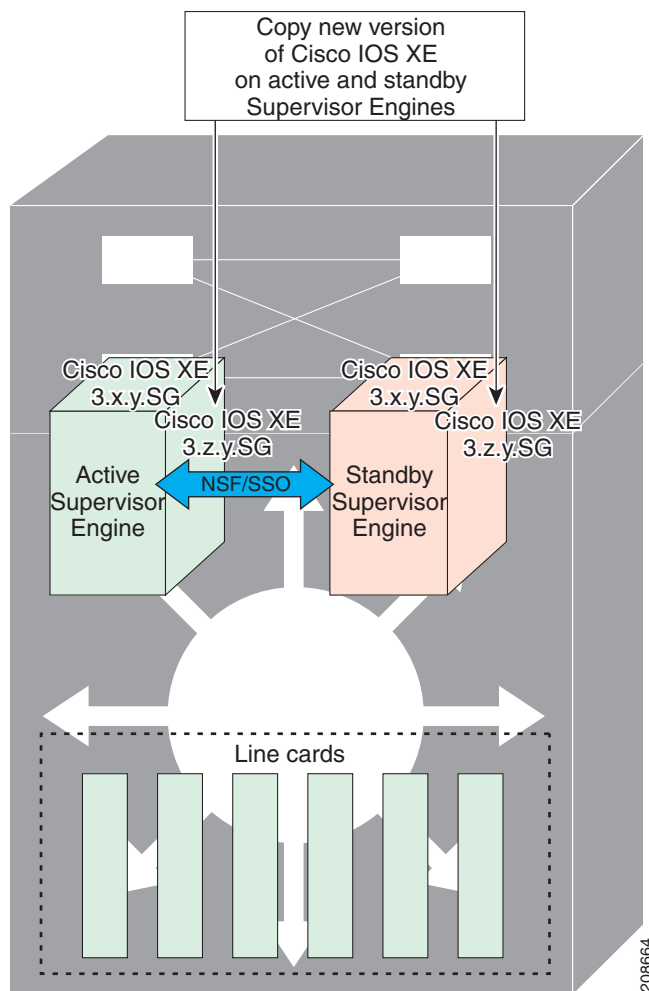


An ISSU-capable switch consists of two supervisor engines (active and standby) and 200 or more linecards. Before initiating the ISSU process, copy the Cisco IOS XE software into the file systems of both supervisor engines (see [Figure 8-4](#)).

**Note**

In the following figure, Cisco IOS XE 3.x.y SG represents the *current* version of Cisco IOS XE 3.z.y SG represents the image you are migrating to.

Figure 8-4 Copy New Version of Cisco IOS XE Software on Both Supervisor Engines

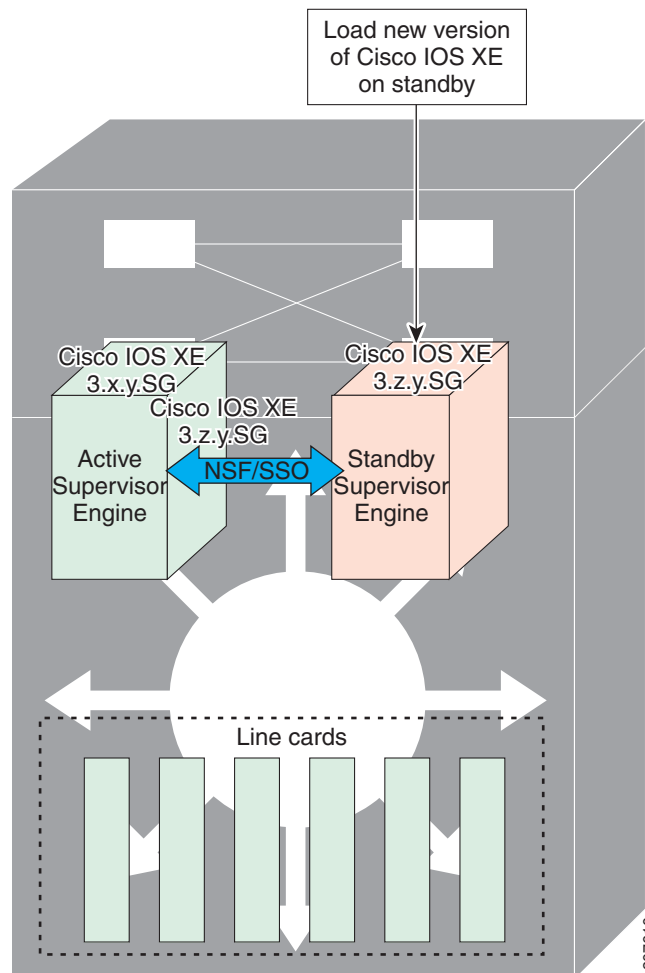


After you have copied the Cisco IOS XE software to both file systems, load the new version of Cisco IOS XE software onto the standby supervisor engine (see [Figure 8-5](#)).

**Note**

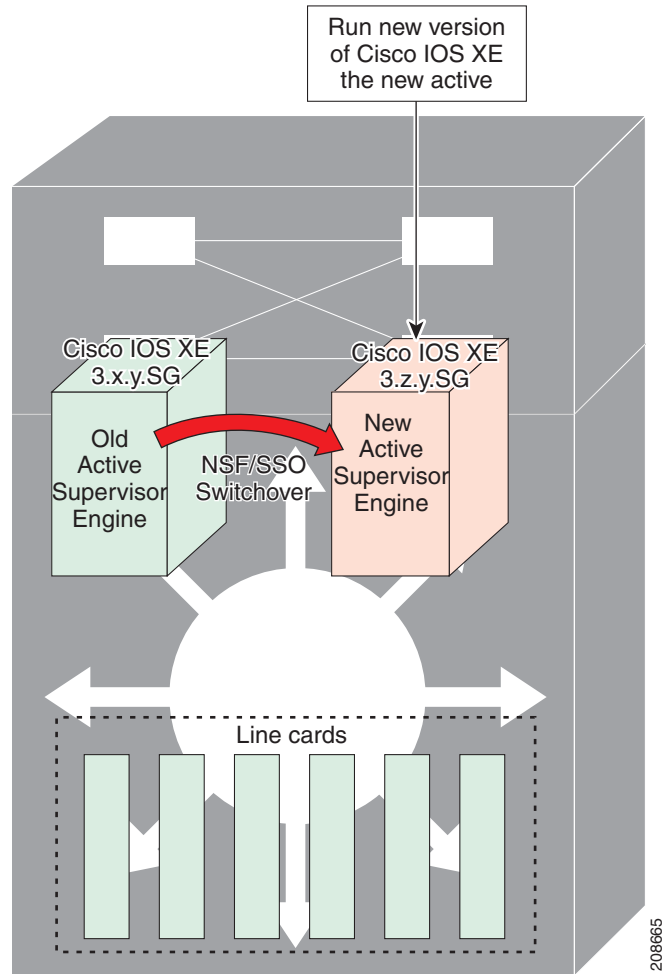
Without the ISSU feature, SSO/NSF cannot function between the active and standby supervisor engines when they are running different versions of the Cisco IOS XE image.

Figure 8-5 Load New Version of Cisco IOS XE Software on the Standby Supervisor Engine



After a switchover (NSF/SSO, not RPR), the standby supervisor engine takes over as the new active supervisor engine (see [Figure 8-6](#)).

Figure 8-6 **Switch Over to Standby Supervisor Engine**



The former active supervisor engine is loaded with an old Cisco IOS XE image so that if the new active supervisor engine experiences problems, you can abort and conduct a switchover to the former active, which is already running the old software image. Next, the former active supervisor engine is loaded with the new version of Cisco IOS XE software and becomes the new standby supervisor engine (see [Figure 8-7](#)).

Figure 8-7 Load New Standby Supervisor Engine with New Cisco IOS XE Software

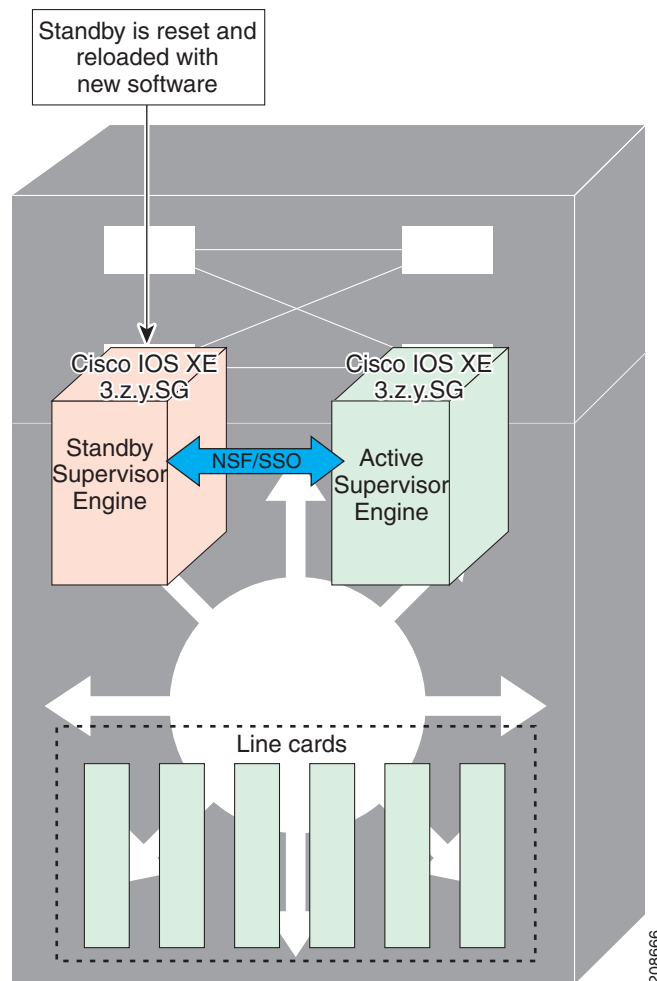
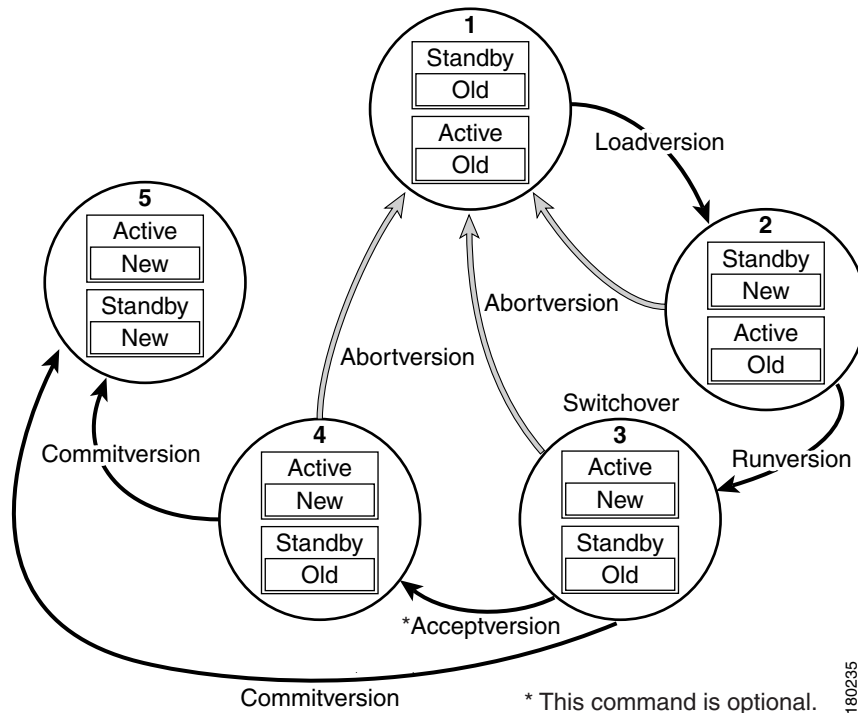


Figure 8-8 shows the steps during the ISSU process.

Figure 8-8 Steps During the ISSU Process



Note **Accept version** stops the rollback timer.

Performing an ISSU Upgrade: 2 Methods

There are two ways to perform an ISSU upgrade: manually, with four commands; or automatically, with one command.

The normal ISSU upgrade process involves issuing four separate ISSU exec commands (**issu loadversion**, **issu runversion**, **issu acceptversion**, **issu commitversion**) along with additional show command invocations to evaluate the success of each command before proceeding. Although the ISSU process is complicated, you should not expect disruption of service. The use of multiple ISSU commands dictates an additional level of care to ensure no service disruption. However, in some scenarios, this upgrade procedure might be cumbersome and of minimal value. A typical example is during a network upgrade that involves performing an ISSU upgrade on a large number of Catalyst 4500 switches. In these cases, we recommend that you first perform the normal (four command) ISSU upgrade procedure on one switch (possibly in a lab environment) to verify successful upgrade. Then, use a single **issu changeversion** command to perform an automatic ISSU on the rest of the Catalyst 4500 switches in the network.

Changeversion Process

The **issu changeversion** command launches a single-step complete ISSU upgrade cycle. It performs the logic for all four of the standard commands (**issu loadversion**, **issu runversion**, **issu acceptversion**, and **issu commitversion**) without user intervention, streamlining the upgrade through a single CLI step.

Additionally, **issu changeversion** allows the upgrade process to be scheduled for a future time. This enables you to stage a number of systems to perform upgrades sequentially when a potential disruption would be least harmful.

After the standby supervisor engine initializes and the system reaches a terminal state (RPR/SSO), the upgrade process is complete and the BOOT variable is permanently written with the new IOS XE software image. Hence, a reset on any RP will keep the system booting the new software image. Console and syslog messages will be generated to notify anyone monitoring the upgrade that the state transition has occurred.

Similar to the normal ISSU upgrade procedure, the in-progress upgrade procedure initiated by the **issu changeversion** command can be aborted with the **issu abortversion** command. If the system detects any problems or detects an unhealthy system during an upgrade, the upgrade might be automatically aborted.

When the **issu runversion** command is entered during the four step manual upgrade process, if any incompatible ISSU clients exist, the upgrade process reports them and their side effects, and allows the user to abort the upgrade. While performing a single-step upgrade process, when the process reaches the runversion state, it will either automatically continue with the upgrade provided the base clients are compatible, or automatically abort because of client incompatibility. If the user wants to continue the upgrade procedure in RPR mode, the user must use the normal ISSU command set and specify the **force** option when entering the **issu loadversion** command.

Changeversion: Quick Option (LV to INIT)

The **issu changeversion** command provides an optional quick command option that can reduce the time required to perform the automatic ISSU upgrade.

When the **quick** command option is applied, the ISSU upgrade state transition differs from that described previously. With this option, the software logic at the loadversion stage remains the same as previously described, and the logic that performs runversion and commitversion is combined. This logic skips the step in the upgrade procedure that loads the old software version on the new standby (old active) supervisor, instead, the new software image is uploaded on the new standby (old active).

This reduces the time required for the automatic ISSU upgrade by about a third (it saves one extra reload time).

Scheduled Changeversion: “in” and “at” Options

issu changeversion provides **in** and **at** command options that enable you to schedule a future automatic ISSU upgrade.

The **at** command option schedules an automatic ISSU upgrade to begin at a specific time. This option specifies an exact time (*hh:mm*, 24 hour format) in the next 24 hours at which the upgrade will occur.

The **in** command option schedules an automatic ISSU upgrade to begin after a certain amount of time has elapsed. This option specifies the number of hours and minutes (*hh:mm* format) that must elapse before an upgrade will occur, with a maximum value of 99:59.

Changeversion Deployment Scenario

The typical **issu changeversion** command usage scenario is for experienced users with a large installed base. These users typically validate a new image using a topology and configuration similar to their production network. The validation process should be done using both the existing multi-command process and the new **issu changeversion** command process. Once users certify an IOS XE software image and want to roll it out broadly, they can use the single command process to perform an efficient upgrade of their network.

Aborting an In-Progress Changeversion Procedure

The **issu changeversion** command functionality is designed to perform an ISSU software upgrade without user intervention. However, status messages are displayed to the console as the upgrade transitions through the various states. If any anomalies are noticed during the automatic upgrade, perhaps with peers or other parts of the network, you can use the **issu abortversion** command to manually abort the upgrade at any point in the process prior to the commitversion operation.

Guidelines for Performing ISSU

Be aware of the following guidelines while performing the ISSU process:

- Even with ISSU, it is recommended that upgrades be performed during a maintenance window.
- The new features should not be enabled (if they require change of configuration) during the ISSU process.



Note

Enabling them will cause the system to enter RPR mode because commands are only supported on the new version.

- In a downgrade scenario, if any feature is not available in the downgrade revision of the Cisco IOS XE software handle, that feature should be disabled prior to initiating the ISSU process.

Compatibility Matrix

ISSU requires additional information to determine compatibility between software versions. Therefore, a compatibility matrix is defined that contains information about other IOS XE software image with respect to the one in question.

This compatibility matrix represents the compatibility of two software versions, one running on the active and the other on the standby supervisor engine, and to allow the system to determine the highest operating mode it can achieve. Incompatible versions will not be able to progress to SSO operational mode.

The compatibility matrix represents the compatibility relationship a Cisco IOS XE software image has with all of the other Cisco IOS XE software versions within the designated support window (for example, all of those software versions the IOS XE software image “knows” about) and is populated and released with every IOS XE software image. The matrix stores compatibility information between its own release and prior releases. It is always the newest release that contains the latest information about compatibility with existing releases in the field. The compatibility matrix is available within the Cisco IOS XE software image and on Cisco.com so that users can determine in advance whether an upgrade can be done using the ISSU process.

You can perform the ISSU process when the old and new Cisco IOS XE software are compatible. The compatibility matrix information stores the compatibility among releases as follows:

- **Compatible**—The base-level system infrastructure and all optional HA-aware subsystems are compatible. An in-service upgrade or downgrade between these versions will succeed with minimal service impact. The matrix entry designates the images to be compatible (C).
- **Base-level compatible**—One or more of the optional HA-aware subsystems is not compatible. An in-service upgrade or downgrade between these versions will succeed; however, some subsystems will not be able to maintain state always during the transition from the old to the new version of Cisco IOS XE. The matrix entry designates the images to be base-level compatible (B).
- **Incompatible**—A core set of system infrastructure exists in Cisco IOS XE that must be able to interoperate in a stateful manner for SSO to function correctly. If any of these required features or subsystems is not interoperable, then the two versions of the Cisco IOS XE software image are declared to be incompatible. An in-service upgrade or downgrade between these versions is not possible. The matrix entry designates the images to be incompatible (I). The system operates in RPR mode during the upgrade process when the versions of Cisco IOS XE at the active and standby supervisor engines are incompatible.
- Cisco IOS XE determines the compatibility between the active and the standby IOS XE software dynamically during STANDBY boot up. The matrix is represented by “x”.

To display the compatibility matrix data between two software versions on a given system, enter the **show issu comp-matrix stored** command.

**Note**

This command is useful *only for verification purposes* because it is available *only after* the ISSU process has started. You might want to check the compatibility matrix prior to starting ISSU. Use the Feature Navigator to obtain the needed information:

<http://tools.cisco.com/ITDIT/CFN/jsp/index.jsp>

SNMP Support for ISSU

SNMP for SSO provides a mechanism for synchronizing the SNMP configurations and the MIBs that support SSO from the active supervisor engine to the standby supervisor engine.

Compatibility Verification Using Cisco Feature Navigator

The ISSU application on Cisco Feature Navigator allows you to:

- Select a specific software bundle.
- Identify which software images are compatible with the selected software image.

- Compare two IOS XE software images and understand the compatibility level of the software images (that is, compatible, base-level compatible, and incompatible), or dynamically determined.
- Compare two software images and see the client compatibility for each ISSU client.
- Provide links to release notes for the software image.

How to Perform the ISSU Process

Unlike SSO, which is a mode of operation for the device and a prerequisite for performing ISSU, the ISSU process is a series of steps performed while the switch is in operation. The steps result in an upgrade to new or modified Cisco IOS XE software, and have a minimal impact to traffic.



Note

For an illustration of the process flow for ISSU, refer to [Figure 8-8 on page 8-12](#).

This section includes the following topics:

- [Upgrading ISSU to Cisco IOS XE 3.4.0SG/15.1\(2\)SG from a Prior Release, page 8-16](#)
- [Downgrading ISSU from Cisco IOS XE 3.4.0SG/15.1\(2\)SG to a Prior Release, page 8-18](#)
- [Verifying the ISSU Software Installation, page 8-19](#)
- [Loading New Cisco IOS XE Software on the Standby Supervisor Engine, page 8-21](#) (required)
- [Switching to the Standby Supervisor Engine, page 8-25](#) (required)
- [Stopping the ISSU Rollback Timer \(Optional\), page 8-27](#) (optional)
- [Loading New Cisco IOS XE Software on the New Standby Supervisor Engine, page 8-28](#) (required)
- [Using changeversion to Automate an ISSU Upgrade, page 8-30](#)
- [Aborting a Software Upgrade During ISSU, page 8-36](#)
- [Configuring the Rollback Timer to Safeguard Against Upgrade Issues, page 8-37](#)
- [Displaying ISSU Compatibility Matrix Information, page 8-39](#)

Upgrading ISSU to Cisco IOS XE 3.4.0SG/15.1(2)SG from a Prior Release

Because images prior to Cisco IOS XE 3.4.0SG/15.1(2)SG use the earlier CLI format and Cisco IOS XE 3.4.0SG and 15.1(2)SG images use a newer CLI format, your upgrade consists of the following:

- Upgrading the image on your switch to Cisco IOS XE 3.4.0SG/15.1(2)SG.
- Upgrading mgmtVrf from the earlier CLI format to the later format, removing any IPv6 addresses on the interface.
- Enabling IPv6 address family under mgmtVrf, and reconfigure IPv6 addresses on fa1.

A configuration like the following should exist on pre-Cisco IOS XE 3.4.0SG/15.1(2)SG image:

```
ip vrf mgmtVrf
!
interface FastEthernet1
 ip vrf forwarding mgmtVrf
 ip address 10.1.1.1 255.255.255.0
 speed auto
 duplex auto
 ipv6 address 2000::1/64
```

!

Step 1 Perform an ISSU upgrade to a Cisco IOS XE 3.4.0SG/15.1(2)SG image.

Step 2 Run the VRF upgrade command.

```
Switch# config t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# vrf upgrade-cli multi-af-mode common-policies vrf mgmtVrf
You are about to upgrade to the multi-AF VRF syntax commands.
You will lose any IPv6 address configured on interfaces
belonging to upgraded VRFs.

Are you sure ? [yes]:
Number of VRFs upgraded: 1
Switch(config)# exit
```

Your configuration will appear as follows:

```
vrf definition mgmtVrf
!
 address-family ipv4
 exit-address-family
!
interface FastEthernet1
 vrf forwarding mgmtVrf
 ip address 10.1.1.1 255.255.255.0
 speed auto
 duplex auto
!
```

Step 3 Configure the switch to enable the IPv6 address family and add the IPv6 address.

```
Switch# config t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# vrf definition mgmtVrf
Switch(config-vrf)# address-family ipv6
Switch(config-vrf-af)# exit
Switch(config-vrf)# exit
Switch(config)# interface fa1
Switch(config-if)# ipv6 address 2000::1/64
Switch(config-if)# end
```

Your configuration will appear as follows.

```
vrf definition mgmtVrf
!
 address-family ipv4
 exit-address-family
!
 address-family ipv6
 exit-address-family
!
interface FastEthernet1
 vrf forwarding mgmtVrf
 ip address 10.1.1.1 255.255.255.0
 speed auto
 duplex auto
 ipv6 address 2000::1/64
```

Downgrading ISSU from Cisco IOS XE 3.4.0SG/15.1(2)SG to a Prior Release

Because a Cisco IOS XE 3.4.0SG/15.1(2)SG image uses a new CLI format and prior images use earlier CLI formats, the downgrade procedure include the following:

- Downgrading mgmtVrf from new CLI format to older CLI format, removing any IPv6 addresses on the interface.
- Downgrading the image on your switch to a prior release.
- Reconfiguring the IPv6 addresses on fa1.

A configuration like the following will appear on a switch running a Cisco IOS XE 3.4.0SG/15.1(2)SG image:

```
vrf definition mgmtVrf
!
 address-family ipv4
 exit-address-family
!
 address-family ipv6
 exit-address-family
!
interface FastEthernet1
 vrf forwarding mgmtVrf
 ip address 10.1.1.1 255.255.255.0
 speed auto
 duplex auto
 ipv6 address 2000::1/64
!
```

Step 1 Perform a downgrade to a release prior to Cisco IOS XE 3.4.0SG/15.1(2)SG.

```
Switch# config t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# no vrf upgrade-cli multi-af-mode common-policies vrf mgmtVrf
You are about to downgrade to the single-AF VRF syntax commands.
You will lose any IPv6 address configured on interfaces
belonging to downgraded VRFs.

Are you sure ? [yes]:
% ipv6 addresses from all interfaces in VRF mgmtVrf have been removed
Number of VRFs downgraded: 1
Switch(config)#
```

Your configuration will appear as follows:

```
ip vrf mgmtVrf
!
interface FastEthernet1
 ip vrf forwarding mgmtVrf
 ip address 10.1.1.1 255.255.255.0
 speed auto
 duplex auto
!
```

Step 2 Perform an ISSU downgrade to a pre-Cisco IOS XE 3.4.0SG/15.1(2)SGn image.

Step 3 Reconfigure the IPv6 address.

```
Switch# config t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface fa1
Switch(config-if)# ipv6 address 2000::1/64
```



```
Switch(config-if)# end
Switch#
```

Your configuration will appear as follows.

```
ip vrf mgmtVrf
!
interface FastEthernet1
 ip vrf forwarding mgmtVrf
 ip address 10.1.1.1 255.255.255.0
 speed auto
 duplex auto
 ipv6 address 2000::1/64
```

Verifying the ISSU Software Installation

During the ISSU process, there are five valid states: disabled, init, load version, run version, and system reset. Use the **show issu state** command to obtain the current ISSU state:

- Disabled state—The state for the standby supervisor engine while this engine is resetting.
- Init state—The initial state for two supervisor engines, one active and one standby, before the ISSU process is started. It is also the final state after the ISSU process completes.
- Load version (LV) state—The standby supervisor engine is loaded with the new version of Cisco IOS XE software.
- Run version (RV) state—The **issu runversion** command forces the switchover of the supervisor engines. The newly active supervisor engine runs the new Cisco IOS XE software image.
- While running ISSU, if both supervisor engines are reset due to power outage, for example, the ISSU context is lost and returns to the init state. Both supervisor engines return to the old software.

You can verify the ISSU software upgrade by entering **show** commands to provide information on the state of the during the ISSU process:

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode.
		Enter your password if prompted.
Step 2	Switch# show issu state [detail]	Displays the state of the during the ISSU process.
Step 3	Switch# show redundancy	Displays current or historical status, mode, and related redundancy information about the device.

This example shows how to display the state and the current status of the supervisor engine during the ISSU process:

```
Switch> enable
Switch# show issu state
Switch# show redundancy
```

Verifying Redundancy Mode Before Beginning the ISSU Process

Before you begin the ISSU process, verify the redundancy mode for the system and be sure to configure NSF and SSO.

The following example displays verification that the system is in SSO mode, that slot 3 is the active supervisor engine, and that slot 4 is the standby supervisor engine. Both supervisor engines are running the same Cisco IOS XE software release.

```
Switch# show redundancy state
    my state = 13 -ACTIVE
    peer state = 8 -STANDBY HOT
        Mode = Duplex
        Unit = Primary
        Unit ID = 5

Redundancy Mode (Operational) = Stateful Switchover
Redundancy Mode (Configured) = Stateful Switchover
    Redundancy State = Stateful Switchover
        Manual Swact = enabled

Communications = Up

    client count = 81
    client_notification_TMR = 240000 milliseconds
        keep_alive TMR = 9000 milliseconds
        keep_alive count = 1
    keep_alive threshold = 9
        RF debug mask = 0

Switch# show redundancy
Redundant System Information :

-----
    Available system uptime = 9 hours, 0 minute
Switchovers system experienced = 2
    Standby failures = 1
    Last switchover reason = user_forced

    Hardware Mode = Duplex
    Configured Redundancy Mode = Stateful Switchover
    Operating Redundancy Mode = Stateful Switchover
    Maintenance Mode = Disabled
    Communications = Up

Current Processor Information :
-----
    Active Location = slot 5
    Current Software state = ACTIVE
    Uptime in current state = 7 hours, 31 minutes
        Image Version = Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3
Switch Software (cat4500e-UNIVERSALK9-M), Version 03.06.05a.E RELEASE SOFTWARE (fc1)
    Technical Support: http://www.cisco.com/techsupport
    Copyright (c) 1986-2016 by Cisco Systems, Inc.
    Compiled Wed 12-Oct-16 02:37 by pro
        BOOT =
bootflash:cat4500e-universalk9.SPA.03.06.05a.E.152-2.E5a.bin,12;bootflash:cat4500e-univers
alk9.SPA.03.08.03.E.152-4.E3.bin,12;
    Configuration register = 0x2102

Peer Processor Information :
-----
    Standby Location = slot 6
    Current Software state = STANDBY HOT
    Uptime in current state = 6 hours, 39 minutes
        Image Version = Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3
Switch Software (cat4500e-UNIVERSALK9-M), Version 03.06.05a.E RELEASE SOFTWARE (fc1)
    Technical Support: http://www.cisco.com/techsupport
    Copyright (c) 1986-2016 by Cisco Systems, Inc.
```

```
Compiled Wed 12-Oct-16 02:37 by p
BOOT =
bootflash:cat4500e-universalk9.SPA.03.06.05a.E.152-2.E5a.bin,12;bootflash:cat4500e-univers
alk9.SPA.03.08.03.E.152-4.E3.bin,12;
Configuration register = 0x2102
```

Verifying the ISSU State Before Beginning the ISSU Process

Ensure that the active and standby supervisor engines are up and in ISSU Init state and that both supervisor engines are running the same current image.

The following example displays the ISSU state before the process begins:

```
Switch# show issu state detail

      Slot = 5
      RP State = Active
      ISSU State = Init
      Operating Mode = Stateful Switchover
      Current Image = bootflash:old_image
      Pre-ISSU (Original) Image = N/A
      Post-ISSU (Targeted) Image = N/A

      Slot = 6
      RP State = Standby
      ISSU State = Init
      Operating Mode = Stateful Switchover
      Current Image = bootflash:old_image
      Pre-ISSU (Original) Image = N/A
      Post-ISSU (Targeted) Image = N/A
```

The new version of the Cisco IOS XE software must be present on both of the supervisor engines. The directory information displayed for each of the supervisor engines shows that the new version is present.

```
Switch# dir bootflash:
Directory of bootflash:/

29177  -rw-   178623288  Nov 14 2016 00:27:19 -07:00  old_image.bin
29149  -rw-   190288756  Nov 14 2016 00:18:08 -07:00  new_image.bin

820875264 bytes total (450797568 bytes free)

Switch# dir slavebootflash:
Directory of slavebootflash:/

29194  -rw-   190288756  Nov 14 2016 00:31:21 -07:00  new_image.bin
29195  -rw-   178623288  Nov 14 2016 00:36:49 -07:00  old_image.bin

822910976 bytes total (66076672 bytes free)
```

Loading New Cisco IOS XE Software on the Standby Supervisor Engine

This task describes how to use ISSU to load a new version of Cisco IOS XE software to the standby supervisor engine.

Prerequisites

- Ensure that the new version of Cisco IOS XE software image is already present in the file system of both the active and standby supervisor engines. Also ensure that appropriate boot parameters (BOOT string and config-register) are set for the active and standby supervisor engines.



Note **auto-boot** must be enabled for ISSU to succeed.

- Optionally, perform additional tests and commands to determine the current state of peers and interfaces for later comparison.
- Ensure the system (both active and standby supervisor engines) is in SSO redundancy mode. If the system is in RPR mode, you can still upgrade the system using the ISSU CLI commands, but the system will experience extended packet loss during the upgrade.'

Refer to the *Stateful Switchover* document for more details on how to configure SSO mode on supervisor engines.

- For ISSU to function, the IOS XE file names on the active and standby supervisor engines must match.

Perform the following steps at the active supervisor engine:

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	Switch# issu loadversion [active-slot] active-image-new [standby-slot] standby-image-new [forced]	Starts the ISSU process and (optionally) overrides the automatic rollback when the new Cisco IOS XE software version is detected to be incompatible. It may take several minutes after entering the issu loadversion command for Cisco IOS XE software to load onto the standby supervisor engine and for the standby supervisor engine to transition to SSO mode. This causes the standby supervisor engine to reload with the new software image. If you use the forced option, the standby supervisor engine is booted with the new software image. After the software image is loaded on the standby supervisor engine, if the software image is incompatible, the system is forced to the RPR mode. Otherwise the system will continue in the SSO mode.

	Command or Action	Purpose
Step 3	Switch# show issu state [detail]	Displays the state of ISSU during the ISSU process. At this point in the ISSU process, use this command to check that the standby supervisor engine is loaded and is in SSO mode. It may take several minutes after entering the issu loadversion command for Cisco IOS XE software to load onto the standby supervisor engine and the standby supervisor engine to transition to SSO mode. If you enter the show issu state command too quickly, you may not see the information you need.
Step 4	Switch# show redundancy [states]	Displays redundancy facility state information.

This example shows how to start the ISSU process, boot the standby supervisor engine in the Standby Hot state, and load the standby supervisor engine slot (6) with the new IOS XE software image:

```
Switch> enable
Switch# issu loadversion 5 bootflash:new_image 6 slavebootflash:new_image
%issu loadversion executed successfully, Standby is being reloaded

Switch# show issu state detail
          Slot = 5
          RP State = Active
          ISSU State = Load Version
          Operating Mode = Stateful Switchover
          Current Image = bootflash:old_image.bin
          Pre-ISSU (Original) Image = bootflash:old_image.bin
          Post-ISSU (Targeted) Image = bootflash:new_image.bin

          Slot = 6
          RP State = Standby
          ISSU State = Load Version
          Operating Mode = Stateful Switchover
          Current Image = bootflash:new_image.bin
          Pre-ISSU (Original) Image = bootflash:old_image.bin
          Post-ISSU (Targeted) Image = bootflash:new_image.bin

Switch# show redundancy states
  my state = 13 -ACTIVE
  peer state = 8 -STANDBY HOT
    Mode = Duplex
    Unit = Primary
    Unit ID = 5

Redundancy Mode (Operational) = Stateful Switchover
Redundancy Mode (Configured) = Stateful Switchover
Redundancy State = Stateful Switchover
Manual Swact = enabled

Communications = Up

  client count = 81
  client_notification_TMR = 240000 milliseconds
    keep_alive TMR = 9000 milliseconds
      keep_alive count = 1
      keep_alive threshold = 9
      RF debug mask = 0
```

The following example shows how the forced option places the system in RPR mode:

```
Switch# show issu state detail
          Slot = 5
          RP State = Active
          ISSU State = Load Version
          Operating Mode = Stateful Switchover
          Current Image = bootflash:old_image.bin
          Pre-ISSU (Original) Image = bootflash:old_image.bin
          Post-ISSU (Targeted) Image = bootflash:new_image.bin

          Slot = 6
          RP State = Standby
          ISSU State = Load Version
          Operating Mode = Stateful Switchover
          Current Image = bootflash:new_image.bin
          Pre-ISSU (Original) Image = bootflash:old_image.bin
          Post-ISSU (Targeted) Image = bootflash:new_image.bin
```

The following example shows the redundancy mode as RPR:

```
Switch# show redundancy states
  my state = 13 -ACTIVE
  peer state = 4 -STANDBY COLD
    Mode = Duplex
    Unit = Primary
    Unit ID = 3

  Redundancy Mode (Operational) = RPR
  Redundancy Mode (Configured) = Stateful Switchover
    Redundancy State = RPR
    Manual Swact = enabled
    Communications = Up

  client count = 64
  client_notification_TMR = 240000 milliseconds
    keep_alive TMR = 9000 milliseconds
    keep_alive count = 1
  keep_alive threshold = 18
    RF debug mask = 0

Switch# show redundancy
Redundant System Information :

-----
  Available system uptime = 10 hours, 35 minutes
  Switchovers system experienced = 0
    Standby failures = 3
  Last switchover reason = none

    Hardware Mode = Duplex
  Configured Redundancy Mode = Stateful Switchover
  Operating Redundancy Mode = Stateful Switchover
    Maintenance Mode = Disabled
    Communications = Up

Current Processor Information :
-----
  Active Location = slot 5
  Current Software state = ACTIVE
  Uptime in current state = 10 hours, 34 minutes
```

```

Image Version = Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3
Switch Software (cat4500e-UNIVERSALK9-M), Version 03.06.05a.E RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2016 by Cisco Systems, Inc.
Compiled Wed 12-Oct-16 02:37 by pro
BOOT = bootflash:old_image.bin,12;
Configuration register = 0x2102

Peer Processor Information :
-----
Standby Location = slot 6
Current Software state = STANDBY HOT
Uptime in current state = 4 minutes
Image Version = Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3
Switch Software (cat4500e-UNIVERSALK9-M), Version 03.08.03.E RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2016 by Cisco Systems, Inc.
Compiled Sun 06-Nov-16 13:49 by pr
BOOT = bootflash:old_image.bin,12;
Configuration register = 0x2102

```

Switching to the Standby Supervisor Engine

This task describes how to switchover to the standby supervisor engine, which is running the new Cisco IOS XE software image. Perform the following steps at the active supervisor engine.

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	Switch# issu runversion [standby-slot] [standby-image-new]	Forces a switchover from the active to the standby supervisor engine and reloads the former active (current standby) supervisor engines with the old IOS XE image. When you enter the issu runversion command, an SSO switchover will be performed, and NSF procedures will be invoked if so configured.
Step 3	Switch# show issu state [detail]	Displays the state of the during the ISSU process. At this point in the ISSU process, use this command to check that a switchover occurs to slot 6.
Step 4	Switch# show redundancy [states]	Displays redundancy facility state information.

This example shows how to cause a switchover to the former standby supervisor engine (slot 6), reset the former active supervisor engine and reload it with the old IOS XE software image so it becomes the standby supervisor engine:

```

Switch> enable
Switch# issu runversion 6 slavebootflash:new_image
%issu runversion initiated successfully

```

A switchover happens at this point. At the new active supervisor engine, do the following after old active supervisor engine comes up as standby.

```

Switch# show issu state detail
Slot = 6

```

```

RP State = Active
ISSU State = Run Version
Operating Mode = Stateful Switchover
Current Image = bootflash:new_image.bin
Pre-ISSU (Original) Image = bootflash:old_image.bin
Post-ISSU (Targeted) Image = bootflash:new_image.bin

Slot = 5
RP State = Standby
ISSU State = Run Version
Operating Mode = Stateful Switchover
Current Image = bootflash:old_image.bin
Pre-ISSU (Original) Image = bootflash:old_image.bin
Post-ISSU (Targeted) Image = bootflash:new_image.bin

```

**Note**

The new active supervisor engine is now running the new version of software, and the standby supervisor engine is running the old version of software and is in the standby hot state.

```

Switch# show redundancy states
my state = 13 -ACTIVE
peer state = 8 -STANDBY HOT
Mode = Duplex
Unit = Primary
Unit ID = 6

Redundancy Mode (Operational) = Stateful Switchover
Redundancy Mode (Configured) = Stateful Switchover
Redundancy State = Stateful Switchover
Manual Swact = enabled

Communications = Up

client count = 88
client_notification_TMR = 240000 milliseconds
keep_alive TMR = 9000 milliseconds
keep_alive count = 0
keep_alive threshold = 9
RF debug mask = 0

```

Once Runversion has completed, the new active supervisor engine will be running the new version of software and the previously active supervisor engine will now become the standby supervisor engine. The standby will be reset and reloaded, but it will remain on the previous version of software and come back online in standbyhot status. The following example shows how to verify these conditions:

```

Switch# show redundancy
Redundant System Information :
-----
Available system uptime = 10 hours, 45 minutes
Switchovers system experienced = 1
Standby failures = 0
Last switchover reason = user_forced

Hardware Mode = Duplex
Configured Redundancy Mode = Stateful Switchover
Operating Redundancy Mode = Stateful Switchover
Maintenance Mode = Disabled
Communications = Up

Current Processor Information :
-----

```



```

        Active Location = slot 6
        Current Software state = ACTIVE
        Uptime in current state = 18 minutes
        Image Version = Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3
Switch Software (cat4500e-UNIVERSALK9-M), Version 03.08.03.E RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2016 by Cisco Systems, Inc.
Compiled Sun 06-Nov-16 13:49 by prod
        BOOT = bootflash:old_image.bin,12;
        Configuration register = 0x2102

Peer Processor Information :
-----
        Standby Location = slot 5
        Current Software state = STANDBY HOT
        Uptime in current state = 2 minutes
        Image Version = Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3
Switch Software (cat4500e-UNIVERSALK9-M), Version 03.06.05a.E RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2016 by Cisco Systems, Inc.
Compiled Wed 12-Oct-16 02:37 by p
        BOOT = bootflash:old_image.bin,12;
        Configuration register = 0x2102

```

Stopping the ISSU Rollback Timer (Optional)

This optional task describes how to stop the rollback timer.

If you do not run the following procedure before the rollback timer “timeout,” the system automatically aborts the ISSU process and reverts to the original Cisco IOS XE software version. By default the rollback timer is 45 minutes.

Use the following information to decide what action you should take:

- If you want to retain your switch in the runversion state for an extended period, you need to stop the rollback timer by entering the **acceptversion** command. Then validate the new software and enter the **commitversion** command (as described in the following section).
- If you want to proceed with the commitversion operation within the rollback timer window of 45 minutes (or the configured value), you do not need to use the **acceptversion** command to stop the roll-back timer.



Note

The **issu acceptversion** command may be optionally executed after the **issu runversion** command.

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	Switch# issu acceptversion [<i>active-slot</i>] [<i>active-image-new</i>]	Halts the rollback timer and ensures the new Cisco IOS XE ISSU process is not automatically aborted during the ISSU process. Enter the issu acceptversion command within the time period specified by the rollback timer to acknowledge that the supervisor engine has achieved connectivity to the outside world; otherwise, the ISSU process is terminated, when the rollback timer expires, and the system reverts to the previous version of Cisco IOS XE software by switching to the standby supervisor engine.
Step 3	Switch# show issu rollback-timer	Displays the amount of time left before an automatic rollback will occur.

This example displays the Timer before you stop it. In the following example, the “Automatic Rollback Time” information indicates the amount of time remaining before an automatic rollback will occur.

```
Switch> enable
Switch# show issu rollback-timer
    Rollback Process State = In progress
    Configured Rollback Time = 00:45:00
    Automatic Rollback Time = 00:35:48

Switch# issu acceptversion 6 bootflash:new_image.bin
% Rollback timer stopped. Please issue the commitversion command.

Switch# show issu rollback-timer
    Rollback Process State = Not in progress
    Configured Rollback Time = 00:45:00
```

Loading New Cisco IOS XE Software on the New Standby Supervisor Engine

This task explains how to load new version of Cisco IOS XE software to the new standby supervisor engine.

Perform the following steps at the active supervisor engine:

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	Switch# issu commitversion [<i>standby-slot</i>] [<i>standby-image-new</i>]	Allows the new Cisco IOS XE software image to be loaded into the standby supervisor engine.

	Command or Action	Purpose
Step 3	Switch# show redundancy [states]	Displays redundancy facility state information.
Step 4	Switch# show issu state [detail]	Displays the state of the during the ISSU process. At this point in the ISSU process, use this command to check that the standby supervisor engine is loaded with the new image.

This example shows how to reset and reload the current standby supervisor engine (slot 1) with the new Cisco IOS XE software version. After you enter the **commitversion** command, the standby supervisor engine boots in the Standby Hot state.

```
Switch> enable
Switch# issu commitversion 5 slavebootflash:new_image
%issu commitversion executed successfully

Switch# show issu state
                        Slot = 6
                        RP State = Active
                        ISSU State = Init

                        Slot = 5
                        RP State = Standby
                        ISSU State = Init
```

Use the following to verify that the standby supervisor engine is reloaded with the new software.

```
Switch# show redundancy states
  my state = 13 -ACTIVE
  peer state = 8 -STANDBY HOT
    Mode = Duplex
    Unit = Primary
    Unit ID = 6

Redundancy Mode (Operational) = Stateful Switchover
Redundancy Mode (Configured) = Stateful Switchover
  Redundancy State = Stateful Switchover
    Manual Swact = enabled

Communications = Up

  client count = 88
  client_notification_TMR = 240000 milliseconds
    keep_alive TMR = 9000 milliseconds
    keep_alive count = 0
    keep_alive threshold = 9
    RF debug mask = 0

Switch# show redundancy
Redundant System Information :

-----
  Available system uptime = 10 hours, 56 minutes
  Switchovers system experienced = 1
    Standby failures = 1
    Last switchover reason = user_forced

    Hardware Mode = Duplex
  Configured Redundancy Mode = Stateful Switchover
  Operating Redundancy Mode = Stateful Switchover
  Maintenance Mode = Disabled
  Communications = Up
```

Current Processor Information :

```
-----
      Active Location = slot 6
      Current Software state = ACTIVE
      Uptime in current state = 29 minutes
      Image Version = Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3
Switch Software (cat4500e-UNIVERSALK9-M), Version 03.08.03.E RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2016 by Cisco Systems, Inc.
Compiled Sun 06-Nov-16 13:49 by prod
      BOOT = bootflash:new_image.bin,12;bootflash:old_image.bin,12;
      Configuration register = 0x2102
```

Peer Processor Information :

```
-----
      Standby Location = slot 5
      Current Software state = STANDBY HOT
      Uptime in current state = 1 minute
      Image Version = Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3
Switch Software (cat4500e-UNIVERSALK9-M), Version 03.08.03.E RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2016 by Cisco Systems, Inc.
Compiled Sun 06-Nov-16 13:49 by pr
      BOOT = bootflash:new_image.bin,12;bootflash:old_image.bin,12;
      Configuration register = 0x2102
```

Switch# **show issu state detail**

```
      Slot = 6
      RP State = Active
      ISSU State = Init
      Operating Mode = Stateful Switchover
      Current Image = bootflash:new_image
      Pre-ISSU (Original) Image = N/A
      Post-ISSU (Targeted) Image = N/A

      Slot = 5
      RP State = Standby
      ISSU State = Init
      Operating Mode = Stateful Switchover
      Current Image = bootflash:new_image
      Pre-ISSU (Original) Image = N/A
      Post-ISSU (Targeted) Image = N/A
```

The ISSU process has completed. At this stage, any further Cisco IOS XE software version upgrade or downgrade will require that a new ISSU process be invoked.

Using changeversion to Automate an ISSU Upgrade

This task describes how to use the **issu changeversion** command to perform a one step ISSU upgrade.

Prerequisites

- Ensure that the new version of Cisco IOS XE software image is already present in the file system of both the active and standby supervisor engines. Also ensure that appropriate boot parameters (BOOT string and config-register) are set for the active and standby supervisor engines
- Optionally, perform additional tests and commands to determine the current state of peers and interfaces for later comparison.

- Ensure the system (both active and standby supervisor engines) is in SSO redundancy mode. If the system is in RPR mode, you can still upgrade the system using the ISSU CLI commands, but the system will experience extended packet loss during the upgrade.'

Refer to the Stateful Switchover document for more details on how to configure SSO mode on supervisor engines (refer to [Chapter 10, “Configuring Supervisor Engine Redundancy Using RPR and SSO on Supervisor Engine 6-E and Supervisor Engine 6L-E”](#)).

- For ISSU to function, the IOS XE software image file names on the active and standby supervisor engines must match.

Perform the following steps at the active supervisor engine:

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	Switch# issu changeversion [<i>active-slot active-image-new</i>] [<i>standby-slot standby-image-new</i>] [at <i>hh:mm</i> in <i>hh:mm</i>] [quick]	Initiates a single-step complete upgrade process cycle. Performs the logic of the four standard commands (issu loadversion, issu runversion, issu acceptversion, and issu commitversion) without user intervention. <i>active-slot</i> —Defines the active slot number. <i>new-image</i> —Specifies IOS XE image URL to be upgraded to. <i>standby-slot</i> —Defines the standby slot number. <i>standby-image</i> —Specifies the standby IOS XE image URL. at <i>hh:mm</i> —Schedules an ISSU upgrade to begin in the future. Provides an exact time (<i>hh:mm</i> , 24 hour format) in the next 24 hours when the upgrade will occur. in <i>hh:mm</i> —Schedules an ISSU upgrade to begin in the future. Provides the number of hours and minutes (<i>hh:mm</i> format) that will elapse before an upgrade will occur (99:59 max). quick —Upon switchover, boots the standby supervisor engine with the new, rather than old, image for faster upgrade.
Step 3	Switch# show issu state [detail]	Displays the state of the during the ISSU process. At this point in the ISSU process, use this command to check that the standby supervisor engine is loaded and is in SSO mode.
Step 4	Switch# show redundancy [states]	Displays redundancy facility state information.

This example shows how to initiate an ISSU upgrade process using the `issu changeversion` command on slot number 5, the slot for the current active supervisor engine. The `show issu state detail` and `show redundancy` command output is included to show the supervisor state before and after the upgrade procedure.



Note

The success messages included in the output below is displayed after some delay because the ISSU upgrade procedure progresses through the ISSU states.

```

Switch> enable
Switch# show issu state detail
                Slot = 5
                RP State = Active
                ISSU State = Init
                Operating Mode = Stateful Switchover
                Current Image = bootflash:x.bin
                Pre-ISSU (Original) Image = N/A
                Post-ISSU (Targeted) Image = N/A

                Slot = 6
                RP State = Standby
                ISSU State = Init
                Operating Mode = Stateful Switchover
                Current Image = bootflash:x.bin
                Pre-ISSU (Original) Image = N/A
                Post-ISSU (Targeted) Image = N/A

switch#sh redundancy
Redundant System Information :

-----
        Available system uptime = 10 hours, 58 minutes
Switchovers system experienced = 1
        Standby failures = 1
        Last switchover reason = user_forced

        Hardware Mode = Duplex
        Configured Redundancy Mode = Stateful Switchover
        Operating Redundancy Mode = Stateful Switchover
        Maintenance Mode = Disabled
        Communications = Up

Current Processor Information :
-----
        Active Location = slot 6
        Current Software state = ACTIVE
        Uptime in current state = 31 minutes
        Image Version = Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3
Switch Software (cat4500e-UNIVERSALK9-M), Version 03.08.03.E RELEASE SOFTWARE (fc2)
        Technical Support: http://www.cisco.com/techsupport
        Copyright (c) 1986-2016 by Cisco Systems, Inc.
        Compiled Sun 06-Nov-16 13:49 by prod
        BOOT = bootflash:new_image.bin,12;bootflash:old_image.bin,12;
        Configuration register = 0x2102

Peer Processor Information :
-----
        Standby Location = slot 5
        Current Software state = STANDBY HOT
        Uptime in current state = 3 minutes
        Image Version = Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3
Switch Software (cat4500e-UNIVERSALK9-M), Version 03.08.03.E RELEASE SOFTWARE (fc2)
        Technical Support: http://www.cisco.com/techsupport
        Copyright (c) 1986-2016 by Cisco Systems, Inc.
        Compiled Sun 06-Nov-16 13:49 by pr
        BOOT = bootflash:new_image.bin,12;bootflash:old_image.bin,12;
        Configuration register = 0x2102

Switch# issu changeversion bootflash:y.bin
% 'issu changeversion' is now executing 'issu loadversion'
% issu loadversion executed successfully, Standby is being reloaded

```

```
% changeversion finished executing loadversion, waiting for standby to reload and reach SSO ...
```



Note Standby reloads with target image.

```
.....
.....
```

```
*Feb 25 20:41:00.479: %INSTALLER-7-ISSU_OP_SUCC: issu changeversion is now executing 'issu runversion'
*Feb 25 20:41:03.639: %INSTALLER-7-ISSU_OP_SUCC: issu changeversion successfully executed 'issu runversion'
```



Note Switchover occurs.

```
.....
.....
```

Look at the console of new active supervisor engine.

```
*Feb 25 20:47:39.859: %RF-5-RF_TERMINAL_STATE: Terminal state reached for (SSO)
*Feb 25 20:47:39.971: %INSTALLER-7-ISSU_OP_SUCC: issu changeversion is now executing 'issu commitversion'
```

```
....
....
```



Note The new standby supervisor engine reloads with target image; changeversion is successful upon SSO terminal state is reached.

```
*Feb 25 20:54:16.092: %HA_CONFIG_SYNC-6-BULK_CFGSYNC_SUCCEED: Bulk Sync succeeded
*Feb 25 20:54:16.094: %RF-5-RF_TERMINAL_STATE: Terminal state reached for (SSO)
Switch#
```

Switch# **show issu state detail**

```

                Slot = 6
                RP State = Active
                ISSU State = Init
                Operating Mode = Stateful Switchover
                Current Image = bootflash:y.bin
                Pre-ISSU (Original) Image = N/A
                Post-ISSU (Targeted) Image = N/A
```

```

                Slot = 5
                RP State = Standby
                ISSU State = Init
                Operating Mode = Stateful Switchover
                Current Image = bootflash:y.bin
                Pre-ISSU (Original) Image = N/A
                Post-ISSU (Targeted) Image = N/A
```

Switch# **show redundancy**

Redundant System Information :

```
-----
    Available system uptime = 10 hours, 58 minutes
    Switchovers system experienced = 1
    Standby failures = 1
```

```

Last switchover reason = user_forced

      Hardware Mode = Duplex
Configured Redundancy Mode = Stateful Switchover
Operating Redundancy Mode = Stateful Switchover
      Maintenance Mode = Disabled
      Communications = Up

Current Processor Information :
-----
      Active Location = slot 6
      Current Software state = ACTIVE
      Uptime in current state = 31 minutes
      Image Version = Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3
Switch Software (cat4500e-UNIVERSALK9-M), Version 03.08.03.E RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2016 by Cisco Systems, Inc.
Compiled Sun 06-Nov-16 13:49 by prod
      BOOT = bootflash:new_image.bin,12;bootflash:old_image.bin,12;
      Configuration register = 0x2102

Peer Processor Information :
-----
      Standby Location = slot 5
      Current Software state = STANDBY HOT
      Uptime in current state = 3 minutes
      Image Version = Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3
Switch Software (cat4500e-UNIVERSALK9-M), Version 03.08.03.E RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2016 by Cisco Systems, Inc.
Compiled Sun 06-Nov-16 13:49 by pr
      BOOT = bootflash:new_image.bin,12;bootflash:old_image.bin,12;
      Configuration register = 0x2102

```

This example shows how to use `issu changeversion` with the `at` command option to schedule an ISSU upgrade procedure to automatically start at the specified time. This example specifies that the ISSU upgrade should be started at 16:30 (24 hour format). The **show issu state detail** and **show redundancy** command output is included to show the supervisor state before and after the **issu changeversion** command was entered.

```

Switch> enable
Switch# show issu state detail

      Slot = 5
      RP State = Active
      ISSU State = Init
      Operating Mode = Stateful Switchover
      Current Image = bootflash:x.bin
      Pre-ISSU (Original) Image = N/A
      Post-ISSU (Targeted) Image = N/A

      Slot = 6
      RP State = Standby
      ISSU State = Init
      Operating Mode = Stateful Switchover
      Current Image = bootflash:x.bin
      Pre-ISSU (Original) Image = N/A
      Post-ISSU (Targeted) Image = N/A

Switch# show redundancy
Redundant System Information :

-----
      Available system uptime = 10 hours, 58 minutes

```



```

Switchovers system experienced = 1
    Standby failures = 1
    Last switchover reason = user_forced

    Hardware Mode = Duplex
    Configured Redundancy Mode = Stateful Switchover
    Operating Redundancy Mode = Stateful Switchover
    Maintenance Mode = Disabled
    Communications = Up

Current Processor Information :
-----
    Active Location = slot 6
    Current Software state = ACTIVE
    Uptime in current state = 31 minutes
    Image Version = Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3
Switch Software (cat4500e-UNIVERSALK9-M), Version 03.08.03.E RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2016 by Cisco Systems, Inc.
Compiled Sun 06-Nov-16 13:49 by prod
    BOOT = bootflash:new_image.bin,12;bootflash:old_image.bin,12;
    Configuration register = 0x2102

Peer Processor Information :
-----
    Standby Location = slot 5
    Current Software state = STANDBY HOT
    Uptime in current state = 3 minutes
    Image Version = Cisco IOS Software, IOS-XE Software, Catalyst 4500 L3
Switch Software (cat4500e-UNIVERSALK9-M), Version 03.08.03.E RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2016 by Cisco Systems, Inc.
Compiled Sun 06-Nov-16 13:49 by pr
    BOOT = bootflash:new_image.bin,12;bootflash:old_image.bin,12;
    Configuration register = 0x2102

Switch# issu changeversion 5 bootflash:y.bin 6 slavebootflash:y at 16:30
% 'issu changeversion' was executed at [ Apr 12 16:27:43 ].
% The planned ISSU changeversion is to occur in (hh:mm:ss) [ 00:03:00 ] at [ Apr 12
16:30:43 ].
% Current system time: [ Apr 12 16:27:43 ]
% Planned upgrade image: bootflash:y.bin
% To cancel the planned upgrade, please execute 'issu abortversion'

Switch# show issu state detail

    Slot = 5
    RP State = Active
    ISSU State = Init
    Changeversion = TRUE
    Operating Mode = Stateful Switchover
    Current Image = bootflash:x.bin
    Pre-ISSU (Original) Image = N/A
    Post-ISSU (Targeted) Image = N/A

    Slot = 6
    RP State = Standby
    ISSU State = Init
    Changeversion = TRUE
    Operating Mode = Stateful Switchover
    Current Image = bootflash:x.bin
    Pre-ISSU (Original) Image = N/A
    Post-ISSU (Targeted) Image = N/A

```

Aborting a Software Upgrade During ISSU

You can abort the ISSU process at any stage manually (prior to entering the **issu commitversion** command) by entering the **issu abortversion** command. The **issu abortversion** command may also be issued after entering the **issu changeversion** command while the automatic ISSU upgrade is still in progress. The ISSU process also aborts on its own if the software detects a failure.



Note If you enter the **issu abortversion** command before the standby supervisor engine becomes hot, the traffic might be disrupted.

If you abort the process after you issue the **issu loadversion** command, the standby supervisor engine is reset and reloaded with the original software.

If the process is aborted after you enter either the **issu runversion** or **issu acceptversion** command, then a second switchover is performed to the new standby supervisor engine that is still running the original software version. The supervisor engine that had been running the new software is reset and reloaded with the original software version.



Note Ensure that the standby supervisor is fully booted *before* issuing the **abortversion** command on an active supervisor engine.

The following task describes how to abort the ISSU process before you complete the ISSU process with the **issu commitversion** command.

Perform the following task on the active supervisor engine.

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	Switch# issu abortversion [<i>active slot</i> [<i>active-image-new</i>]]	Cancels the ISSU upgrade or downgrade process in progress and restores the switch to its state before the process had started.

This example shows how to abort the ISSU process on slot number 6, the slot for the current active supervisor engine. In this example, the ISSU upgrade process is in the Runversion state when the **issu abortversion** command is entered:

```
Switch> enable
Switch# show issu state detail
      Slot = 6
      RP State = Active
      ISSU State = Run Version
      Operating Mode = Stateful Switchover
      Current Image = bootflash:x.bin
      Pre-ISSU (Original) Image = bootflash:y.bin
      Post-ISSU (Targeted) Image = bootflash:x.bin

      Slot = 5
      RP State = Standby
      ISSU State = Run Version
      Operating Mode = Stateful Switchover
      Current Image = bootflash:y.bin
      Pre-ISSU (Original) Image = bootflash:y.bin
```

```

Post-ISSU (Targeted) Image = bootflash:x.bin

Switch# issu abortversion 6
% issu abortversion initiated successfully
Switch# show issu state detail

          Slot = 5
          RP State = Active
          ISSU State = Init
          Operating Mode = Stateful Switchover
          Current Image = bootflash:y.bin
    Pre-ISSU (Original) Image = N/A
    Post-ISSU (Targeted) Image = N/A

          Slot = 6
          RP State = Standby
          ISSU State = Init
          Operating Mode = Stateful Switchover
          Current Image = bootflash:y.bin
    Pre-ISSU (Original) Image = N/A
    Post-ISSU (Targeted) Image = N/A

Switch#

```

Configuring the Rollback Timer to Safeguard Against Upgrade Issues

Cisco IOS XE software maintains an ISSU rollback timer, to safeguard against an upgrade that may leave the new active supervisor engine in a state in which communication with the standby supervisor engine is severed.

You may want to configure the rollback timer to fewer than 45 minutes (the default) so that the user need not wait in case the new software is not committed or the connection to the switch was lost while it was in runversion mode. A user may want to configure the rollback timer to more than 45 minutes in order to have enough time to verify the operation of the new Cisco IOS XE software before committing the new software image.



Note

The valid timer value range is from 0 to 7200 seconds (two hours). A value of 0 seconds disables the rollback timer.

Once you are satisfied that the new image at the active supervisor engine has been successful and you want to remain in the current state, you may indicate acceptance by issuing the **issu acceptversion** command, which stops the rollback timer.

Issuing the **issu commitversion** command at this stage is equal to entering both the **issu acceptversion** and the **issu commitversion** commands. Use the **issu commitversion** command if you do not intend to run in the current state for a period of time and are satisfied with the new software version.



Note

The rollback timer can be configured only in the ISSU Init state.

This task explains how to configure the rollback timer:

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	Switch# configure terminal	Enters global configuration mode.
Step 3	Switch(config)# issu set rollback-timer value	Configures the rollback timer value, which can range from 0 to 7200.
Step 4	Switch(config)# exit	Returns the user to privileged EXEC mode.
Step 5	Switch# show issu rollback-timer	Displays the current setting of the ISSU rollback timer.

This example shows how to set the rollback timer to 3600 seconds:

```
Switch> enable
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# issu set rollback-timer 3600
% Rollback timer value set to [ 3600 ] seconds

Switch(config)# exit

Switch# show issu rollback-timer
Rollback Process State = Not in progress
Configured Rollback Time = 60:00
```

The Rollback Timer cannot be set in loadversion or runversion state, as the following example illustrates:

```
Switch# show issu state detail
Slot = 5
RP State = Active
ISSU State = Load Version
Operating Mode = Stateful Switchover
Current Image = bootflash:old_image
Pre-ISSU (Original) Image = bootflash:old_image
Post-ISSU (Targeted) Image = bootflash:new_image

Slot = 6
RP State = Standby
ISSU State = Load Version
Operating Mode = Stateful Switchover
Current Image = bootflash:new_image
Pre-ISSU (Original) Image = bootflash:old_image
Post-ISSU (Targeted) Image = bootflash:new_image

Switch# show issu rollback-timer
Rollback Process State = Not in progress
Configured Rollback Time = 60:00

Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# issu set rollback-timer 20
% ISSU state should be [ init ] to set the rollback timer
```

Displaying ISSU Compatibility Matrix Information

The ISSU compatibility matrix contains information about other IOS XE software releases and the version in question. This compatibility matrix represents the compatibility of the two software versions, one running on the active and the other on the standby supervisor engine, and the matrix allows the system to determine the highest operating mode it can achieve. This information helps the user identify whether to use ISSU.

This task shows how to display information about the ISSU compatibility matrix.

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	Switch# show issu comp-matrix { negotiated stored xml }	Displays information regarding the ISSU compatibility matrix. <ul style="list-style-type: none"> • negotiated—Displays negotiated compatibility matrix information. • stored—Displays negotiated compatibility matrix information. • xml—Displays negotiated compatibility matrix information in XML format. <p>Note These commands display only the data within IOSd process. Use the show package compatibility to display the information for the whole system.</p>
Step 3	Switch# show package compatibility	Displays information regarding all client compatibility in the system.

This example shows how to display negotiated information regarding the compatibility matrix:

```
Switch> enable
Switch# show issu comp-matrix negotiated

CardType: WS-C4507R-E(182), Uid: 4, Image Ver: 03.00.00.1.68
Image Name: cat4500e-UNIVERSALK9-M

  Cid      Eid      Sid      pSid      pUid      Compatibility
  =====
  2         1      131078    3         3      COMPATIBLE
  3         1      131100    5         3      COMPATIBLE
  4         1      131123    9         3      COMPATIBLE
  .....
  .....

Message group summary:
  Cid      Eid      GrpId      Sid      pSid      pUid      Nego Result
  =====
  2         1         1      131078    3         3         Y
  3         1         1      131100    5         3         Y
  4         1         1      131123    9         3         Y
  .....
  .....

List of Clients:
```

```

Cid      Client Name      Base/Non-Base
=====
2        ISSU Proto client  Base
3        ISSU RF          Base
4        ISSU CF client   Base
.....
.....

```

This example shows how to display stored information regarding the compatibility matrix:

```
switch#show issu comp-matrix stored
```

```
Number of Matrices in Table = 1
```

```
(1) Matrix for cat4500e-UNIVERSALK9-M(182) - cat4500e-UNIVERSALK9-M(182)
```

```
=====
```

```
Start Flag (0xDEADBABE)
```

```

My Image ver: 03.06.05.E
Peer Version  Compatibility
-----
03.07.00.E    Dynamic(0)
03.06.02.E    Dynamic(0)
03.07.01.E    Dynamic(0)
03.07.02.E    Dynamic(0)
03.06.03.E    Dynamic(0)
03.08.00.E    Dynamic(0)
03.07.03.E    Dynamic(0)
03.08.01.E    Dynamic(0)
03.06.04.E    Dynamic(0)
03.06.05.E    Comp(3)

```

Dynamic Image Version Compatibility (DIVC) feature is supported in IOS XE releases. With DIVC, we store Dynamic(0) rather than Incomp(1), Base(2), or Comp(3), and determine compatibility during run-time when two different DIVC-capable IOS XE software images are running in the active and standby supervisor engines during ISSU.

For Catalyst 4500 switches, a value of Dynamic(0) in the stored compatibility-matrix normally results in Base(2) or Comp(3) upon run-time negotiation between the two IOS XE software images; as of today, you never observe Incomp(1) as long as the other IOS XE name is present in the stored compatibility-matrix.

This example shows how to display negotiated information regarding non-IOSd clients:

```
Switch# show package compatibility
```

```

PackageName      PeerPackageName      ModuleName      Compatibility
-----
rp_base          rp_base              aaa             COMPATIBLE
rp_base          rp_base              aaacommon      COMPATIBLE
rp_base          rp_base              access_policy  COMPATIBLE
rp_base          rp_base              app_sess       COMPATIBLE
rp_base          rp_base              app_sess_ios   COMPATIBLE
rp_base          rp_base              auth_mgr       COMPATIBLE
.....
.....

```

Cisco High Availability Features in Cisco IOS XE 3.1.0SG

This section provides a list of High Availability software features that are supported in Cisco IOS XE 3.1.0SG. Links to the feature documentation are included.

Feature guides may contain information about more than one feature. To find information about a specific feature within a feature guide, see the Feature Information table at the end of the guide.

Feature guides document features that are supported on many different software releases and platforms. Your Cisco software release or platform may not support all the features documented in a feature guide. See the Feature Information table at the end of the feature guide for information about which features in that guide are supported in your software release. Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

ISSU (IOS In-Service Software Upgrade)

http://www.cisco.com/en/US/products/ps7149/products_ios_protocol_group_home.html

Enhanced High System Availability

<http://www.cisco.com/en/US/docs/ios-xml/ios/ha/configuration/xe-3s/ha-config-stateful-switchover.html>



Configuring Interfaces

This chapter describes how to configure interfaces for the standalone and modular Catalyst 4500 series switches. It also provides guidelines, procedures, and configuration examples.

This chapter includes the following major sections:

- [Restrictions for Configuring Interfaces, page 9-2](#)
- [About Interface Configuration, page 9-2](#)
- [Using the interface Command, page 9-2](#)
- [Configuring a Range of Interfaces, page 9-5](#)
- [Using the Ethernet Management Port, page 9-7](#)
- [Defining and Using Interface-Range Macros, page 9-13](#)
- [Deploying SFP+ in X2 Ports, page 9-14](#)
- [Deploying 10-Gigabit Ethernet and Gigabit Ethernet SFP Ports on Supervisor Engine V-10GE, page 9-14](#)
- [Configuring MultiGigabit Ports on WS-X4748-12X48U+E, page 9-18](#)
- [Invoking Shared-Backplane Uplink Mode on Supervisor Engine 6-E and Supervisor Engine 6L-E, page 9-22](#)
- [Selecting Uplink Mode on a Supervisor Engine 6-E, page 9-23](#)
- [Selecting the Uplink Port on a Supervisor Engine 7L-E, page 9-24](#)
- [Selecting the Uplink Port on a Supervisor Engine 7L-E, page 9-24](#)
- [Configuring Supervisor Engine 7-E Uplink Mode on Supervisor Engine 8-E, page 9-26](#)
- [Supervisor Engine 9-E Uplink Configurations, page 9-29](#)
- [Digital Optical Monitoring Transceiver Support, page 9-32](#)
- [Digital Optical Monitoring Transceiver Support, page 9-32](#)
- [Configuring Optional Interface Features, page 9-32](#)
- [Understanding Online Insertion and Removal, page 9-47](#)
- [Online Insertion and Removal on a WS-4500X-32, page 9-48](#)
- [Monitoring and Maintaining the Interface, page 9-50](#)

**Note**

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

Restrictions for Configuring Interfaces

On a Cisco Catalyst 4500E Series Switches with Supervisor Engine 8-E, the management interface receive counter value is unreliable due to deficient functionality in the underlying CPU chip hardware. Fixing this issue requires a new revision of the CPU (when available).

About Interface Configuration

By default, all available interfaces, with the exception of the 40-Gigabit Ethernet (40-GE) interfaces, are enabled. The 10/100-Mbps Ethernet interfaces autonegotiate connection speed and duplex. The 10/100/1000-Mbps Ethernet interfaces negotiate speed, duplex, and flow control. The 1000-Mbps Ethernet interfaces negotiate flow control only. Autonegotiation automatically selects the fastest speed possible on that port for the given pair. If a speed is explicitly stated for an interface, that interface defaults to half duplex unless it is explicitly set for full duplex.

Many features are enabled on a per-interface basis. When you enter the **interface** command, you must specify the following:

- Interface type:
 - Fast Ethernet (keyword **fastethernet**)
 - Gigabit Ethernet (keyword **gigabitethernet**)
 - 10-Gigabit Ethernet (keyword **tengigabitethernet**).
 - 40-Gigabit Ethernet (keyword **fortygigabitethernet**)

Available on Cisco Catalyst 4500E Series Switches with Supervisor Engine 9-E. The corresponding uplink mode (**hw-module uplink mode 80Gig**) must be enabled to use this interface type. When enabled, the 10-GE uplink ports on the supervisor, are not available. See [Selecting the Uplink Mode on Supervisor Engine 9-E, page 9-31](#).

- Slot number—The slot in which the interface module is installed. Slots are numbered starting with 1, from top to bottom.
- Interface number—The interface number on the module. The interface numbers always begin with 1. When you are facing the front of the switch, the interfaces are numbered from left to right.

You can identify interfaces by physically checking the slot/interface location on the switch. You can also use the Cisco IOS **show** commands to display information about a specific interface or all the interfaces.

Using the interface Command

These general instructions apply to all interface configuration processes:

- Step 1** At the privileged EXEC prompt, enter the **configure terminal** command to enter global configuration mode:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#
```

- Step 2** In global configuration mode, enter the **interface** command. Identify the interface type and the number of the connector on the interface card. The following example shows how to select Fast Ethernet, slot 5, interface 1:

```
Switch(config)# interface fastethernet 5/1
Switch(config-if)#
```

- Step 3** Interface numbers are assigned at the factory at the time of installation or when modules are added to a system. Enter the **show interfaces** EXEC command to see a list of all interfaces installed on your switch. A report is provided for each interface that your switch supports, as shown in this display:

```
Switch(config-if)#Ctrl-Z
Switch#show interfaces
Vlan1 is up, line protocol is down
  Hardware is Ethernet SVI, address is 0004.dd46.7aff (bia 0004.dd46.7aff)
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    0 packets input, 0 bytes, 0 no buffer
    Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
    0 packets output, 0 bytes, 0 underruns
    0 output errors, 0 interface resets
    0 output buffer failures, 0 output buffers swapped out
GigabitEthernet1/1 is up, line protocol is down
  Hardware is Gigabit Ethernet Port, address is 0004.dd46.7700 (bia 0004.dd46.7700)
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Auto-duplex, Auto-speed
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/2000/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    0 packets input, 0 bytes, 0 no buffer
    Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
    0 input packets with dribble condition detected
    0 packets output, 0 bytes, 0 underruns
    0 output errors, 0 collisions, 0 interface resets
    0 babbles, 0 late collision, 0 deferred
    0 lost carrier, 0 no carrier
    0 output buffer failures, 0 output buffers swapped out
```

```
GigabitEthernet1/2 is up, line protocol is down
  Hardware is Gigabit Ethernet Port, address is 0004.dd46.7701 (bia 0004.dd46.7701)
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Auto-duplex, Auto-speed
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/2000/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    0 packets input, 0 bytes, 0 no buffer
    Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
    0 input packets with dribble condition detected
    0 packets output, 0 bytes, 0 underruns
    0 output errors, 0 collisions, 0 interface resets
    0 babbles, 0 late collision, 0 deferred
    0 lost carrier, 0 no carrier
    0 output buffer failures, 0 output buffers swapped out
--More--
<...output truncated...>
```

- Step 4** To begin configuring Fast Ethernet interface 5/5, as shown in the following example, enter the **interface** keyword, interface type, slot number, and interface number in global configuration mode:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface fastethernet 5/5
Switch(config-if)#
```



Note You do not need to add a space between the interface type and interface number. For example, in the preceding line you can specify either `fastethernet 5/5` or `fastethernet5/5`.

- Step 5** Follow each **interface** command with the interface configuration commands your particular interface requires. The commands you enter define the protocols and applications that run on the interface. The commands are collected and applied to the **interface** command until you enter another **interface** command or press **Ctrl-Z** to exit interface configuration mode and return to privileged EXEC mode.

- Step 6** After you configure an interface, check its status by using the EXEC **show** commands listed in the [“Monitoring and Maintaining the Interface” section on page 9-50](#).



Note Carrier-delay timer modification is not supported on Cisco Catalyst 4500 series switch. A default two second timer is applicable for interface transitions.

Configuring a Range of Interfaces

The interface-range configuration mode allows you to configure multiple interfaces with the same configuration parameters. When you enter the interface-range configuration mode, all command parameters you enter are attributed to all interfaces within that range until you exit interface-range configuration mode.

The following tables show how to configure a range of interfaces with the same configuration—in the default mode and in the 40-GE mode:

Command	Purpose
<pre>Switch(config)# interface range {vlan vlan_ID - vlan_ID} {{fastethernet gigabitethernet tengigabitethernet macro macro_name} slot/interface - interface} [, {vlan vlan_ID - vlan_ID} {{fastethernet gigabitethernet tengigabitethernet macro macro_name} slot/interface - interface}]</pre>	<p>Selects the range of interfaces to be configured. Note the following:</p> <ul style="list-style-type: none"> You are required to enter a space before the dash. You can enter up to five comma-separated ranges. You are not required to enter spaces before or after the comma.

Command	Purpose
<pre>Switch(config)# interface range {vlan vlan_ID - vlan_ID} {{fastethernet gigabitethernet tengigabitethernet fortygigabitethernet macro macro_name} slot/interface - interface} [, {vlan vlan_ID - vlan_ID} {{fastethernet gigabitethernet tengigabitethernet fortygigabitethernet macro macro_name} slot/interface - interface}]</pre>	<p>Selects the range of interfaces to be configured when the 40-Gigabit Ethernet mode is enabled on Supervisor Engine 9-E (hw-module uplink mode 80Gig). Note the following:</p> <ul style="list-style-type: none"> You are required to enter a space before the dash. You can enter up to five comma-separated ranges. You are not required to enter spaces before or after the comma. <p>Note In this mode, the 10-GE uplink ports on Supervisor Engine 9-E are not available, but if there are other 10-GE linecards in the chassis, the tengigabitethernet option is available.</p>



Note

When you use the **interface range** command, you must add a space between the **vlan**, **fastethernet**, **gigabitethernet**, **tengigabitethernet**, **fortygigabitethernet** or **macro** keyword and the dash. For example, the command **interface range fastethernet 5/1 - 5** specifies a valid range; the command **interface range fastethernet 5/1-5** does not contain a valid range command.

**Note**

The **interface range** command works only with VLAN interfaces that have been configured with the **interface vlan** command (the **show running-configuration** command displays the configured VLAN interfaces). VLAN interfaces that are not displayed by the **show running-configuration** command cannot be used with the **interface range** command.

This example shows how to reenable all Fast Ethernet interfaces 5/1 to 5/5:

```
Switch(config)# interface range fastethernet 5/1 - 5
Switch(config-if-range)# no shutdown
Switch(config-if-range)#
*Oct 6 08:24:35: %LINK-3-UPDOWN: Interface FastEthernet5/1, changed state to up
*Oct 6 08:24:35: %LINK-3-UPDOWN: Interface FastEthernet5/2, changed state to up
*Oct 6 08:24:35: %LINK-3-UPDOWN: Interface FastEthernet5/3, changed state to up
*Oct 6 08:24:35: %LINK-3-UPDOWN: Interface FastEthernet5/4, changed state to up
*Oct 6 08:24:35: %LINK-3-UPDOWN: Interface FastEthernet5/5, changed state to up
*Oct 6 08:24:36: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet5/
5, changed state to up
*Oct 6 08:24:36: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet5/
3, changed state to up
*Oct 6 08:24:36: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet5/
4, changed state to up
Switch(config-if)#
```

This example shows how to use a comma to add different interface type strings to the range to reenable all Fast Ethernet interfaces ranging from 5/1 to 5/5 and both Gigabit Ethernet interfaces 1/1 and 1/2:

```
Switch(config-if)# interface range fastethernet 5/1 - 5, gigabitethernet 1/1 - 2
Switch(config-if)# no shutdown
Switch(config-if)#
*Oct 6 08:29:28: %LINK-3-UPDOWN: Interface FastEthernet5/1, changed state to up
*Oct 6 08:29:28: %LINK-3-UPDOWN: Interface FastEthernet5/2, changed state to up
*Oct 6 08:29:28: %LINK-3-UPDOWN: Interface FastEthernet5/3, changed state to up
*Oct 6 08:29:28: %LINK-3-UPDOWN: Interface FastEthernet5/4, changed state to up
*Oct 6 08:29:28: %LINK-3-UPDOWN: Interface FastEthernet5/5, changed state to up
*Oct 6 08:29:28: %LINK-3-UPDOWN: Interface GigabitEthernet1/1, changed state to
up
*Oct 6 08:29:28: %LINK-3-UPDOWN: Interface GigabitEthernet1/2, changed state to
up
*Oct 6 08:29:29: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet5/
5, changed state to up
*Oct 6 08:29:29: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet5/
3, changed state to up
*Oct 6 08:29:29: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet5/
4, changed state to up
Switch(config-if)#
```

**Note**

If you enter multiple configuration commands while in interface-range configuration mode, each command is run as it is entered (they are not batched together and run after you exit interface-range configuration mode). If you exit interface-range configuration mode while the commands are being run, some commands might not be run on all interfaces in the range. Wait until the command prompt is displayed before exiting interface-range configuration mode.

Using the Ethernet Management Port

This section has this information:

- [Understanding the Ethernet Management Port, page 9-7](#)
- [Supported Features on the Ethernet Management Port, page 9-12](#)
- [Configuring the Ethernet Management Port, page 9-13](#)

Understanding the Ethernet Management Port

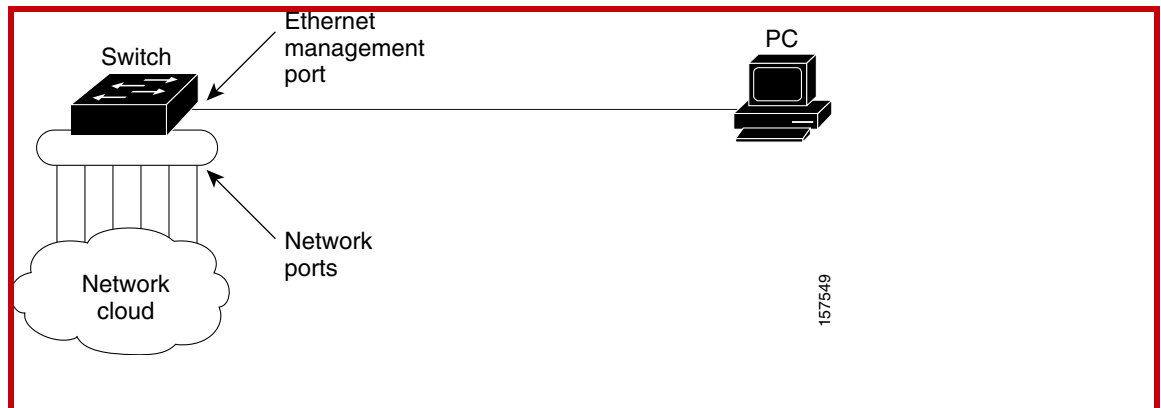
The Ethernet management port, also referred to as the *Fa1* or *fastethernet1* port, is a Layer 3 host port to which you can connect a PC. Use the Ethernet management port instead of the switch console port for network management. When managing a switch, connect the PC to the Ethernet management port on a Catalyst 4500 series switch. (Figure 9-1).



Note

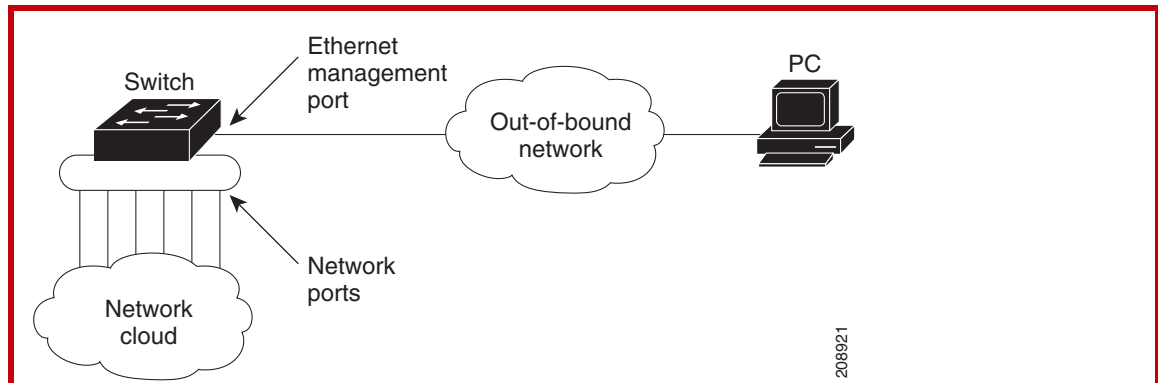
When connecting a PC to the Ethernet management port, you must assign an IP address.

Figure 9-1 Connecting a Switch to a PC



By default, the Ethernet management port is enabled. The switch cannot route packets from the Ethernet management port to a network port, and from the network port to the Ethernet port. To obtain these, the Fa1 interface is automatically placed in a separate routing domain (or VRF domain), called *mgmtVrf*. (You observe the *ip Vrf forwarding mgmtVrf* line in the running configuration when you boot up.) For details, read the “[Fa1 Interface and mgmtVrf](#)” section on page 9-8.

Even though the Ethernet management port does not support routing, you might need to enable routing protocols on the port. As illustrated in [Figure 9-2](#), you must enable routing protocols on the Ethernet management port when the PC is multiple hops away from the switch and the packets must pass through multiple Layer 3 devices to reach the PC.

Figure 9-2 Network with Routing Protocols Enabled

The specific implementation of Ethernet management port depends on the redundancy model you are applying.

For details on configuring SSO and ISSU, refer to [Chapter 11, “Configuring Supervisor Engine Redundancy Using RPR and SSO on Supervisor Engine 6-E and Supervisor Engine 6L-E”](#) and [Chapter 7, “Configuring the Cisco IOS In-Service Software Upgrade Process”](#).

Sections include:

- [Fa1 Interface and mgmtVrf, page 9-8](#)
- [SSO Model, page 9-11](#)
- [ISSU Model, page 9-12](#)

Fa1 Interface and mgmtVrf



Caution

The Ethernet management port is intended for out-of-band access only. Like the console port, the Ethernet management port has direct access to critical resources on the switch. Connecting this port to an in-band network might cause performance degradation and vulnerability to a denial of service attack.



Note

A service-policy that is applied to the control-plane interface is not applicable to traffic incoming on the management port.

All features that use fa1 now need to be VRF-aware.



Note

You cannot configure any other interface in the same routing domain and you cannot configure a different routing domain for the Fa1 interface.

On bootstrap the fa1 port assumes the following default configuration.

Images prior to Cisco IOS XE 3.4.0SG/15.1(2)SG use the old VRF definition format for management VRF as shown below.

```
!
ip vrf mgmtVrf
!
interface FastEthernet1
ip vrf forwarding mgmtVrf
```



```
speed auto
duplex auto
!
```

Images starting from Cisco IOS XE 3.4.0SG/15.1(2)SG use the new VRF definition format for management VRF as shown below.

```
!
vrf definition mgmtVrf
!
address-family ipv4
exit-address-family
!
address-family ipv6
exit-address-family
!
interface FastEthernet1
vrf forwarding mgmtVrf
speed auto
duplex auto
!
```

```
Switch# show ip vrf
```

Name	Default RD	Interfaces
mgmtVrf		Fa1

In Cisco IOS XE 3.10.1E/15.2(6)E1 and later releases, use the new VRF definition format for management VRF as shown below.

```
!
vrf definition mgmtVrf
!
address-family ipv4
exit-address-family
!
address-family ipv6
exit-address-family
!
interface FastEthernet1
vrf forwarding mgmtVrf
ip address address
ipv6 address address
speed auto
duplex auto
!
Switch# show ip vrf
Name Default RD Interfaces
mgmtVrf Fa1
```

Because the management port is placed in mgmtVrf, you should be aware of the VRF aware commands required for the following tasks:

- [Ping, page 9-10](#)
- [TraceRoute, page 9-10](#)
- [Telnet, page 9-10](#)
- [TFTP, page 9-11](#)
- [FTP, page 9-11](#)
- [SSH, page 9-11](#)

**Note**

Command usage specific to the mgmtVrf are mentioned below. The additional configuration that is necessary to make the feature work needs to be configured.

**Note**

The following features support IPv6 VRF

Ping

If you want to ping an IP address that is reachable through an fa1 port, enter the following command:

```
Switch# ping vrf mgmtVrf ip address
```

For example:

```
Switch# ping vrf mgmtVrf 20.20.20.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 20.20.20.1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

For Example:

```
IPv6 Management interface for Ping
Switch#ping vrf mgmtVrf 11.27.25.19
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 11.27.25.19, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
Switch#ping vrf mgmtVrf 2031::1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2031::1, timeout is 2 seconds
```

TraceRoute

```
Switch# traceroute vrf mgmtVrf ip address
```

For example:

Eg: Switch# **traceroute vrf mgmtVrf 20.20.20.1**

```
Type escape sequence to abort.
Tracing the route to 20.20.20.1
 1 20.20.20.1 0 msec 0 msec *
```

For example:

IPv6 Management interface for Traceroute

```
Switch#traceroute vrf mgmtVrf 2031::1
Type escape sequence to abort.
Tracing the route to 2031::1
2031::1 0 msec 0 msec 0 msec
```

Telnet

If you want to Telnet to a remote switch through the Fa1 port, enter the following command:

```
Switch# telnet word /vrf mgmtVrf
word IP address or hostname of a remote system
```

For example :

```
Switch# telnet 20.20.20.1 /vrf mgmtVrf
```

```
Trying 20.20.20.1 ... Open
User Access Verification
Password:
switch> en
Password:
switch#
```

For example:

```
IPv6 Management interface for Telnet
Switch# telnet 2031::1/vrf mgmtVrf
Trying 2031::1 ... Open
User Access Verification
Password:
switch> en
Password:
switch#
```

TFTP

If you want to use Fa1 port for TFTP operation, configure the Fa1 port as the source interface for TFTP as follows:

```
Switch(config)# ip tftp source-interface fastEthernet1
```

FTP

If you want to use an Fa1 port for an FTP operation, configure the Fa1 port as the source interface for FTP as follows:

```
Switch(config)# ip ftp source-interface fastEthernet1
```

SSH

If you want initiate SSH from your switch through the Fa1 port, enter the following command:

```
Switch# ssh -l login name -vrf mgmtVrf ip address
```

For example:

```
Switch# ssh -l xyz -vrf mgmtVrf 20.20.20.1
```

For example:

```
IPv6 Management Interface for SSH:
Switch# ssh -l xyz -vrf mgmtVrf 2031::1
```

SSO Model

On a redundant chassis, management port behavior differs from that of a standard Ethernet port in that each supervisor engine possesses a management port, and only the port on the active supervisor engine is enabled. The management port on the standby supervisor engine is always disabled; it cannot switch any kind of traffic.

When a switchover occurs, the management port of the standby supervisor engine (now, active) is enabled and can be used to switch traffic, while the management port on the “old” active supervisor engine is disabled.

**Note**

The Cisco IOS configuration for the management port is synchronized between the two supervisor engines. Under Cisco IOS, they possess the same IP address. To avoid address overlapping during a switchover on a redundant chassis, you should assign a different IP address on the management port from the one you assigned to the same port in the ROMMON configuration.

ISSU Model

In SSO mode, the running configurations on the active and standby supervisor engines must match. You cannot enable the management port on a redundant chassis if one of the two supervisor engines is running an Cisco IOS image prior to Cisco IOS Release 12.2(50)SG (wherein a management port is not supported).

When you perform an ISSU upgrade or downgrade between Cisco IOS Release 12.2(50)SG and a software image prior to Cisco IOS Release 12.2(50)SG, we automatically disable the management port. The port configuration is restored when both software images running on the supervisor engines are at least Cisco IOS Release 12.2(50)SG. A warning message is also displayed.

Supported Features on the Ethernet Management Port

The Ethernet management port supports these features:

- Express setup
- Network Assistant
- Telnet with passwords
- TFTP
- Secure Shell (SSH)
- DHCP-based autoconfiguration
- SNMP (only the ENTITY-MIB and the IF-MIB)
- IP ping
- Interface features
 - Speed—10 Mb/s, 100 Mb/s, 1000Mb/s, and autonegotiation
 - Duplex mode—Full, half, and autonegotiation
 - Loopback detection
- Cisco Discovery Protocol (CDP)
- IPv4 access control lists (ACLs)
- Routing protocols
- AAA

**Caution**

Before enabling a feature on the Ethernet management port, ensure that the feature is supported. If you try to configure an unsupported feature on an Ethernet management port, the feature might not work properly, and the switch might fail.

Configuring the Ethernet Management Port

To specify the Ethernet management port, enter **fastethernet1**.

To disable the port, use the **shutdown** interface configuration command. To enable the port, use the **no shutdown** interface configuration command.

To determine the link status to the PC, you can monitor the LED for the Ethernet management port:

- The LED is green (on) when the link is active.
- The LED is off when the link is down.
- The LED is amber when there is a POST failure.

To display the link status, use the **show interfaces fastethernet 1** privileged EXEC command.

Defining and Using Interface-Range Macros

You can define an interface-range macro to automatically select a range of interfaces for configuration. Before using the **macro** keyword in the **interface-range** macro command string, you must define the macro.

To define an interface-range macro, enter this command:

This example shows how to define an interface-range macro named **enet_list** to select Fast Ethernet

Command	Purpose
Switch(config)# define interface-range <i>macro_name</i> { vlan <i>vlan_ID</i> - <i>vlan_ID</i> } {{ fastethernet gigabitethernet tengigabitethernet } <i>slot/interface</i> - <i>interface</i> } [, { vlan <i>vlan_ID</i> - <i>vlan_ID</i> } {{ fastethernet gigabitethernet tengigabitethernet } <i>slot/interface</i> - <i>interface</i> }]	Defines the interface-range macro and saves it in the running configuration file.

interfaces 5/1 through 5/4:

```
Switch(config)# define interface-range enet_list fastethernet 5/1 - 4
```

To show the defined interface-range macro configuration, enter this command:

Command	Purpose
Switch# show running-config	Shows the defined interface-range macro configuration.

This example shows how to display the defined interface-range macro named **enet_list**:

```
Switch# show running-config | include define  
define interface-range enet_list FastEthernet5/1 - 4  
Switch#
```

To use an interface-range macro in the **interface range** command, enter this command:

Command	Purpose
Switch(config)# interface range macro <i>name</i>	Selects the interface range to be configured using the values saved in a named interface-range macro.

This example shows how to change to the interface-range configuration mode using the interface-range macro **enet_list**:

```
Switch(config)# interface range macro enet_list
Switch(config-if)#
```

Deploying SFP+ in X2 Ports



Note

This feature is supported on Supervisor Engine 6-E, 6L-E, and WS-X4606-X2-E.

To use an SFP+ in an X2 port to obtain 10-Gigabit Ethernet bandwidth, the Catalyst 4500 series switch supports OneX Convertor modules. When you plug a OneX Convertor module into an X2 port, it converts the X2 port into an SFP+ port into which you can plug in an SFP+. An SFP+ in a OneX Convertor module provides the same functionality as an X2 and maintains the same port numbering.

The output for the **show idprom tengigabitethernet slot/interface** command displays the contents of both the SFP+ and the OneX Convertor module EEPROMs when an SFP+ in a OneX Convertor module is plugged into an X2 port.

Deploying 10-Gigabit Ethernet and Gigabit Ethernet SFP Ports on Supervisor Engine V-10GE



Note

On a Catalyst 4510R series switch, if you enable both the 10-Gigabit Ethernet and Gigabit Ethernet SFP uplink ports, you must reboot the switch. On the Catalyst 4503, 4506, and 4507R series switches, this capability is automatically enabled.

Prior to Cisco IOS Release 12.2(25)SG, you could enable either the dual wire-speed 10-Gigabit Ethernet ports or four alternatively wired Gigabit Ethernet SFP uplink ports.

Beginning with Cisco IOS Release 12.2(25)SG, you could simultaneously deploy the dual 10-Gigabit Ethernet ports and the four Gigabit Ethernet SFP ports on the Catalyst 4503, Catalyst 4506, and Catalyst 4507R chassis.

When you deploy a Catalyst 4510R chassis, one of the following configurations is supported:

- Dual 10-Gigabit Ethernet ports (X2 optics) only.
- Four Gigabit Ethernet ports (SFP optics) only.

- Both the dual 10-Gigabit Ethernet and the four Gigabit Ethernet ports. The tenth slot (Flex-Slot) only supports a 2-port gigabit interface converter (GBIC) line card (WS-X4302-GB) when in this mode.
- You cannot place a line card with a backplane traffic capacity exceeding 6 Gbps in slots 8, 9, and 10 of a Catalyst 4510R-E chassis when used with a Supervisor Engine 6-E or 6L-E.

To select the 10-Gigabit Ethernet or the Gigabit Ethernet SFP uplink port, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Establishes global configuration mode.
Step 2	Switch(config)# hw-module uplink select [all gigabitethernet tengigabitethernet]	Selects the port type to enable.

The following example shows how to enable both 10-Gigabit Ethernet and Gigabit Ethernet SFP uplink ports on a Catalyst 4510R series switch:

```
Switch# configure terminal
Switch(config)# hw-module uplink select all
Warning: This configuration mode will place slot 10 in flex slot mode
```



Note

When you modify the uplink mode, you must reboot the switch.

Deploying 10-Gigabit Ethernet or Gigabit Ethernet Ports

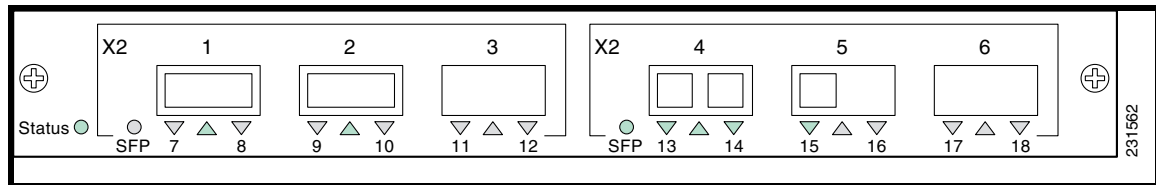
To increase the flexibility of X2 ports, Catalyst 4500 series switch support TwinGig Convertor modules. When you plug a TwinGig Convertor module into an X2 hole, it converts a single X2 hole (capable of holding one pluggable X2 optic) into two SFP holes (capable of holding two pluggable SFP optics). This enables you to have 10-Gigabit ports and 1-Gigabit ports on the same line card. It also allows you to use Gigabit ports, and then switch to a 10-Gigabit port, when needed.

This section includes these topics:

- [Port Numbering TwinGig Convertors, page 9-15](#)
- [Limitations on Using a TwinGig Convertor, page 9-16](#)
- [Selecting X2/TwinGig Convertor Model, page 9-16](#)

Port Numbering TwinGig Convertors

When a TwinGig Convertor is enabled or disabled, the number and type of ports on the line card change dynamically. The terminology must reflect this behavior. In Cisco IOS, 10-Gigabit ports are named *10-Gigabit* and 1-Gigabit ports are named *Gigabit*. Starting with Cisco IOS Release 12.2(40)SG, to avoid having two ports named 10-Gigabit1/1 and Gigabit1/1, the 10-Gigabit and 1-Gigabit port numbers are independent. For example, for a WS-X4606-10GE-E module with six X2 holes, the X2 ports are named *10-Gigabit slot-num/<1-6>*, and the SFP ports are named *Gigabit slot-num/<7-18>*.

Figure 9-3 Faceplate for WS-X4606-10GE

In Cisco IOS, ports 1 through 18 always exist. This means that you can apply configurations on them and they display in the CLI output. However, only the X2 or the SFP ports can be active at any particular time. For example, if an X2 is plugged into the second hole, the X2 port 2 is active and SFP ports 9 and 10 are inactive. If a TwinGig Converter is plugged into the second hole, the X2 port 2 is inactive, and the SFP ports 9 and 10 are active. The inactive ports are treated analogously to the inactive ports on Supervisor Engines IV and V-10GE, where at no time are all of the uplinks connected to the switching ASICs.

**Note**

When using both TwinGig and X2 transceivers on the WS-X4606-X2-E module, place ports 1-3 in one group and ports 4-6 in another. (The mode selected with the **show hw-module module port-group** command determines the behavior. See “[Selecting X2/TwinGig Converter Model](#)”.) Mixing within a port group does not work. For example, you cannot have an X2 in port 1 and a TwinGig in port 2 and expect both of them to function.

Limitations on Using a TwinGig Converter

Supervisor Engine 6-E, Supervisor Engine 6L-E, connect ports to the switching engine through a stub ASIC. This stub ASIC imposes some limitations on the ports: Gigabit and 10-Gigabit ports cannot be mixed on a single stub ASIC; they must either be all 10-Gigabit Ethernet (X2), or all Gigabit (TwinGig Converter and SFP). The faceplates of X2 modules show this stub port grouping, either with actual physical grouping with a box drawn around a grouping.

Selecting X2/TwinGig Converter Model

The default configuration mode is X2. If you plan to deploy 10-Gigabit Ethernet interfaces, you do not need to configure anything. However, if you want to deploy Gigabit interfaces (that is, use TwinGig Convertors), you must configure the associated port-group:

To determine how the X2 holes on a module are grouped, enter the **show hw-module module *m* port-group *p*** command.

**Note**

Place a 10-Gigabit Ethernet port that accepts CVR-X2-SFP into 1-Gigabit mode instead of 10-Gigabit Ethernet mode.

If you configure a 10-Gigabit Ethernet port as 1-Gigabit port, an output similar to the following appears:

```
Switch# show hw-module module 5 port-group
Module Port-group Active Inactive
-----
5      1      Gi5/3-6      Te5/1-2
```

If the port is set to the default, 10-Gigabit Ethernet mode, an output similar to the following appears:


```

Switch# show hw-module module 6 port-group
Module Port-group Active Inactive
-----
      6      1   Te6/1-2             Gi6/3-6

Switch# show int status mod 1

Port      Name      Status      Vlan      Duplex  Speed Type
Tel/1      Name      notconnect  1         full    10G  10GBase-LR
Tel/2      Name      connected   1         full    10G  10GBase-LR
Tel/3      Name      notconnect  1         full    10G  No X2
Tel/4      Name      notconnect  1         full    10G  No X2
Tel/5      Name      notconnect  1         full    10G  No X2
Tel/6      Name      notconnect  1         full    10G  No X2
Gi1/7      Name      inactive    1         full    1000 No Gbic
Gi1/8      Name      inactive    1         full    1000 No Gbic
Gi1/9      Name      inactive    1         full    1000 No Gbic
Gi1/10     Name      inactive    1         full    1000 No Gbic
Gi1/11     Name      inactive    1         full    1000 No Gbic
Gi1/12     Name      inactive    1         full    1000 No Gbic
Gi1/13     Name      inactive    1         full    1000 No Gbic
Gi1/14     Name      inactive    1         full    1000 No Gbic
Gi1/15     Name      inactive    1         full    1000 No Gbic
Gi1/16     Name      inactive    1         full    1000 No Gbic
Gi1/17     Name      inactive    1         full    1000 No Gbic
Gi1/18     Name      inactive    1         full    1000 No Gbic
Switch#

```

- To configure the modes of operation for each X2 port group in which you want to deploy Gigabit, enter the **hw-module module *m* port-group *p* select gigabitethernet** command. This configuration is preserved across power cycles and reloads.

To deploy Gigabit Ethernet interfaces using the TwinGig Converter, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Establishes global configuration mode.
Step 2	Switch(config)# hw-module module <i>m</i> port-group <i>p</i> select [gigabitethernet tengigabitethernet]	Selects the mode of operation for each X2 port-group. Default is 10-Gigabit Ethernet (x2).
Step 3	Switch(config)# exit	Exits configuration mode.
Step 4	Switch# show int status mod <i>n</i>	Verifies the setting.

This example shows how to select Gigabit Ethernet interfaces on a WS-X4606-10GE-E using the TwinGig Converter:

```

Switch# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# hw-module module 1 port-group 1 select gigabitethernet
Switch(config)# exit
Switch# show int status mod 1

Port      Name      Status      Vlan      Duplex  Speed Type
Tel/1      Name      inactive    1         full    10G  No X2
Tel/2      Name      inactive    1         full    10G  No X2
Tel/3      Name      inactive    1         full    10G  No X2
Tel/4      Name      notconnect  1         full    10G  No X2
Tel/5      Name      notconnect  1         full    10G  No X2
Tel/6      Name      notconnect  1         full    10G  No X2
Gi1/7      Name      notconnect  1         full    1000 No Gbic
Gi1/8      Name      notconnect  1         full    1000 No Gbic

```

Gi1/9	notconnect	1	full	1000	No	Gbic
Gi1/10	notconnect	1	full	1000	No	Gbic
Gi1/11	notconnect	1	full	1000	No	Gbic
Gi1/12	notconnect	1	full	1000	No	Gbic
Gi1/13	inactive	1	full	1000	No	Gbic
Gi1/14	inactive	1	full	1000	No	Gbic
Gi1/15	inactive	1	full	1000	No	Gbic
Gi1/16	inactive	1	full	1000	No	Gbic
Gi1/17	inactive	1	full	1000	No	Gbic
Gi1/18	inactive	1	full	1000	No	Gbic

Configuring MultiGigabit Ports on WS-X4748-12X48U+E

Cisco's Multigigabit Ethernet technology allows you to leverage 802.11ac Wave 2 speeds on your device. Beginning in Cisco IOS XE Release 3.7.1E, you can configure the WS-X4748-12X48U+E module to auto-negotiate multiple speeds on switch ports, and support 100 Mbps, 1 Gbps, 2.5 Gbps, and 5 Gbps speeds on Category 5e cables, and up to 10 Gbps over Category 6 and Category 6a cables. For more information on supported cables, see [Supported Cable Types and Speed, page 9-20](#).

Beginning in Cisco IOS XE 3.9.1E, by default, downshift is enabled on multigigabit ports. When an interface is unable to establish a high speed link, the line rate is automatically downshifted or reduced to a lower speed. The interface tries up to four times to reestablish a link using the current speed, before downshifting to the next available lower speed. Multigigabit interfaces support downshifting only when the interface speed is set to *auto* on both sides of the link. For information about enabling downshift on WS-X4748-12X48U+E, see [Setting Multigigabit Ethernet Interface Speeds, page 9-35](#).

For more information on 802.11ac Wave 2, see

http://www.cisco.com/c/en/us/products/collateral/wireless/aironet-3600-series/white_paper_c11-713103.html.

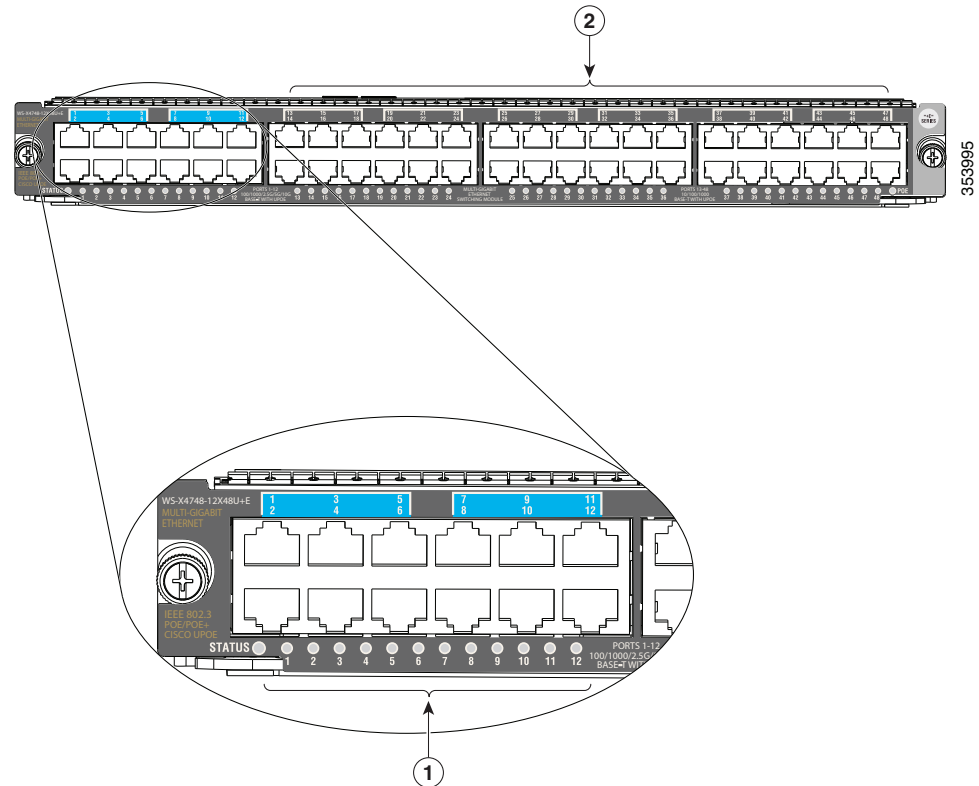
This section includes:

- [Module Modes on WS-X4748-12X48U+E Overview, page 9-18](#)
- [Configuring Module Modes on WS-X4748-12X48U+E, page 9-21](#)
- [Upgrading the Line Card FPGA Image on WS-X4748-12X48U+E, page 9-21](#)

For information about setting 2.5G, 5G and 10G interface speeds on WS-X4748-12X48U+E, see [Setting Multigigabit Ethernet Interface Speeds, page 9-35](#).

Module Modes on WS-X4748-12X48U+E Overview

The WS-X4748-12X48U+E module has 48 ports, of which the first 12 ports are 10-Gigabit Ethernet ports and, ports 13 to 48 are GigabitEthernet ports.

Figure 9-4 The WS-X4748-12X48U+E module front panel

1	Ports 1 to 12 - Multigigabit Ethernet ports	2	Ports 13 to 48 - 1 Gigabit Ethernet ports
----------	---	----------	---

You can configure the WS-X4748-12X48U+E module in one of the 3 modes

Mode 1 - Multigigabit Lite (Default)

- Ports 1 to 12 are 100/1000/2.5G/5G/10G Ethernet UPOE ports, oversubscribed 10:1 for 10Gbps port speed.
- Ports 13 to 48 are 10/100/1000 UPOE ports, with no oversubscription.

Mode 2 - Multigigabit Enhanced

- Ports 1 to 12 are 100/1000/2.5G/5G/10G Ethernet UPOE ports, oversubscribed 5:1 for 10Gbps port speed. Ports 1 to 6 share 12G aggregate bandwidth, and ports 7 to 12 share 12G aggregate bandwidth.
- Ports 13 to 24 are inactive (disabled). The LED display for these ports, on the front panel of the device is Off.
- Ports 25 to 48 are 10/100/1000 UPOE ports, with no oversubscription.

Mode 3 - Multigigabit Performance for 10Gbps port speed.

- Ports 1 to 12 are 100/1000/2.5G/5G/10G Ethernet UPOE ports, oversubscribed 2.5:1 for 10Gbps port speed. In this mode, ports 1 to 3, 4 to 6, 7 to 9, 10 to 12 share an aggregate bandwidth of 12G amongst the 3 ports in each group.
- Ports 13 to 48 are inactive (disabled). The LED display for these ports, on the front panel of the device is Off.


Note

10-Gigabit Ethernet, for links up to 100 meters, is supported on Category 6A cables or higher only.

Restrictions for Multigigabit Ports


Note

These restrictions do not apply to ports 13 to 48 that do not support Multigigabit Ethernet

- Ports 1 to 12 do not support 10Mbps speed
- Ports 1 to 12 do not support half-duplex mode for 100Mbps speed.
- Ports 1 to 12 do not support Energy Efficient Ethernet.

Supported Cable Types and Speed

Table 9-1 Cable types and speed

Cable Type	100Mbps	1G	2.5G	5G	10G
Category 5e	Yes	Yes	Yes	Yes	N/A
Category 6	Yes	Yes	Yes	Yes	Yes (55 meters)
Category 6a	Yes	Yes	Yes	Yes	Yes

Supported Hardware and Power Supply

Table 9-2 Supported hardware and power supply

Chassis Type	Supervisor	Power Supply
<ul style="list-style-type: none"> • Cisco Catalyst 4503-E • Cisco Catalyst 4506-E • Cisco Catalyst 4507R+E • Cisco Catalyst 4510R+E 	<ul style="list-style-type: none"> • Supervisor Engine 9-E • Supervisor Engine 8L-E • Supervisor Engine 8-E • Supervisor Engine 7-E • Supervisor Engine 7L-E 	<ul style="list-style-type: none"> • PWR-C45-1300AC • PWR-C45-2800AC • PWR-C45-4200AC • PWR-C45-6000AC • PWR-C45-9000AC • PWR-C45-1400AC¹

1. Power on Ethernet is not supported on PWR-C45-1400AC.

Table 9-3 Hardware and power supply not supported

Chassis Type	Supervisor
<ul style="list-style-type: none"> Cisco Catalyst 4507R-E Cisco Catalyst 4510R-E 	<ul style="list-style-type: none"> Supervisor Engine 6-E, Supervisor Engine 6L-E and earlier.

Configuring Module Modes on WS-X4748-12X48U+E

**Note**

This mode command is applicable only to the selected module. When the mode command is changed for a module, the operation of other modules on the switch is not impacted.

	Command	Purpose
Step 1	Switch# configure terminal	Establishes global configuration mode.
Step 2	Switch(config)# hw-module module <slot> mode <number>	Selects the configuration mode for the selected module.

**Note**

For the selected configuration to be applied, you must reset the module using the **Switch#hw-module module <slot> reset** command.

To verify that the selected configuration is applied, enter the **Switch#show module** command.

The following example shows how to configure an Ethernet mode on WS-X4748-12X48U+E:

```
Switch(config)#hw-module module 3 mode ?
1 Multigigabit lite :12 MultiGigabit ports(10:1 oversubscribed) and 36 1Gigabit ports
2 Multigigabit enhanced :12 MultiGigabit ports(5:1 oversubscribed) and 24 1Gigabit ports
3 Multigigabit performance :12 MultiGigabit ports(2.5:1 oversubscribed) only
```

For information about setting interface speeds to 2.5G, 5G or 10G, see [Setting the Interface Speed, page 9-33](#).

Upgrading the Line Card FPGA Image on WS-X4748-12X48U+E

Beginning in Cisco IOS Release 3.7.1E, a new function to upgrade the line card FPGA on the WS-X4748-12X48U+E module has been introduced to handle oversubscription on multigigabit ports. This upgrade capability allows feature updates or fixes related to the line card FPGA on this module.

**Note**

Before you begin, download the new FPGA image to the bootflash, to a TFTP server, or to a USB drive.

**Caution**

Configuration for this module will be suspended during FPGA upgrade. The module is automatically reset after the upgrade.

To upgrade the line card FPGA on WS-X4748-12X48U+E:

	Command	Purpose
Step 1	Switch# configure terminal	Establishes global configuration mode.
Step 2	Switch(config)# hw-module module <slot> fpga-upgrade image <path:filename>	Upgrades the line card FPGA image on the module.

The following example shows how to upgrade the line card FPGA image:

```
Switch(config)#hw-module slot 1 fpga-upgrade image ?
```

```
bootflash: Path to Linecard FPGA image
tftp: Path to Linecard FPGA image
slot0: Path to Linecard FPGA image
usb0: Path to Linecard FPGA image
```

Invoking Shared-Backplane Uplink Mode on Supervisor Engine 6-E and Supervisor Engine 6L-E

This feature enables you to use all four 10-Gigabit Ethernet ports on the supervisor engines as blocking ports when in redundant mode.

Prior to Cisco IOS Release 12.2(40)SG, Catalyst 4500 Supervisor Engine V-10GE allowed you to enable either the dual wire-speed 10-Gigabit Ethernet ports or four TwinGig convertor based Gigabit Ethernet SFP uplink ports when operating in redundant mode.

Beginning with Cisco IOS Release 12.2(40)SG, you could deploy all four 10-Gigabit Ethernet ports, two blocking ports on an active supervisor engine and two blocking ports on the standby supervisor engine, or all eight Gigabit Ethernet SFP ports, four on the active supervisor and four on the standby supervisor engine. This capability is supported on all Catalyst 4500 and 4500E series chassis.

To enable shared-backplane mode, enter this command:

Command	Purpose
Switch(config)# hw-mod uplink mode shared-backplane	A reload of the active supervisor engine is required to apply the new configuration.

To disable shared-backplane mode, enter this command:

Command	Purpose
Switch(config)# no hw-mod uplink mode shared-backplane	A reload of the active supervisor engine is required to apply the new configuration.

Selecting Uplink Mode on a Supervisor Engine 6-E

You can use the **hw-module uplink mode** command to change the uplink mode to either shared-backplane or tengigabitethernet mode.



Note

Only two 10-Gigabit Ethernet ports or four 1-Gigabit Ethernet ports can be used on the supervisor engine.



Note

When changing the uplink mode using the **hw-module uplink mode shared-backplane** command, you must reload the system. A message appears on the console to reflect this.

To select shared-backplane mode, do the following:

```
Switch(config)# hw-module uplink mode shared-backplane
A reload of the active supervisor is required to apply the new configuration.
Switch(config)# exit
Switch#
```

On a Supervisor Engine 6-E in a 6 or 7-slot chassis (Catalyst 4506-E, 4507R-E, and 4507R+E), the default uplink mode does not allow a WS-X4640-CSFP-E linecard to boot in the last slot because of a hardware limitation. After you use the **hw-module uplink mode tengigabitethernet** command, you must reload the system to enable TenGigabit mode. The configuration is NVGEN'd after you save the running configuration to the startup configuration. You can use the **show run | incl uplink** command to check the uplink configuration before reloading the system. Furthermore, you can enter the **show hw-module uplink** command to display the uplink mode. It reports the current uplink mode, as well as the mode after the system reloads.

In uplink Ten Gigabit mode, the uplink is limited to two Ten Gigabit Ethernet interfaces in non-redundant and in redundant mode; Gigabit Ethernet interfaces are not supported. The WS-X4640-CSFP-E linecard boots in the last slot on 6 and 7-slot chassis. To return to default mode, reload the system from tengigabitethernet mode. SharedBackplane mode can be selected from Default mode, where a system reload is required as well.

The **hw-module module x port-group x select gigabitethernet** command is blocked in uplink TenGigabit mode, preventing you from selecting gigabitethernet mode.

Support for WS-X46490-CSFP-E on a 10-slot Chassis

Starting with Release IOS XE 3.5.0E and IOS 15.2(1)E, the WS-X4640-CSFP-E linecard is supported in a 10-slot chassis for switch mode stand alone and switch mode virtual (VSS). New configurations are saved and reloaded so that the new configuration is applied to the entire switch.

To enable support for WS-X4640-CSFP-E on a 10-slot chassis, enter the following:

```
switch# configuration terminal
switch# hw-module system max-port-num-mode 2
```

To restore the default mode (**Mode 1**), you need to enter the following:

```
switch# configuration terminal
switch# no hw-module system max-port-num-mode 2
```

OR

```
switch# configuration terminal
switch# no hw-module system max-port-num-mode 1
```

OR

```
switch# configuration terminal
switch# hw-module system max-port-num-mode 1
```

Mode 1—a default mode in which the switch is configured to max Vfe subports of 48 per linecard; this configuration does not allow "WS-X4640-CSFP-E" in any of the 8 LC slots.

Mode 2—requires the above configuration to be copied to the startup-config. In this mode, only the first five slots of the chassis are eligible for the line cards. The remaining three slots are unuseable by any other line card including WS-X4640-CSFP-E, effectively reducing a 10-slot chassis to a 7-slot. You can place WS-X4640-CSFP-E in the top five slots, enabling you to use C-Sfp and thereby allow the max 80 ports on each WS-X4640-CSFP-E.

The behavior of a Supervisor Engine 6-E in Mode 2 (which is 10-slot chassis behaving effectively as a 7-slot chassis) matches that of a 7-slot chassis with Supervisor Engine 6-E. In Mode 2, the supervisor engine WS-X45-SUP6-E with the linecard WS-X4640-CSFP-E in the 7th slot will be inactive until an uplink mode change (using the **hw-module uplink mode tengigabitethernet** command) and a reboot. In this case, a syslog is printed to notify the presence of WS-X4640-CSFP-E in the 7th slot to prevent you from performing repetitive reboots.

To display the current mode in which the system is running, enter the following:

```
switch# show hw-module system max-port-num-mode
```



Note

This command is visible only on 10-slot chassis or if 10-slot chassis is present in VSS.

The output of this command is:

```
Active max-port-num-mode configuration is 2
In this mode, last 3 Line card slots shall not be active"
```

In VSS, this output provides the current mode of both active and standby switches.

Limitation and Restrictions on Supervisor Engine 7-E and Supervisor Engine 7L-E

When you use WS-X45-SUP7-E in RPR or SSO mode, only the first two uplinks on each supervisor engine are available. The second two uplinks are unavailable.

Selecting the Uplink Port on a Supervisor Engine 7L-E

With Cisco IOS Release 15.0(2)SG, the SFP+/SFP uplink modes on Supervisor Engine 7L-E (WS-X45-SUP-7L-E) have changed. The number of uplink ports now depends on the supervisor engine mode (single or redundant) and the uplink mode configuration (1-Gigabit or 10-Gigabit). To configure the uplink mode, use the **hw-module uplink select [gigabitethernet/ tengigabitethernet]** command, as follows:

```
Switch(config)# hw-module uplink select?
gigabitethernet      Select the gigabit uplinks
tengigabitethernet   Select the 10G uplinks
```


**Note**

Supervisor Engine 7L-E is not supported on a ten-slot chassis. USB device and SD card support is applicable to Supervisor Engines 9-E, 8-E, and 7-E only.

**Note**

Supervisor Engine 8-E is supported on a 10-slot chassis.

Single Supervisor Mode

In single supervisor mode, WS-X45-SUP-7L-E supports the uplink configuration of at most either two 10-Gigabit or four 1-Gigabit ports ([Table 9-4](#)).

Table 9-4 Uplink Options for Single Supervisor Mode

Supervisor Engine Uplink Ports		Speeds Achievable with the Following Combination of Pluggables (Band Width)
A1	A2	
Choose 10-Gigabit operation through the command line interface.		
SFP+	SFP+	20 Gbps
SFP+	SFP	11 Gbps

SFP+ and SFP can be inserted in any order for lines 2 to 4.

Redundant Supervisor Mode

In redundant supervisor mode, WS-X45-SUP-7L-E supports 1+1 (in 10-Gigabit mode) and 2+2 (in 1-Gigabit mode) ([Table 9-5](#)).

Table 9-5 Uplink Options for Redundant Supervisor Mode

Active Supervisor Uplink Ports				Standby Supervisor Uplink Ports				Speeds Achievable with this Combination of Pluggables
A1	A2	A3	A4	B1	B2	B3	B4	
Choose 10-Gigabit operation through the command line interface.								
SFP+		—	—	SFP+		—	—	20 Gbps
SFP+		—	—	SFP		—	—	11 Gbps
SFP		—	—	SFP+		—	—	11 Gbps
SFP	SFP	—	—	SFP	SFP	—	—	4 Gbps

Limitations and Restrictions on Supervisor Engine 8-E

- When you use Supervisor Engine 8-E in RPR or SSO mode, only the first four uplinks on each supervisor engine are available. The second set of four uplinks are unavailable.

- Memory utilization for the device is calculated using a combination of the **show memory location active** and the **show memory location active-dc** commands. With a daughter card enabled on Supervisor Engine 8-E, the CLI output for the command **show memory location active-dc** shows the value 0 for the config total as no configuration is stored in the daughter card.

Configuring Supervisor Engine 7-E Uplink Mode on Supervisor Engine 8-E

In a ten-slot chassis, by default, Supervisor Engine 8-E supports 80GB uplink bandwidth. The supervisor supports eight active interfaces in non-redundancy mode and the first four active interfaces on both active and standby supervisors, in redundancy mode. Only 47xx series line cards are supported in the tenth slot of the 4510 R+E chassis. The 4510R-E chassis does not support any line cards in the tenth slot, when the daughter card is enabled.

When the daughter card is enabled, Supervisor Engine 8-E restricts uplink bandwidth to 40GB.

For all non-ten-slot chassis (3,6, and 7-slot chassis), Supervisor Engine 8-E mode allows 80GB uplink bandwidth, even with the daughter card enabled.

Supervisor Engine 7-E Mode on Supervisor Engine 8-E



Note

You can configure Supervisor Engine 7-E mode (Sup7-E mode) on Supervisor Engine 8-E in a 10-slot chassis only.

When Sup 7-E mode is enabled, the Supervisor Engine 8-E base board uplink bandwidth is restricted to 40GB as the default configuration in a ten-slot chassis. In non-redundancy mode, the supervisor can support the first four active interfaces. In redundancy mode, this mode supports the first two interfaces on both active and standby supervisors. All line cards are supported in all ten slots in the chassis, with no restriction on the tenth slot.

To enable Sup7-E mode, you must disable the daughter card on the switch.

Supervisor Engine 8-E with Daughter Card Enabled

By default the daughter card is enabled when the supervisor engine is booted in Install Boot mode, and disabled when the supervisor engine is booted in Bundle Boot mode.

When the daughter card is enabled, Supervisor Engine 8-E base board uplink bandwidth is restricted to 40GB as the default configuration in a ten-slot chassis. In non-redundancy mode, the supervisor can support the first four active interfaces. In redundancy mode, the first two interfaces on both the active and the standby supervisors become active. Only 47xx series line cards are supported in the tenth slot of the 4510 R+E chassis. The 4510R-E chassis does not support any line cards in the tenth slot, when the daughter card is enabled.

Supervisor Engine 8-E Uplink Configurations

The following table displays the default uplink configuration for Supervisor Engine 8-E, based on the redundancy mode and whether the daughter card is enabled.

Table 9-6 **Sup 8-E Uplink Configuration Modes**

Supervisor Configuration	Uplink Ports	Wireless Termination/Daughter Card	Linecard/Chassis Restrictions	User Configuration
Single Supervisor (Sup7E Mode)	4x10GE		10-slot Chassis: No restriction	Requires CLI configuration followed by a supervisor engine reload.
			3,6 and 7 Slot Chassis: Configuration not supported	
Single Supervisor	4x10GE	20GE	10-slot Chassis: Restriction on the 10th slot.*	Default configuration in Install Boot mode.
			3,6 and 7 Slot Chassis: No restriction	
Single Supervisor	8x10GE		10-slot Chassis: Restriction on the 10th slot.*	Default configuration in Bundle Boot mode.
			3,6 and 7 Slot Chassis: No restriction	
Single Supervisor	8x10GE		10-slot Chassis: Restriction on the 10th slot.* Wireless/Daughter Card enabled.	Default configuration in Install Boot mode. Requires CLI configuration to force disable the daughter card followed by a supervisor engine reload.
		20GE	3,6 and 7 Slot Chassis: No restriction	Default configuration in Install Boot mode.
Dual Supervisor (Sup 7-E mode)	Active Supervisor: 2x10GE		10-slot Chassis: No restriction	Requires CLI configuration followed by a supervisor engine reload.
	Standby Supervisor: 2x10GE		3,6 and 7 Slot Chassis: Configuration not supported	
Dual Supervisor	Active Supervisor: 2x10GE	20GE	10-slot Chassis: Restriction on the 10th slot.*	Default configuration in Install Boot mode.
	Standby Supervisor: 2x10GE		3,6 and 7 Slot Chassis: Configuration not supported	

Table 9-6 Sup 8-E Uplink Configuration Modes

Supervisor Configuration	Uplink Ports	Wireless Termination/Daughter Card	Linecard/Chassis Restrictions	User Configuration
Dual Supervisor	Active Supervisor: 4x10GE		10-slot Chassis: Restriction on the 10th slot.*	Default configuration in Bundle Boot mode.
	Standby Supervisor: 4x10GE		3,6 and 7 Slot Chassis: No restriction	Requires CLI configuration followed by a supervisor engine reload.
Dual Supervisor	Active Supervisor: 4x10GE	20GE	10-slot Chassis: Restriction on the 10th slot.* Wireless Mode enabled by default.	Default configuration in Install Boot mode. Requires CLI configuration to force disabled the daughter card followed by a supervisor engine reload.
	Standby Supervisor: 4x10GE		3,6 and 7 Slot Chassis: No restriction	Default configuration in Install Boot mode.

*Only 47xx series line cards are supported in the tenth slot of the 4510 R+E chassis. The 4510R-E chassis does not support any line cards in the tenth slot, when the daughter card is enabled.

Restrictions for Configuring Sup 7-E Uplink Mode on Supervisor Engine 8-E

- In Install Boot mode, Supervisor Engine 7-E cannot be configured directly configured as the daughter card is enabled by default.
- Supervisor Engine 7-E mode cannot be configured on a 3, 6, and 7-slot chassis.
- Supervisor Engine 7-E mode is supported only for a symmetric VSS system.

Configuring Supervisor Engine 7-E Mode on Supervisor Engine 8-E



Note

Before you begin, ensure that the daughter card is disabled on the switch.

By default, the daughter card is enabled when the supervisor engine is booted in Install Boot mode and you cannot configure Supervisor Engine 7-E (Sup 7-E) mode until you disable the daughter card on the switch.



Note

Ensure that the supervisor engine is reloaded each time you make uplink configuration changes.

To disable the daughter card, enter the following commands:

	Command	Purpose
Step 1	switch (config)# hw-module daughtercard disable	Disables the daughter card on the switch. Note This command is not available in Bundle Boot mode.
Step 2	Save the configuration and perform a power cycle or reload the supervisor. Then enter: switch# sh hw-module daughtercard	Displays updated daughter card details for Supervisor Engine 8-E, after the supervisor engine is reloaded.

The **no** form of the **hw-module daughtercard disable** command enables the daughter card.

To enable Sup7-E mode on Supervisor Engine 8-E, when the daughter card is disabled, enter the following commands:

	Command	Purpose
Step 1	switch (config)# hw-module uplink mode 40Gig	Configures the uplink mode to 40GB.
Step 2	Save the configuration and perform a power cycle or reload the supervisor. Then enter: switch# show run incl uplink	Displays the uplink configuration for Supervisor Engine 8-E.

The **no** form of the **hw-module uplink mode 40Gig** command restores the default Supervisor Engine 8-E mode.

Supervisor Engine 9-E Uplink Configurations

Starting with Cisco IOS XE Release 3.10.0E, Cisco Catalyst 4500E Series Switches support Supervisor Engine 9-E. This supervisor model provides

- Four 10 Gigabit Ethernet (4 x 10GE) uplink ports.
- Two 40 Gigabit Ethernet (2 x 40GE) uplink ports

Table 9-7 **Sup 9-E Uplink Configuration Modes**

Supervisor Configuration	Uplink Ports	Wireless Termination or Daughter Card	Linecard/Chassis Restrictions	User Configuration
Single Supervisor	4x10GE	—	10-slot Chassis: None 3,6 and 7 Slot Chassis: None	Requires CLI configuration followed by a supervisor engine reload.
Single Supervisor	4x10GE	20GE	10-slot Chassis: Restriction on the 10th slot.* 3,6 and 7 Slot Chassis: None	

Table 9-7 **Sup 9-E Uplink Configuration Modes**

Supervisor Configuration	Uplink Ports	Wireless Termination or Daughter Card	Linecard/Chassis Restrictions	User Configuration
Single Supervisor	2x40GE	—	10-slot Chassis: Restriction on the 10th slot.*	Default configuration in Bundle Boot mode.
			3,6 and 7 Slot Chassis: None	
Single Supervisor	2x40GE	20GE	10-slot Chassis: 10th slot disabled Daughter Card enabled.	Default configuration in Install Boot mode. Requires CLI configuration to force disable the daughter card followed by a supervisor engine reload.
			3,6 and 7 Slot Chassis: None	Default configuration in Install Boot mode.
Dual Supervisor	Active Supervisor: 2x10GE Standby Supervisor: 2x10GE	—	10-slot Chassis: None	Requires CLI configuration followed by a supervisor engine reload.
			7 Slot Chassis: None	
Dual Supervisor	Active Supervisor: 2x10GE Standby Supervisor: 2x10GE	20GE	10-slot Chassis: Restriction on the 10th slot.*	Default configuration in Install Boot mode.
			7 Slot Chassis: None	
Dual Supervisor	Active Supervisor: 1x40GE Standby Supervisor: 1x40GE	—	10-slot Chassis: Restriction on the 10th slot.*	Default configuration in Bundle Boot mode.
			7 Slot Chassis: None	Requires CLI configuration followed by a supervisor engine reload.
Dual Supervisor	Active Supervisor: 1x40GE Standby Supervisor: 1x40GE	20GE	10-slot Chassis: Unsupported (10-slot is disabled)	-
			7 Slot Chassis: None	Default configuration in Install Boot mode.

*Only 47xx series line cards are supported in the tenth slot of the 4510R+E chassis.

Limitations and Restrictions on Supervisor Engine 9-E

- The supervisor is not compatible with the Catalyst 4510R-E and 4507R-E chassis

- When the **hw-module uplink mode 80Gig** command is enabled, the wireless mode is unsupported in the 10th slot of the Catalyst 4510R+E chassis

Selecting the Uplink Mode on Supervisor Engine 9-E

By default, when the supervisor boots, the 4 x 10GE uplinks are available. To boot with 2 x 40GE uplinks, you must enable the 80Gig mode:

1. Check current configuration on the active supervisor. The example shows that the current mode is to boot with 4 x 10GE uplinks:

```
Switch (config)# show hw-module uplink
Active uplink mode configuration is 4X10Gig
```

2. Specify 80Gig in the global configuration mode

```
Switch# configure terminal
Switch(config)# hw-module uplink mode 80Gig
A reload or power-cycle of the active supervisor is required to apply the new
configuration.
Switch(config)#
```

3. Power cycle the active supervisor engine for configuration to take effect:

```
rommon 0 >set
PS1=rommon ! >
RommonVer=15.1(1r)SG(11)
...<output truncated>
cisco WS-C4507R+E (P5040) processor (revision 2) with 4194304K bytes of physical
memory.
Processor board ID FXS1736Q20Q
P5040 CPU at 2.2GHz, Supervisor 9-E
Last reset from Reload
1 Virtual Ethernet interface
48 Gigabit Ethernet interfaces
2 Forty Gigabit Ethernet interfaces
511K bytes of non-volatile configuration memory
...<output truncated>
```

4. Verify current configuration on the active supervisor again. The example shows that the current mode is now 2 x40GE uplinks (a total of 80Gig):

```
Switch# show hw-module uplink
Active uplink mode configuration is 2X40Gig
```

5. You will now be able to configure 40-GE interfaces

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface ?
Bluetooth          Bluetooth interface
FastEthernet        FastEthernet IEEE 802.3
FortyGigabitEthernet Forty Gigabit Ethernet
GigabitEthernet     GigabitEthernet IEEE 802.3z
LISP                Locator/ID Separation Protocol Virtual Interface
Loopback            Loopback interface
Lspvif              LSP virtual interface
Null                Null interface
Port-channel        Ethernet Channel of interfaces
Tunnel              Tunnel interface
Vlan                Catalyst Vlans
range               interface range command
```

```
Switch(config)#interface fortygigabitethernet1/1
Switch (config-if)#
```

The **no** form of the **hw-module uplink mode 80Gig** command restores the default Supervisor Engine 9-E mode (4 x 10GE uplinks).

Digital Optical Monitoring Transceiver Support

Command line interface (CLI) commands (**show inventory**, **show idprom interface**) are used on transceivers to obtain serial number, model name, inventory information.

The following commands are specific to the transceivers that support the DOM capability:

- Displays current values and thresholds for all sensor on a particular interface transceiver:

```
show interfaces int-name transceiver [detail] [threshold]
```

- Enables or disables the *entSensorThresholdNotification* for all sensors in all the transceivers:

```
snmp-server enable trap transceiver
```

- Enables or disables transceiver monitoring:

```
transceiver type all
monitoring
```



Note

This feature is only available when a DOM capable transceiver is present and configured for monitoring. The frequency at which the sensor information is refreshed depends on default values configured in the transceiver EEPROM (Serial Electrically Erasable Programmable Read Only Memory).



Note

For details on transceiver module compatibility, refer to this URL:

http://www.cisco.com/en/US/products/hw/modules/ps5455/products_device_support_tables_list.html

Configuring Optional Interface Features

The following sections describe optional procedures:

- [Configuring Ethernet Interface Speed and Duplex Mode, page 9-32](#)
- [Configuring Flow Control, page 9-38](#)
- [Configuring Jumbo Frame Support, page 9-40](#)
- [Interacting with Baby Giants, page 9-44](#)
- [Configuring the Port Debounce Timer, page 9-44](#)
- [Configuring Auto-MDIX on a Port, page 9-45](#)

Configuring Ethernet Interface Speed and Duplex Mode

Topics include:

- [Speed and Duplex Mode Configuration Guidelines, page 9-33](#)
- [Setting the Interface Speed, page 9-33](#)
- [Setting the Interface Duplex Mode, page 9-35](#)
- [Displaying the Interface Speed and Duplex Mode Configuration, page 9-36](#)
- [Adding a Description for an Interface, page 9-38](#)

Speed and Duplex Mode Configuration Guidelines



Note

You do not configure the client device for autonegotiation. Instead, you configure the switch with the speed, or range of speeds, that you want to autonegotiate.

You can configure the interface speed and duplex mode parameters to **auto** and allow the Catalyst 4500 series switch to negotiate the interface speed and duplex mode between interfaces. If you decide to configure the interface **speed** and **duplex** commands manually, consider the following:

- If you enter the **no speed** command, the switch automatically configures both interface **speed** and **duplex** to **auto**.
- When you set the interface speed to **1000** (Mbps) or **auto 1000**, the duplex mode is full duplex. You cannot change the duplex mode.
- If the interface speed is set to **10** or **100**, the duplex mode is set to half duplex by default unless you explicitly configure it.

On a Catalyst 4500-X switch, the duplex mode is always automatically configured as full duplex (whether you set the interface speed to **10** or **100**). You cannot change this.



Caution

Changing the interface speed and duplex mode configuration might shut down and restart the interface during the reconfiguration.

Setting the Interface Speed

If you set the interface speed to **auto** on a 10/100-Mbps Ethernet interface, speed and duplex are autonegotiated. The forced 10/100 autonegotiation feature allows you to limit interface speed auto negotiation up to 100 Mbps on a 10/100/1000BASE-T port.

To set the port speed for a 10/100-Mbps Ethernet interface, perform this task:

	Command	Purpose
Step 1	Switch(config)# interface fastethernet <i>slot/interface</i>	Specifies the interface to be configured.
Step 2	Switch(config-if)# speed [10 100 auto [10 100]]	Sets the interface speed.

This example shows how to set the interface speed to 100 Mbps on the Fast Ethernet interface 5/4:

```
Switch(config)# interface fastethernet 5/4
Switch(config-if)# speed 100
```

This example shows how to allow Fast Ethernet interface 5/4 to autonegotiate the speed and duplex mode:

```
Switch(config)# interface fastethernet 5/4
Switch(config-if)# speed auto
```



Note The preceding CLI is analogous to **speed auto 10 100**.

This example shows how to limit the interface speed to 10 and 100 Mbps on the Gigabit Ethernet interface 1/1 in auto-negotiation mode:

```
Switch(config)# interface gigabitethernet 1/1
Switch(config-if)# speed auto 10 100
```

This example shows how to limit speed negotiation to 100 Mbps on the Gigabit Ethernet interface 1/1:

```
Switch(config)# interface gigabitethernet 1/1
Switch(config-if)# speed auto 100
```



Note Turning off autonegotiation on a Gigabit Ethernet interface results in the port being forced into 1000 Mbps and full-duplex mode.

To turn off the port speed autonegotiation for Gigabit Ethernet interface 1/1, perform this task:

	Command	Purpose
Step 1	Switch(config)# interface gigabitethernet1/1	Specifies the interface to be configured.
Step 2	Switch(config-if)# speed nonegotiate	Disables autonegotiation on the interface.

To restore autonegotiation, enter the **no speed nonegotiate** command in the interface configuration mode.



Note For the blocking ports on the WS-X4416 module, do not set the speed to autonegotiate. To set port speed to 10/ 100 Mbps on a 10-Gigabit Ethernet interface (on a 1000Base-T port), perform this task:

	Command	Purpose
Step 1	Switch(config)# interface tengigabitethernet slot/interface	Specifies the interface to be configured.
Step 2	Switch(config-if)# speed [10 100]	Sets the interface speed. Forced 10/100 auto-negotiation is not supported on a 1000Base-T port

To restore autonegotiation (default setting), enter the **no speed** command in the interface configuration mode.

This example shows how to set the interface speed to 100 Mbps on a 10-Gigabit Ethernet interface 2/4:

```
Switch(config)# interface tengigabitethernet 2/4
Switch (config-if)# speed 100
```

Setting Multigigabit Ethernet Interface Speeds

Only ports 1 to 12 on the WS-X4748-12X48U+E module support Multigigabit Ethernet. For more information, see [Configuring MultiGigabit Ports on WS-X4748-12X48U+E, page 9-18](#).

To set port speed to 2500Mbps (2.5Gbps) / 5000Mbps (5Gbps) / 10000Mbps (10Gbps) on a Multigigabit Ethernet interface (on a 1000Base-T port), perform this task:



Note

For Multigigabit Ethernet ports 1 to 12, autonegotiation is enabled by default, and the link is active if both peers are Multigigabit Ethernet ports, and if forced speed is configured.

For 1 Gigabit Ethernet ports 13 to 48, the link is active only if autonegotiation is enabled at least at one end, and forced speed is configured.

	Command	Purpose
Step 1	Switch(config)# interface tengigabitethernet <i>slot/interface</i>	Specifies the interface to be configured.
Step 2	Switch(config-if)# speed [100 1000 2500 5000 10000 auto [100 1000 2500 5000 10000]]	Sets the interface speed. Note 10G speed is supported only on Category 6 and Category 6a cables.
Step 3	Switch(config-if)# [no] downshift disable	By default, downshift is enabled on multigigabit ports. The downshift disable command disables downshift on the specified interface. The no downshift disable command enables downshift on the interface.

This example shows how to set the interface speed to 5G on the Multigigabit Ethernet interface 3/1:

```
Switch(config)# interface tengigabitethernet 3/1
Switch (config-if)# speed 5000
```

This example shows how to allow the Multigigabit Ethernet interface 3/1 to autonegotiate the speed and duplex mode:

```
Switch(config)# interface gigabitethernet 3/1
Switch(config-if)# speed auto
```

This example shows how to limit speed negotiation to 2.5G on the Multigigabit Ethernet interface 3/1:

```
Switch(config)# interface gigabitethernet 3/1
Switch(config-if)# speed auto 2500
```

This example shows how to enable downshift on the Multigigabit Ethernet interface 3/1:

```
Switch(config)# interface gigabitethernet 3/1
Switch(config-if)# speed auto
Switch(config-if)# no disable downshift
```

Setting the Interface Duplex Mode



Note

When the interface is set to 1000 Mbps, you cannot change the duplex mode from full duplex to half duplex.

To set the duplex mode of a Fast Ethernet interface, perform this task:

	Command	Purpose
Step 1	Switch(config)# interface fastethernet <i>slot/interface</i>	Specifies the interface to be configured.
Step 2	Switch(config-if)# duplex [auto full half]	Sets the duplex mode of the interface.

This example shows how to set the interface duplex mode to full on Fast Ethernet interface 5/4:

```
Switch(config)# interface fastethernet 5/4
Switch(config-if)# duplex full
```

Displaying the Interface Speed and Duplex Mode Configuration

To display the interface speed and duplex mode configuration for an interface, enter this command:

Command	Purpose
Switch# show interfaces [fastethernet gigabitethernet tengigabitethernet] <i>slot/interface</i>	Displays the interface speed and duplex mode configuration.

This example shows how to display the interface speed and duplex mode of Fast Ethernet interface 6/1:

```
Switch# show interface fastethernet 6/1
FastEthernet6/1 is up, line protocol is up
  Hardware is Fast Ethernet Port, address is 0050.547a.dee0 (bia 0050.547a.dee0)
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 100Mb/s
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:54, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 50/2000/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    50 packets input, 11300 bytes, 0 no buffer
    Received 50 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
    0 input packets with dribble condition detected
    1456 packets output, 111609 bytes, 0 underruns
    0 output errors, 0 collisions, 0 interface resets
    0 babbles, 0 late collision, 0 deferred
    1 lost carrier, 0 no carrier
    0 output buffer failures, 0 output buffers swapped out
Switch#
```

This example shows how to display the interface speed and duplex mode of 10-Gigabit Ethernet interface 1/5:

```
Switch# show interface tengigabitethernet 1/5
TenGigabitEthernet1/5 is up, line protocol is up (connected)
  Hardware is Ten Gigabit Ethernet Port, address is 0022.bde2.0f6d (bia 0022.bde2.0f6d)
  MTU 1500 bytes, BW 10000000 Kbit/sec, DLY 10 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 10Gb/s, link type is auto, media type is 10GBase-SR
  input flow-control is on, output flow-control is on
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:01, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/2000/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    39 packets input, 8286 bytes, 0 no buffer
    Received 39 broadcasts (39 multicasts)
    0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
    0 input packets with dribble condition detected
    308 packets output, 30276 bytes, 0 underruns
    0 output errors, 0 collisions, 3 interface resets
    0 unknown protocol drops
    0 babbles, 0 late collision, 0 deferred
    0 lost carrier, 0 no carrier
    0 output buffer failures, 0 output buffers swapped out
Switch#
```

Adding a Description for an Interface

You can add a description about an interface to help you remember its function. The description appears in the output of the following commands: **show configuration**, **show running-config**, and **show interfaces**.

To add a description for an interface, enter the following command:

Command	Purpose
Switch(config-if)# description <i>string</i>	Adds a description for an interface.

This example shows how to add a description on Fast Ethernet interface 5/5:

```
Switch(config)# interface fastethernet 5/5
Switch(config-if)# description Channel-group to "Marketing"
```

Configuring Flow Control

Gigabit Ethernet ports use flow control to slow down the transmission of incoming packets. If a buffer on a Gigabit Ethernet port runs out of space, the port transmits a special packet that requests remote ports to delay sending packets for a period of time. The port can also receive this special packet from its link partner for the same purpose. This special packet is called a *pause frame*.

The default settings for Gigabit Ethernet interfaces are as follows:

- Sending pause frames is off—Non-oversubscribed Gigabit Ethernet interfaces.
- Receiving pause frames is desired—Non-oversubscribed Gigabit Ethernet interfaces.
- Sending pause frames is on—Oversubscribed Gigabit Ethernet interfaces.
- Receiving pause frames is desired—Oversubscribed Gigabit Ethernet interfaces

The default settings for 10-Gigabit Ethernet interfaces are as follows:

- Sending pause frames is off.
- Receiving pause frames is on.



Note

desired is not a flow control option on the 10-Gigabit Ethernet interfaces.

To configure flow control, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Enters interface configuration mode and specifies the interface to be enabled for flowcontrol.
Step 3	Switch(config-if)# flowcontrol { receive send } { off on desired }	Configures a Gigabit Ethernet port to send or receive pause frames.
Step 4	Switch(config-if)# end	Returns to configuration mode.
Step 5	Switch(config)# end	Returns to privileged EXEC mode.

This example shows how to configure flow control on an oversubscribed Gigabit Ethernet port 7/5:

```
Switch# configure terminal
Switch(config)# interface g7/5
Switch(config-if)# flowcontrol send on
Switch(config-if)# end
Switch)# show interfaces gigabitEthernet 7/5 capabilities
GigabitEthernet7/5
  Model: WS-X4548-GB-RJ45-RJ-45
  Type: 10/100/1000-TX
  Speed: 10,100,1000,auto
  Duplex: half,full,auto
  Trunk encap. type: 802.1Q,ISL
  Trunk mode: on,off,desirable,nonegotiate
  Channel: yes
  Broadcast suppression: percentage(0-100), hw
  Flowcontrol: rx- (off,on,desired),tx- (off,on,desired)
  VLAN Membership: static, dynamic
  Fast Start: yes
  Queuing: rx- (N/A), tx- (1p3q1t, Sharing/Shaping)
  CoS rewrite: yes
  ToS rewrite: yes
  Inline power: no
  SPAN: source/destination
  UDLD: yes
  Link Debounce: no
  Link Debounce Time: no
  Port Security: yes
  Dot1x: yes
  Maximum MTU: 1552 bytes (Baby Giants)
  Multiple Media Types: no
  Diagnostic Monitoring: N/A

Switch)# show flowcontrol interface GigabitEthernet 7/5
Port      Send FlowControl  Receive FlowControl  RxPause TxPause
      admin   oper      admin   oper
-----
Gi7/5    on      off      desired off      0        0
```

This example shows the output of the **show interfaces** and **show flow control** commands on a non-oversubscribed Gigabit Ethernet port 5/5:

```
Switch# show interfaces gigabitEthernet 5/5 capabilities
GigabitEthernet5/5
  Model: WS-X4306-GB-Gbic
  Type: No Gbic
  Speed: 1000
  Duplex: full
  Trunk encap. type: 802.1Q,ISL
  Trunk mode: on,off,desirable,nonegotiate
  Channel: yes
  Broadcast suppression: percentage(0-100), hw
  Flowcontrol: rx- (off,on,desired),tx- (off,on,desired)
  VLAN Membership: static, dynamic
  Fast Start: yes
  Queuing: rx- (N/A), tx- (1p3q1t, Sharing/Shaping)
  CoS rewrite: yes
  ToS rewrite: yes
  Inline power: no
  SPAN: source/destination
  UDLD: yes
  Link Debounce: no
  Link Debounce Time: no
  Port Security: yes
```

```

Dot1x:          yes
Maximum MTU:    9198 bytes (Jumbo Frames)
Multiple Media Types: no
Diagnostic Monitoring: N/A

```

```

Switch# show flowcontrol interface gigabitEthernet 5/5
Port      Send FlowControl  Receive FlowControl  RxPause TxPause
         admin    oper    admin    oper
-----
Gi5/5     off      off    desired off      0        0

```

This example shows the output of the **show interfaces** and **show flowcontrol** commands on an unsupported Fast Ethernet port 3/5:

```

Switch# show interfaces fa3/5 capabilities
FastEthernet3/5
  Model:          WS-X4148-RJ-45
  Type:           10/100BaseTX
  Speed:          10,100,auto
  Duplex:          half,full,auto
  Trunk encap. type: 802.1Q,ISL
  Trunk mode:      on,off,desirable,nonegotiate
  Channel:         yes
  Broadcast suppression: percentage(0-100), sw
  Flowcontrol:    rx-(none),tx-(none)
  VLAN Membership: static, dynamic
  Fast Start:      yes
  Queuing:          rx-(N/A), tx-(lp3q1t, Shaping)
  CoS rewrite:      yes
  ToS rewrite:      yes
  Inline power:     no
  SPAN:             source/destination
  UDLD:             yes
  Link Debounce:    no
  Link Debounce Time: no
  Port Security:    yes
  Dot1x:            yes
  Maximum MTU:      1552 bytes (Baby Giants)
  Multiple Media Types: no
  Diagnostic Monitoring: N/A

Switch# show flowcontrol interface fa3/5
Port      Send FlowControl  Receive FlowControl  RxPause TxPause
         admin    oper    admin    oper
-----
Fa3/5     Unsupp.  Unsupp.  Unsupp.  Unsupp.  0        0

```

Configuring Jumbo Frame Support

These sections describe jumbo frame support:

- [Ports and Modules That Support Jumbo Frames, page 9-40](#)
- [Jumbo Frame Support, page 9-41](#)
- [Configuring MTU Sizes, page 9-43](#)

Ports and Modules That Support Jumbo Frames

The following ports and modules support jumbo frames:

- Supervisor uplink ports
- WS-C4500X ports
- WS-X4306-GB: all ports
- WS-X4232-GB-RJ: ports 1-2
- WS-X4418-GB: ports 1-2
- WS-X4412-2GB-TX: ports 13-14
- WS-X4506-GB-T
- the 4648-GB-RJ45V
- WS-X4648-GB+RJ45V
- WS-X4648-RJ45V-E
- WS-X4648-RJ45V+E
- WS-X4706-10GE

Jumbo Frame Support

These sections describe jumbo frame support:

- [Maximum Transmission Units, page 9-41](#)
- [Jumbo Frame Support Overview, page 9-41](#)
- [Ethernet Ports, page 9-42](#)
- [VLAN Interfaces, page 9-42](#)

Maximum Transmission Units

The Catalyst 4500 series switch allows you to configure a maximum of 32 different maximum transmission unit (MTU) sizes system wide. This means that the maximum number of different MTU sizes that you can configure with the **system mtu**, **mtu**, **ip mtu**, and **ipv6 mtu** command on all Layer 2 and Layer 3 interfaces combined is 32.

Also, the system stores the IPv4 and IPv6 MTU sizes configured on an interface separately. For every **system mtu** command or per interface **mtu** command, two separate MTU values are stored, one for IPv4 and one for IPv6. This further reduces the number of slots available (out of 32). However, only a single MTU value is stored for each **ip mtu** and **ipv6 mtu** commands.

If the new MTU value you are configuring is already present in the system (that is, configured on some other interface), then no new slot(s) are allocated to store it again.

If the maximum limit of 32 is reached and an attempt is made to configure a new MTU size on a new interface, the system only allows configuration to proceed if the new MTU size has previously been configured on some interface. Otherwise, an error message is displayed and the default MTU size is assigned to the interface being configured.

Jumbo Frame Support Overview

A jumbo frame is a frame larger than the default Ethernet size. Enable jumbo frame support by configuring a larger-than-default MTU size on a port or interface.

Catalyst 4500 series switch Ethernet LAN ports configured with a nondefault MTU size accept frames containing packets with a size between 1500 and 9216 bytes (including Ethernet payload, header and trailer). With a nondefault MTU size configured, the packet size of ingress frames is checked. If the packet is larger than the configured MTU, it is dropped.

For traffic that needs to be routed, the MTU of the egress port is checked. If the MTU is smaller than the packet size, the packet is forwarded to the CPU. If the “do not fragment bit” is not set, it is fragmented. Otherwise, the packet is dropped.

**Note**

Jumbo frame support does not fragment Layer 2 switched packets.

The Catalyst 4500 series switch does not compare the packet size with the MTU at the egress port, but jumbo frames are dropped in ports that do not support them. The frames can be transmitted in ports that do support jumbo frames, even though the MTU is not configured to jumbo size.

**Note**

Jumbo frame support is only configured per interface; jumbo frame support cannot be configured globally.

Ethernet Ports

These sections describe configuring nondefault MTU sizes on Ethernet ports:

- [Ethernet Port Overview, page 9-42](#)
- [Layer 3 and Layer 2 EtherChannels, page 9-42](#)

Ethernet Port Overview

Starting with Cisco IOS Release 12.2(31)SGA, configuring a nondefault MTU size on certain Ethernet ports limits the size of ingress packets. The MTU does not impact the egress packets.

Prior to Cisco IOS Release 12.1(13)EW, you could configure the MTU size only on Gigabit Ethernet.

Layer 3 and Layer 2 EtherChannels

Starting with Release Cisco IOS Release 12.2(31)SGA, you could configure all the interfaces in an EtherChannel provided that they have the same MTU. Changing the MTU of an EtherChannel changes the MTU of all member ports. If the MTU of a member port cannot be changed to the new value, that port is suspended (administratively shut down). A port cannot join an EtherChannel if the port has a different MTU. If a member port of an EtherChannel changes MTU, the member port is suspended.

VLAN Interfaces

If switch ports reside in the same VLAN, either configure all of the switch ports to handle jumbo frames and support the same MTU size, or configure none of them. However, such uniformity of MTU size in the same VLAN is not enforced.

When a VLAN has switch ports with different MTU size, packets received from a port with a larger MTU might be dropped when they are forwarded to a port with a smaller MTU.

If the switch ports in a VLAN have jumbo frames enabled, the corresponding SVI can have jumbo frames enabled. The MTU of an SVI should always be smaller than the smallest MTU among all the switch ports in the VLAN, but this condition is not enforced.

The MTU of a packet is not checked on the ingress side for an SVI; it is checked on the egress side of an SVI. If the MTU of a packet is larger than the MTU of the egress SVI, the packet is sent to the CPU for fragmentation processing. If the “do not fragment” bit is not set, the packet is fragmented. Otherwise, the packet is dropped.

Configuring MTU Sizes

To configure the MTU size, perform this task:

	Command	Purpose
Step 1	Switch(config)# interface {{vlan vlan_ID} {{type ¹ slot/port} {port-channel port_channel_number} slot/port}}	Selects the interface to configure.
Step 2	Switch(config-if)# mtu mtu_size Switch(config-if)# no mtu	Configures the MTU size. Reverts to the default MTU size (1500 bytes).
Step 3	Switch(config-if)# end	Exits configuration interface mode.
Step 4	Switch(config)# end	Exits configuration mode.
Step 5	Switch# show running-config interface [{fastethernet gigabitethernet} slot/port]	Verifies the running configuration.

1. *type* = fastethernet, gigabitethernet, or tengigabitethernet



Note

When you remove a line card, and then reinsert the card, some or all of the MTU values configured on the ports of that line card may be unconfigured. This occurs if the system wide limit of 32 different MTUs is reached while the card is removed. Upon reinserting the line card, the system attempts to reapply the MTU configuration on the ports. If this attempt fails, the MTU values are set to the default.



Note

When configuring the MTU size for VLAN interfaces and Layer 3 and Layer 2 Ethernet ports, note that the supported MTU values are from 1500 to 9198 bytes.

This example shows how to configure the MTU size on Gigabit Ethernet port 1/1:

```
switch# conf terminal
switch(config)# interface gil1/1
switch(config-if)# mtu 9198
switch(config-if)# end
switch(config)# end
switch# show interface gigabitethernet 1/2
GigabitEthernet1/2 is administratively down, line protocol is down
  Hardware is C6k 1000Mb 802.3, address is 0030.9629.9f88 (bia 0030.9629.9f88)
  MTU 9216 bytes, BW 1000000 Kbit, DLY 10 usec,
<...Output Truncated...>
switch#
```

Interacting with Baby Giants

The baby giants feature, introduced in Cisco IOS Release 12.1(12c)EW, uses the global command **system mtu size** to set the global baby giant MTU. This feature also allows certain interfaces to support Ethernet payload size of up to 1552 bytes.

Both the **system mtu** command and the per-interface **mtu** command can operate on interfaces that can support jumbo frames, but the per-interface **mtu** command takes precedence.

For example, before setting the per-interface MTU for interface gi1/1, you enter the **system mtu 1550** command to change the MTU for gi1/1 to 1550 bytes. You enter the per-interface **mtu** command to change the MTU for gi1/1 to 9198 bytes. If you change the baby giant MTU to 1540 bytes with the command **system mtu 1540**, the MTU for gi1/1 remains unchanged at 9198 bytes.

Configuring the Port Debounce Timer



Note

You can only configure port debounce on 10-Gigabit Ethernet ports.

The port debounce timer suppresses notification of short link-down events. Link-down events that are shorter than the port debounce timer are not notified to Layer 2 or Layer 3 protocols, decreasing traffic loss due to network reconfiguration. You can configure the port debounce timer separately on each LAN port.



Caution

Enabling the port debounce timer causes a delay in link down detections, resulting in loss of traffic during the debouncing period. This situation might affect the convergence and reconvergence of some Layer 2 and Layer 3 protocols.

To configure the debounce timer on a port, perform this task:

	Command	Purpose
Step 1	Switch(config)# interface tengigabitethernet slot/port	Selects the port to configure.
Step 2	Switch(config-if)# link debounce [time debounce_time]	Configures the debounce timer.
	Switch(config-if)# no link debounce	Reverts to the default setting.
Step 3	Switch# show interfaces debounce	Verifies the configuration.



Note

The default time is 10ms for E-series supervisor engines and line cards.

When configuring the debounce timer on a port, you can increase the port debounce timer value between 10 milliseconds and 5000 milliseconds on the 10-Gigabit Ethernet ports.

This example shows how to enable the port debounce timer on 10-Gigabit Ethernet port 2/1 and to accept the default value (10 ms):

```
Switch# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

```
Switch(config)# interface tenGigabitEthernet 2/1
Switch(config-if)# link debounce
Warning: Enabling debounce feature causes link down detection to be delayed
Switch(config-if)# exit
```

This example shows how to enable the port debounce timer of 5000 ms on 10-Gigabit Ethernet port 2/2 and to verify the setting:

```
Switch# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface tenGigabitEthernet 2/2
Switch(config-if)# link debounce time 5000
Warning: Enabling debounce feature causes link down detection to be delayed
Switch(config-if)# end
Switch#
Switch# show interfaces debounce | include enable
Te2/1      enable      10
Te2/2      enable      5000
Switch#
```

Configuring Auto-MDIX on a Port

When automatic medium-dependent-interface crossover (auto-MDIX) is enabled on an port, the port automatically detects the required cable connection type (straight through or crossover) and configures the connection appropriately. When connecting switches without the auto-MDIX feature, you must use straight-through cables to connect to devices such as servers, workstations, or routers and crossover cables to connect to other switches or repeaters. With auto-MDIX enabled, use either type of cable to connect to other devices; the interface automatically corrects for any incorrect cabling. For more information about cabling requirements, see the hardware installation guide.

Auto-MDIX is enabled by default. When you enable auto-MDIX, you must also set the speed on the port to **auto** so that for the feature to operate correctly. auto-MDIX is supported on copper media ports. It is not supported on fiber media ports.



Note

The following line cards support Auto-MDIX by default, when port auto-negotiation is enabled: WS-X4424-GB-RJ45, WS-X4448-GB-RJ45, WS-X4548-GB-RJ45 and WS-X4412-2GB-T. You cannot disable them with the **mdix** command.



Note

The following line cards do not support Auto-MDIX, neither by default nor by CLI: WS-X4548-GB-RJ45V, WS-X4524-GB-RJ45V, WS-X4506-GB-T, WS-X4148-RJ, WS-X4248-RJ21V, WS-X4248-RJ45V, WS-X4224-RJ45V and WS-X4232-GB-RJ.



Note

The following line cards support Auto-MDIX through the CLI on their copper media ports: WS-X4124-RJ45, WS-X4148-RJ45 (hardware revision 3.0 or higher), and WS-X4232-GB-RJ45 (hardware revision 3.0, or higher), WS-4648-RJ45V+E, WS-X4748-UPOE+E and WS-X4748-RJ45+E (Auto-MDIX support when inline power is disabled on the port).

Table 9-8 shows the link states that results from auto-MDIX settings and correct and incorrect cabling.

Table 9-8 Link Conditions and auto-MDIX Settings

Local Side auto-MDIX	Remote Side auto-MDIX	With Correct Cabling	With Incorrect Cabling
On	On	Link up	Link up
On	Off	Link up	Link up
Off	On	Link up	Link up
Off	Off	Link up	Link down

To configure auto-MDIX on a port, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode
Step 2	Switch(config)# interface <i>interface-id</i>	Enters interface configuration mode for the physical interface to be configured.
Step 3	Switch(config-if)# speed auto	Configures the port to autonegotiate speed with the connected device.
Step 4	Switch(config-if)# mdix auto	Enables auto-MDIX on the port.
Step 5	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 6	Switch# show interfaces <i>interface-id</i>	Verifies the configuration of the auto-MDIX feature on the interface.
Step 7	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To disable auto-MDIX, use the **no mdix auto** interface configuration command.

This example shows how to enable auto-MDIX on a port:

```
Switch# configure terminal
Switch(config)# interface fastethernet 6/5
Switch(config-if)# speed auto
Switch(config-if)# mdix auto
Switch(config-if)# end
```

Displaying the Interface Auto-MDIX Configuration

To display the interface speed and duplex mode configuration for an interface, perform this task:

	Command	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	Switch# show interfaces type slot/interface	Displays the interface auto-MDIX configuration setting and operational state.

Depending on how the **speed auto** and the **mdix auto** commands are configured on a supported line card interface, the **show interfaces** command displays the following possible auto-MDIX statuses:

[Table 9-9](#) shows the auto-MDIX setting and operational state and the status of auto-MDIX.

Table 9-9 Auto-MDIX and Operational State

Auto-MDIX Setting and Operational State on an Interface	Description
Auto-MDIX on (operational: on)	Auto-MDIX is enabled and is fully functioning.
Auto-MDIX on (operational: off)	Auto-MDIX is enabled on this interface but it is not functioning. To allow auto-MDIX feature to function properly, you must also set the interface speed to be autonegotiated.
Auto-MDIX off	Auto-MDIX has been disabled with the no mdix auto command.

This example shows how to display the auto-MDIX configuration setting and its operational state on Fast Ethernet interface 6/1:

```
Switch# show interfaces fastethernet 6/1
FastEthernet6/1 is up, line protocol is up (connected)
  Hardware is Fast Ethernet Port, address is 0001.64fe.e5d0 (bia 0001.64fe.e5d0)
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 100Mb/s, link type is auto, media type is 10/100BaseTX
  input flow-control is unsupported output flow-control is unsupported
  Auto-MDIX on (operational: on)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:16, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/2000/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    511 packets input, 74464 bytes, 0 no buffer
    Received 511 broadcasts (511 multicasts)
    0 runs, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
    0 input packets with dribble condition detected
    3552 packets output, 269088 bytes, 0 underruns
    0 output errors, 0 collisions, 0 interface resets
    0 babbles, 0 late collision, 0 deferred
    1 lost carrier, 0 no carrier
    0 output buffer failures, 0 output buffers swapped out
Switch#
```

Understanding Online Insertion and Removal

The online insertion and removal (OIR) feature supported on the Catalyst 4500 series switch allows you to remove and replace modules while the system is online. You can shut down the module before removal and restart it after insertion without causing other software or interfaces to shut down.

You do not need to enter a command to notify the software that you are going to remove or install a module. The system notifies the supervisor engine that a module has been removed or installed and scans the system for a configuration change. The OIR process may take up to 70 seconds before the newly

installed module is initialized. Each interface type is verified against the system configuration, after which the system runs diagnostics on the new interface. There is no disruption to normal operation during module insertion or removal.

If you remove a module and then replace it, or insert a different module of the same type into the same slot, no change to the system configuration is needed. An interface of a type that has been configured previously is brought online immediately. If you remove a module and insert a module of a different type, the interface(s) on that module is administratively up with the default configuration for that module.

Online Insertion and Removal on a WS-4500X-32

You must initiate an uplink module removal procedure on a WS-4500X-32 (the Catalyst 4500-X Series Switch) either through a CLI (below) or through the OIR button on the front panel of the uplink.



Note

Unscheduled uplink module removal is not supported because the C4KX-NM-8 module has hardware communication lines to the baseboard that cannot be unplugged in an online state. Communication must be stopped first.

Before removing the uplink module, you must press the OIR button for 5 seconds, and wait for the OIR LED to turn Green. Removing a module without pressing the OIR button causes the system to reboot to ROMMON with the following console error message:

```
Kernel panic - not syncing: Removing a module from switch causes instability.
Rebooting in 15 seconds...
```



Note

No special steps are needed for insertion.

With Cisco Release IOS XE 3.3.0SG and IOS 15.1(1)SG, two CLI's are introduced: **hw-module module number start** and **hw-module module number stop**. For the *number* keyword, the only applicable value for WS-C4500 is 2. With Cisco Release IOS XE 3.3.0SG and IOS 15.1(1)SG, the **start** and **stop** commands are only enabled on the uplink module of WS-4500X-32.

For details, please refer to the hardware portion of the documentation library:

http://www.cisco.com/en/US/products/hw/switches/ps4324/prod_installation_guides_list.html

Shutting down a Module

To shut down a module safely, either enter the **hw-module module stop** command or press the OIR button for 5 seconds.



Note

The **hw-module module stop** command is enabled only on the uplink modules of the WS-C4500X-32.

The following example shows what happens if a module is up and you enter the **hw-module module stop** command:

```
Switch# hw-module module 2 stop
Proceed with module stop? [confirm]
Switch#
*Feb 5 16:34:37.325: %C4K_IOSMODPORTMAN-6-MODULEOFFLINE: Module 2 is offline
Switch# show module
Chassis Type : WS-C4500X-32
```


Power consumed by backplane : 0 Watts

Mod	Ports	Card Type	Model	Serial No.
1	32	4500X-32 10GE (SFP+)	WS-C4900X-32P-10G	JAE153505E9
2	8	Module being held in reset	WS-X4908X-10G-TIM	JAE15340C0J

M	MAC addresses	Hw	Fw	Sw	Status
1	0022.bde2.1061 to 0022.bde2.1080	0.2	15.0(1r)SG(0	0.DEV-0	Ok
2	0022.bde2.1579 to 0022.bde2.1580	0.1			In Reset

Switch#

The following example shows what happens if a module is already stopped and you enter the **hw-module module stop** command:

```
Switch# hw-module module 2 stop
% Module 2 stopped
```

Booting a Module After if it has been Stopped

To bring up a module that has been stopped using the **hw-module module number stop** command or by pressing the **OIR button**, you either enter the **hw-module module number start** command or physically remove and reinsert.

The following example shows what happens if a module has been stopped and you enter this command:

```
Switch# hw-module module 2 start
Switch#
*Feb 5 16:36:27.352: %C4K_IOSMODPORTMAN-6-MODULEINSERTED: Module 2 is inserted
*Feb 5 16:37:15.902: %C4K_IOSMODPORTMAN-6-MODULEONLINE: Module 2 (WS-X4908X-10G-TIM S/N:
JAE15340C0J Hw: 0.1) is online
Switch# show module
Chassis Type : WS-C4500X-32
```

Power consumed by backplane : 0 Watts

Mod	Ports	Card Type	Model	Serial No.
1	32	4500X-32 10GE (SFP+)	WS-C4900X-32P-10G	JAE153505E9
2	8	10GE SFP+	WS-X4908X-10G-TIM	JAE15340C0J

M	MAC addresses	Hw	Fw	Sw	Status
1	0022.bde2.1061 to 0022.bde2.1080	0.2	15.0(1r)SG(0	0.DEV-0	Ok
2	0022.bde2.1579 to 0022.bde2.1580	0.1			Ok

Switch#

The following example shows what happens if a module has not been stopped and you enter this command:

```
Switch# hw-module module 2 start
% Module 2 not stopped
```

Common Scenarios

Table 9-10 lists the common scenarios associated with an OIR on a WS-4500X-32.

Table 9-10 Common Scenarios for OIR on a WS-4500X-32

If you do this or want to do this...	This happens or you need to do this....
Insert a new module for the first time.	The behavior of new module matches that of current linecard.
Pull out an uplink module that was not previously stopped or shut down.	The system reboots to ROMMON.
Press the OIR button accidentally.	Press for less than 5 sec and nothing happens. Press for more than 5 sec and OIR is initiated. The linecard moves to the reset state and the OIR LED turns GREEN.
Change your mind after entering the module stop command or pressing the OIR button.	Enter the module start command or perform a physical OIR. Both actions trigger uplink module restart. If the module is not faulty, it is restored to online state.
Know if the OIR button has taken effect.	The OIR LED on the uplink module turns GREEN and the linecard status LED on the uplink module turns off.
Disable the OIR button and force use of a CLI to initiate an OIR.	Cannot be done.

Monitoring and Maintaining the Interface

The following sections describe how to monitor and maintain the interfaces:

- [Monitoring Interface and Controller Status, page 9-50](#)
- [Clearing and Resetting the Interface, page 9-51](#)
- [Shutting Down and Restarting an Interface, page 9-51](#)
- [Configuring Interface Link Status and Trunk Status Events, page 9-52](#)
- [Resetting the Interface to the Default Configuration, page 9-55](#)

Monitoring Interface and Controller Status

The Cisco IOS software for the Catalyst 4500 series switch contains commands that you can enter at the EXEC prompt to display information about the interface, including the version of the software and the hardware, the controller status, and statistics about the interfaces. The following table lists some of the interface monitoring commands. (You can display the full list of **show** commands by entering the **show ?** command at the EXEC prompt.)

To display information about the interface, enter one of the following commands:

Command	Purpose
Switch# show interfaces [<i>type slot/interface</i>]	Displays the status and configuration of all interfaces or of a specific interface.
Switch# show running-config	Displays the configuration currently running in RAM.
Switch# show protocols [<i>type slot/interface</i>]	Displays the global (system-wide) and interface-specific status of any configured protocol.
Switch# show version	Displays the hardware configuration, software version, the names and sources of configuration files, and the boot images.

This example shows how to display the status of Fast Ethernet interface 5/5:

```
Switch# show protocols fastethernet 5/5
FastEthernet5/5 is up, line protocol is up
Switch#
```

Clearing and Resetting the Interface

To clear the interface counters shown with the **show interfaces** command, enter this command:

Command	Purpose
Switch# clear counters [<i>type slot/interface</i>]	Clears interface counters.

This example shows how to clear and reset the counters on Fast Ethernet interface 5/5:

```
Switch# clear counters fastethernet 5/5
Clear "show interface" counters on this interface [confirm] y
Switch#
*Sep 30 08:42:55: %CLEAR-5-COUNTERS: Clear counter on interface FastEthernet5/5
by vty1 (171.69.115.10)
Switch#
```

The **clear counters** command (without any arguments) clears all the current interface counters from all interfaces.



Note

The **clear counters** command does not clear counters retrieved with SNMP; it clears only those counters displayed with the EXEC **show interfaces** command.

Shutting Down and Restarting an Interface

You can disable an interface, which disables all functions on the specified interface and marks the interface as unavailable on all monitoring command displays. This information is communicated to other network servers through all dynamic routing protocols. The interface is not mentioned in any routing updates.

To shut down an interface and then restart it, perform this task:

	Command	Purpose
Step 1	Switch(config)# interface { vlan <i>vlan_ID</i> } { fastethernet gigabitethernet tengigabitethernet } <i>slot/port</i> } { port-channel <i>port_channel_number</i> }	Specifies the interface to be configured.
Step 2	Switch(config-if)# shutdown	Shuts down the interface.
Step 3	Switch(config-if)# no shutdown	Reenables the interface.

This example shows how to shut down Fast Ethernet interface 5/5:

```
Switch(config)# interface fastethernet 5/5
Switch(config-if)# shutdown
Switch(config-if)#
*Sep 30 08:33:47: %LINK-5-CHANGED: Interface FastEthernet5/5, changed state to a
administratively down
Switch(config-if)#
```

This example shows how to reenabale Fast Ethernet interface 5/5:

```
Switch(config-if)# no shutdown
Switch(config-if)#
*Sep 30 08:36:00: %LINK-3-UPDOWN: Interface FastEthernet5/5, changed state to up
Switch(config-if)#
```

To verify whether an interface is disabled, enter the EXEC **show interfaces** command. An interface that has been shut down appears as “administratively down.”

Configuring Interface Link Status and Trunk Status Events

You can configure interface link status and trunk status events. On the Catalyst 4500 series switch, the following interface logging event notifications are supported both globally and per interface:

- Enable or disable notification on the interface whenever its data link status is changed.
- Enable or disable notification on the trunk interface whenever its trunking status is changed.

Use the **[no] logging event link-status use-global** command to enable or disable the interface link status event. Use the **[no] logging event trunk-status use-global** command to enable or disable the interface trunk status event.

Each interface link status logging event can be configured in one of the following states:

- **logging event link-status**—Link status logging event is enabled explicitly on the interface regardless of the switch global setting.
- **no logging event link-status**—Link status logging event is disabled explicitly on the interface regardless of the switch global setting.
- **logging event link-status use-global**—Default link status logging event configuration on the interface; its configuration should follow the switch global link status logging event setting.

The interface trunk status logging event can be configured in the same configuration states.

Configuring Link Status Event Notification for an Interface

To enable or disable a link status logging event, enter one of the following commands:

Command	Purpose
Switch(config-if)# logging event link-status	Enables interface link status logging.
Switch(config-if)# no logging event link-status	Disables interface link status logging.
Switch(config-if)# logging event link-status use-global	Specifies the global default setting for interface link status logging.

Global Settings

You can also provide a global configuration for the corresponding logging event. A global configuration provides default logging settings for all interfaces. The **[no] logging event link-status global** command lets you enable or disable the interface link status logging for the entire switch. The **[no] logging event trunk-status global** command lets you enable/disable interface trunk status logging for the entire switch.

Each interface link status logging event, if not configured at the interface level, uses the following global logging event setting:

- **logging event link-status global**—Link status logging event is enabled, if not configured on the interface.
- **no logging event link-status global**—Link status logging event is disabled, if not configured on the interface.

The interface trunk status logging event has similar global configurations.

Configuring a Switch Global Link Status Logging Event

To enable or disable the global link status logging event, enter one of the following commands:

Command	Purpose
Switch(config-if)# logging event link-status global	Enables global link status logging.
Switch(config-if)# no logging event link-status global	Disables global link status logging.

Examples

The following example displays a summary of the operating states for the interface logging event using different combinations of global and interface logging settings:

global setting	interface setting	actual logging state
on	on	on
off	on	on
on	off	off
off	off	off
on	default (use-global)	on
off	default (use-global)	off

The following example displays the configuration and logging message output for link status and trunk status logging events:

```
//
// The global link status and trunk status logging events are enabled.
//
Switch# show running | include logging
show running | include logging
logging event link-status global
logging event trunk-status global
Switch#

//
// The interface link status and trunk status logging settings
// are set to default values, which follow regardless of the global
// setting.
//
Switch# show running interface g1/4
Building configuration...

Current configuration: 97 bytes
!
interface GigabitEthernet1/4
 switchport trunk encapsulation dot1q
 switchport mode trunk
end
Switch#

//
// The trunk status logging messages for the interface are
// displayed whenever the interface trunking status is changed.
// Here we change the other end node's trunking encapsulation
// from dot1q to isl.
//
3d00h: %DTP-5-ILGLCFG: Illegal config(on,isl--on,dot1q) on Gi1/4
3d00h: %DTP-5-ILGLCFG: Illegal config(on,isl--on,dot1q) on Gi1/4
3d00h: %DTP-5-ILGLCFG: Illegal config(on,isl--on,dot1q) on Gi1/4

//
// The link and trunk status logging message for the interface
// are displayed whenever the interface link status is changed.
// Here we do a "shut" and "no shut" on the other end link node.
//
3d00h: %DTP-5-NONTRUNKPORTON: Port Gi1/4 has become non-trunk
3d00h: %LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet1/4, changed state to down
3d00h: %LINK-3-UPDOWN: Interface GigabitEthernet1/4, changed state to
down
3d00h: %LINK-3-UPDOWN: Interface GigabitEthernet1/4, changed state to up
3d00h: %DTP-5-TRUNKPORTON: Port Gi1/4 has become dot1q trunk
3d00h: %LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet1/4, changed state to up
```

Resetting the Interface to the Default Configuration

If you have configured a interface with many command lines and you want to clear all the configuration on that interface, use the **default interface** global configuration command, as follows:

```
Switch(config)# default interface fastEthernet 3/5  
Interface FastEthernet3/5 set to default configuration
```

This command clears all the configurations and shut down the interface:

```
Switch# show run interface fastethernet 3/5  
Building configuration...  
  
Current configuration : 58 bytes  
!  
interface FastEthernet3/5  
  no ip address  
  shutdown  
end
```




Checking Port Status and Connectivity

This chapter describes how to check switch port status and connectivity on the Catalyst 4500 series switch.

This chapter includes the following major sections:

- [Checking Module Status, page 10-1](#)
- [Checking Interfaces Status, page 10-2](#)
- [Displaying MAC Addresses, page 10-3](#)
- [Checking Cable Status Using Time Domain Reflectometer, page 10-3](#)
- [Using Telnet, page 10-5](#)
- [Changing the Logout Timer, page 10-6](#)
- [Monitoring User Sessions, page 10-6](#)
- [Using Ping, page 10-7](#)
- [Using IP Traceroute, page 10-8](#)
- [Using Layer 2 Traceroute, page 10-9](#)
- [Configuring ICMP, page 10-11](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

Checking Module Status

The Catalyst 4500 series switch is a multimodule system. You can see which modules are installed, as well as the MAC address ranges and version numbers for each module, by entering the **show module** command. Use the *mod_num* argument to specify a particular module number and display detailed information on that module.

This example shows how to check module status for all modules on your switch:

```
Switch# show module all
```

Mod	Ports	Card Type	Model	Serial No.
1	2	1000BaseX (GBIC) Supervisor Module	WS-X4014	JAB012345AB
5	24	10/100/1000BaseTX (RJ45)	WS-X4424-GB-RJ45	JAB045304EY
6	48	10/100BaseTX (RJ45)	WS-X4148	JAB023402QK

M	MAC addresses	Hw	Fw	Sw	Stat
1	0004.dd46.9f00 to 0004.dd46.a2ff	0.0	12.1(10r)EW(1.21)	12.1(10)EW(1)	Ok
5	0050.3e7e.1d70 to 0050.3e7e.1d87	0.0			Ok
6	0050.0f10.2370 to 0050.0f10.239f	1.0			Ok

```
Switch#
```

Checking Interfaces Status

You can view summary or detailed information on the switch ports using the **show interfaces status** command. To see summary information on all ports on the switch, enter the **show interfaces status** command with no arguments. Specify a particular module number to see information on the ports on that module only. Enter both the module number and the port number to see detailed information about the specified port.

To apply configuration commands to a particular port, you must specify the appropriate logical module. For more information, see the [“Checking Module Status” section on page 10-1](#).

This example shows how to display the status of all interfaces on a Catalyst 4500 series switch, including transceivers. Output of this command displays “Unapproved GBIC” for non-Cisco transceivers:

```
Switch# show interfaces status
```

Port	Name	Status	Vlan	Duplex	Speed	Type
Gi1/1		notconnect	1	auto	auto	No Gbic
Gi1/2		notconnect	1	auto	auto	No Gbic
Gi5/1		notconnect	1	auto	auto	10/100/1000-TX
Gi5/2		notconnect	1	auto	auto	10/100/1000-TX
Gi5/3		notconnect	1	auto	auto	10/100/1000-TX
Gi5/4		notconnect	1	auto	auto	10/100/1000-TX
Fa6/1		connected	1	a-full	a-100	10/100BaseTX
Fa6/2		connected	2	a-full	a-100	10/100BaseTX
Fa6/3		notconnect	1	auto	auto	10/100BaseTX
Fa6/4		notconnect	1	auto	auto	10/100BaseTX

```
Switch#
```

This example shows how to display the status of interfaces in error-disabled state:

```
Switch# show interfaces status err-disabled
```

Port	Name	Status	Reason
Fa9/4		err-disabled	link-flap

informational error message when the timer expires on a cause

```
-----
5d04h:%PM-SP-4-ERR_RECOVER:Attempting to recover from link-flap err-disable state on Fa9/4
Switch#
```

Displaying MAC Addresses

In addition to displaying the MAC address range for a module using the **show module** command, you can display the MAC address table information of a specific MAC address or a specific interface in the switch using the **show mac-address-table address** and **show mac-address-table interface** commands.

This example shows how to display MAC address table information for a specific MAC address:

```
Switch# show mac-address-table address 0050.3e8d.6400
```

vlan	mac address	type	protocol	qos	ports
200	0050.3e8d.6400	static	assigned	--	Switch
100	0050.3e8d.6400	static	assigned	--	Switch
5	0050.3e8d.6400	static	assigned	--	Switch
4	0050.3e8d.6400	static	ipx	--	Switch
1	0050.3e8d.6400	static	ipx	--	Switch
1	0050.3e8d.6400	static	assigned	--	Switch
4	0050.3e8d.6400	static	assigned	--	Switch
5	0050.3e8d.6400	static	ipx	--	Switch
100	0050.3e8d.6400	static	ipx	--	Switch
200	0050.3e8d.6400	static	ipx	--	Switch
100	0050.3e8d.6400	static	other	--	Switch
200	0050.3e8d.6400	static	other	--	Switch
5	0050.3e8d.6400	static	other	--	Switch
4	0050.3e8d.6400	static	ip	--	Switch
1	0050.3e8d.6400	static	ip	--	Route
1	0050.3e8d.6400	static	other	--	Switch
4	0050.3e8d.6400	static	other	--	Switch
5	0050.3e8d.6400	static	ip	--	Switch
200	0050.3e8d.6400	static	ip	--	Switch
100	0050.3e8d.6400	static	ip	--	Switch

Switch#

This example shows how to display MAC address table information for a specific interface:

```
Switch# show mac-address-table interface gigabit 1/1
```

Multicast Entries

vlan	mac address	type	ports
1	ffff.ffff.ffff	system	Switch, Gi6/1, Gi6/2, Gi6/9, Gi1/1

Switch#

Checking Cable Status Using Time Domain Reflectometer

The Time Domain Reflectometer (TDR) feature allows you to determine if cable is OPEN or SHORT when it is at fault.

Overview

With TDR, you can check the status of copper cables on the 48-port 10/100/1000 BASE-T modules for the Catalyst 4500 series switch. TDR detects a cable fault by sending a signal through the cable and reading the signal that is reflected back. All or part of the signal can be reflected back either by cable defects or by the end of the cable.

**Note**

Four pairs of standard category 5 cable exist. Each pair can assume one of the following states: open (not connected), broken, shorted, or terminated. The TDR test detects all four states and displays the first three as “Fault” conditions, and displays the fourth as “Terminated.” Although the CLI output is shown, the cable length is displayed only if the state is “Faulty.”

TDR feature is supported on the following modules:

WS-X4524-GB-RJ45V

WS-X4548-GB-RJ45

WS-X4548-GB-RJ45V

WS-X4548-GB-RJ45V+

WS-X4548-RJ45V+

WS-X4748-RJ45+E

WS-X4748-RJ45V+E

WS-X4748-UPOE+E

TDR detects a cable fault by sending a signal along its wires and depending on the reflected signal it can determine roughly where a cable fault could be. The variations on how TDR signal is reflected back determine the results on TDR. On cat4k products, we only support cable fault types: OPEN or SHORT. We do display Terminated status in case cable is proper terminated and this is done for illustrative purpose.

Running the TDR Test

To start the TDR test, perform this task:

	Command	Purpose
Step 1	Switch# test cable-diagnostics tdr { interface { <i>interface interface-number</i> }}	Starts the TDR test.
Step 2	Switch# show cable-diagnostics tdr { interface { <i>interface interface-number</i> }}	Displays the TDR test counter information.

This example shows how to start the TDR test on port 1 on module 2:

```
Switch# test cable-diagnostics tdr int gi2/1
Switch#
```

This example shows the message that displays when the TDR test is not supported on a module:

```
Switch# test cable-diagnostics tdr int gi2/1
00:03:15:%C4K_IOSDIAGMAN-4-TESTNOTSUPPORTEDONMODULE: Online cable
diag tdr test is not supported on this module
Switch#
```

This example shows how to display TDR test results for a port:

```
Switch# show cable-diagnostics tdr interface gi4/13
Interface Speed Local pair Cable length Remote channel Status
Gi4/13    0Mbps   1-2      102 +-2m    Unknown      Fault
           3-6      100 +-2m    Unknown      Fault
           4-5      102 +-2m    Unknown      Fault
```

7-8 102 +-2m Unknown Fault

**Note**

After this command is deprecated, use the diagnostic start and the **show diagnostic result** commands to run the TDR test and display the test results.

**Note**

TDR is a port test; the port cannot handle traffic for the duration of the test (generally, 1 minute).

TDR Guidelines

The following guidelines apply to the use of TDR:

- If you connect a port undergoing a TDR test to an Auto-MDIX enabled port, the TDR result might be invalid. In those instances, the port on the WS-X4148-RJ45V should be administratively down before the start of the TDR test.
- If you connect a port undergoing a TDR test to a 100BASE-T port such as that on the WS-X4148-RJ45V, the unused pairs (4-5 and 7-8) is reported as faulty because the remote end does not terminate these pairs.
- Do not change the port configuration while the TDR test is running.
- Due to cable characteristics, you should run the TDR test multiple times to get accurate results.
- Do not change port status (for example, remove the cable at the near or far end) because the results might be inaccurate.
- TDR works best if the test cable is disconnected from the remote port. Otherwise, it might be difficult for you to interpret results correctly.
- TDR operates across four wires. Depending on the cable conditions, the status might show one pair is OPEN or SHORT while all other wire pairs display as faulty. This operation is acceptable because you should declare a cable faulty provided one pair of wires is either OPEN or SHORT.
- TDR intent is to determine how poorly a cable is functioning rather than to locate a faulty cable.
- When TDR locates a faulty cable, you should still use an offline cable diagnosis tool to better diagnose the problem.
- TDR results might differ between runs on different Catalyst 4500 modules because of the resolution difference of TDR implementations. When this occurs, you should refer to offline cable diagnosis tool.

Using Telnet

You can access the switch command-line interface (CLI) using Telnet. In addition, Telnet allows you to access other devices in the network. You can have up to eight simultaneous Telnet sessions.

Before you can open a Telnet session to the switch, you must first set the IP address (and in some cases the default gateway) for the switch. For information about setting the IP address and default gateway, see [Chapter 3, “Configuring the Switch for the First Time.”](#)

**Note**

To establish a Telnet connection to a host by using the hostname, configure and enable DNS.

To establish a Telnet connection to another device on the network from the switch, enter this command:

Command	Purpose
Switch# telnet <i>host</i> [<i>port</i>]	Opens a Telnet session to a remote host.

This example shows how to establish a Telnet connection from the switch to the remote host named labsparc:

```
Switch# telnet labsparc
Trying 172.16.10.3...
Connected to labsparc.
Escape character is '^]'.

UNIX(r) System V Release 4.0 (labsparc)

login:
```

Changing the Logout Timer

The logout timer automatically disconnects a user from the switch when the user is idle for longer than the specified time. To set the logout timer, enter this command:

Command	Purpose
Switch# logoutwarning <i>number</i>	Changes the logout timer value (a timeout value of 0 prevents idle sessions from being disconnected automatically). Use the no keyword to return to the default value.

Monitoring User Sessions

You can display the currently active user sessions on the switch using the **show users** command. The command output lists all active console port and Telnet sessions on the switch.

To display the active user sessions on the switch, enter this command:

Command	Purpose
Switch# show users [<i>all</i>]	Displays the currently active user sessions on the switch.

This example shows the output of the **show users** command when local authentication is enabled for console and Telnet sessions (the asterisk [*] indicates the current session):

```
Switch# show users
  Line      User      Host(s)      Idle      Location
*  0 con 0      idle          00:00:00

  Interface      User      Mode          Idle      Peer Address

Switch# show users all
  Line      User      Host(s)      Idle      Location
```

```

* 0 con 0          idle          00:00:00
  1 vty 0          00:00:00
  2 vty 1          00:00:00
  3 vty 2          00:00:00
  4 vty 3          00:00:00
  5 vty 4          00:00:00

Interface      User      Mode      Idle      Peer Address
Switch#

```

To disconnect an active user session, enter this command:

Command	Purpose
Switch# disconnect { console <i>ip_addr</i> }	Disconnects an active user session on the switch.

This example shows how to disconnect an active console port session and an active Telnet session:

```

Switch> disconnect console
Console session disconnected.
Console> (enable) disconnect tim-nt.bigcorp.com
Telnet session from tim-nt.bigcorp.com disconnected. (1)
Switch# show users
  Session  User      Location
  -----  -
telnet    jake      jake-mac.bigcorp.com
* telnet  suzy      suzy-pc.bigcorp.com
Switch#

```

Using Ping

These sections describe how to use IP ping:

- [Understanding How Ping Works, page 10-7](#)
- [Running Ping, page 10-8](#)

Understanding How Ping Works

The **ping** command allows you to verify connectivity to remote hosts. If you attempt to ping a host in a different IP subnetwork, you must define a static route to the network or configure a router to route between those subnets.

The **ping** command is configurable from normal executive and privileged EXEC mode. Ping returns one of the following responses:

- Normal response—The normal response (*hostname* is alive) occurs in 1 to 10 seconds, depending on network traffic.
- Destination does not respond—If the host does not respond, a No Answer message is returned.
- Unknown host—If the host does not exist, an Unknown Host message is returned.
- Destination unreachable—If the default gateway cannot reach the specified network, a Destination Unreachable message is returned.
- Network or host unreachable—If there is no entry in the route table for the host or network, a Network or Host Unreachable message is returned.

To stop a ping in progress, press **Ctrl-C**.

Running Ping

To ping another device on the network from the switch, enter this command in normal executive and privileged EXEC mode:

Command	Purpose
Switch# ping <i>host</i>	Checks connectivity to a remote host.

This example shows how to ping a remote host from normal executive mode:

```
Switch# ping labsparc
labsparc is alive
Switch> ping 72.16.10.3
12.16.10.3 is alive
Switch#
```

This example shows how to use a **ping** command in privileged EXEC mode to specify the number of packets, the packet size, and the timeout period:

```
Switch# ping
Target IP Address []: 12.20.5.19
Number of Packets [5]: 10
Datagram Size [56]: 100
Timeout in seconds [2]: 10
Source IP Address [12.20.2.18]: 12.20.2.18
!!!!!!!!!!!!

----12.20.2.19 PING Statistics----
10 packets transmitted, 10 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 1/1/1
Switch
```

Using IP Traceroute

These sections describe how to use IP traceroute feature:

- [Understanding How IP Traceroute Works, page 10-8](#)
- [Running IP Traceroute, page 10-9](#)

Understanding How IP Traceroute Works

IP traceroute allows you to identify the path that packets take through the network on a hop-by-hop basis. The command output displays all network layer (Layer 3) devices, such as routers, that the traffic passes through on the way to the destination.

Layer 2 switches can participate as the source or destination of the **trace** command but does not appear as a hop in the **trace** command output.

The **trace** command uses the time to live (TTL) field in the IP header to cause routers and servers to generate specific return messages. Traceroute starts by sending a User Datagram Protocol (UDP) datagram to the destination host with the TTL field set to 1. If a router finds a TTL value of 1 or 0, it

drops the datagram and sends back an Internet Control Message Protocol (ICMP) Time-Exceeded message to the sender. Traceroute determines the address of the first hop by examining the source address field of the ICMP Time-Exceeded message.

To identify the next hop, traceroute sends a UDP packet with a TTL value of 2. The first router decrements the TTL field by 1 and sends the datagram to the next router. The second router sees a TTL value of 1, discards the datagram, and returns the Time-Exceeded message to the source. This process continues until the TTL is incremented to a value large enough for the datagram to reach the destination host or until the maximum TTL is reached.

To determine when a datagram reaches its destination, traceroute sets the UDP destination port in the datagram to a very large value that the destination host is unlikely to be using. When a host receives a datagram with an unrecognized port number, it sends an ICMP Port Unreachable error message to the source. The Port Unreachable error message indicates to traceroute that the destination has been reached.

Running IP Traceroute

To trace the path that packets take through the network, enter this command in EXEC or privileged EXEC mode:

Command	Purpose
Switch# trace [<i>protocol</i>] [<i>destination</i>]	Runs IP traceroute to trace the path that packets take through the network.

This example shows how to use the **trace** command to display the route a packet takes through the network to reach its destination:

```
Switch# trace ip ABA.NYC.mil
```

```
Type escape sequence to abort.
```

```
Tracing the route to ABA.NYC.mil (26.0.0.73)
```

```
 0 DEBRIS.CISCO.COM (192.180.1.6) 1000 msec 8 msec 4 msec
 1 BARRNET-GW.CISCO.COM (192.180.16.2) 8 msec 8 msec 8 msec
 2 EXTERNAL-A-GATEWAY.STANFORD.EDU (192.42.110.225) 8 msec 4 msec 4 msec
 3 BB2.SU.BARRNET.NET (192.200.254.6) 8 msec 8 msec 8 msec
 4 SU.ARC.BARRNET.NET (192.200.3.8) 12 msec 12 msec 8 msec
 5 MOFFETT-FLD-MB.in.MIL (192.52.195.1) 216 msec 120 msec 132 msec
 6 ABA.NYC.mil (26.0.0.73) 412 msec 628 msec 664 msec
```

```
Switch#
```

Using Layer 2 Traceroute

The Layer 2 traceroute feature allows the switch to identify the physical path that a packet takes from a source device to a destination device. Layer 2 traceroute supports only unicast source and destination MAC addresses. It determines the path by using the MAC address tables of the switches in the path. When the switch detects a device in the path that does not support Layer 2 traceroute, the switch continues to send Layer 2 trace queries and lets them time out.

If you want the switch to trace the path from a host on a source device to a host on a destination device, the switch can identify only the path from the source device to the destination device. It cannot identify the path that a packet takes from source host to the source device or from the destination device to the destination host.

These sections describe how to use the Layer 2 traceroute feature:

- [Layer 2 Traceroute Usage Guidelines, page 10-10](#)
- [Running Layer 2 Traceroute, page 10-11](#)

Layer 2 Traceroute Usage Guidelines

These are the Layer 2 traceroute usage guidelines:

- CDP must be enabled on all the devices in the network. For Layer 2 traceroute to function properly, do not disable CDP.

If any devices in the physical path are transparent to CDP, the switch cannot identify the path through these devices.



Note For more information about enabling CDP, see [Chapter 32, “Configuring Cisco Discovery Protocol.”](#)

- All switches in the physical path must have IP connectivity. When a switch is reachable from another switch, you can test connectivity by using the **ping** command in privileged EXEC mode.
- The maximum number of hops identified in the path is ten.
- You can enter the **traceroute mac** or the **traceroute mac ip** command in privileged EXEC mode on a switch that is not in the physical path from the source device to the destination device. All switches in the path must be reachable from this switch.
- The **traceroute mac** command output shows the Layer 2 path only when the specified source and destination MAC addresses belong to the same VLAN. If you specify source and destination MAC addresses that belong to different VLANs, the Layer 2 path is not identified, and an error message appears.
- If you specify a multicast source or destination MAC address, the path is not identified, and an error message appears.
- If the source or destination MAC address belongs to multiple VLANs, you must specify the VLAN to which both the source and destination MAC addresses belong. If the VLAN is not specified, the path is not identified, and an error message appears.
- The **traceroute mac ip** command output shows the Layer 2 path when the specified source and destination IP addresses belong to the same subnet. When you specify the IP addresses, the switch uses Address Resolution Protocol (ARP) to associate the IP address with the corresponding MAC address and the VLAN ID.
 - If an ARP entry exists for the specified IP address, the switch uses the associated MAC address and identifies the physical path.
 - If an ARP entry does not exist, the switch sends an ARP query and tries to resolve the IP address. If the IP address is not resolved, the path is not identified, and an error message appears.
- When multiple devices are attached to one port through hubs (for example, multiple CDP neighbors are detected on a port), the Layer 2 traceroute feature is not supported. When more than one CDP neighbor is detected on a port, the Layer 2 path is not identified, and an error message appears.
- This feature is not supported in Token Ring VLANs.

Running Layer 2 Traceroute

To display the physical path that a packet takes from a source device to a destination device, enter either one of these commands:

Command	Purpose
Switch# traceroute mac {source-mac-address} {destination-mac-address}	Runs Layer 2 traceroute to trace the path that packets take through the network.

or

Command	Purpose
Switch# traceroute mac ip {source-mac-address} {destination-mac-address}	Runs IP traceroute to trace the path that packets take through the network.

These examples show how to use the **traceroute mac** and **traceroute mac ip** commands to display the physical path a packet takes through the network to reach its destination:

```
Switch# traceroute mac 0000.0201.0601 0000.0201.0201

Source 0000.0201.0601 found on con6[WS-C2950G-24-EI] (2.2.6.6)
con6 (2.2.6.6) :Fa0/1 => Fa0/3
con5           (2.2.5.5       ) : Fa0/3 => Gi0/1
con1           (2.2.1.1       ) : Gi0/1 => Gi0/2
con2           (2.2.2.2       ) : Gi0/2 => Fa0/1
Destination 0000.0201.0201 found on con2[WS-C3550-24] (2.2.2.2)
Layer 2 trace completed
Switch#

Switch# traceroute mac ip 2.2.66.66 2.2.22.22 detail
Translating IP to mac .....
2.2.66.66 => 0000.0201.0601
2.2.22.22 => 0000.0201.0201

Source 0000.0201.0601 found on con6[WS-C2950G-24-EI] (2.2.6.6)
con6 / WS-C2950G-24-EI / 2.2.6.6 :
      Fa0/1 [auto, auto] => Fa0/3 [auto, auto]
con5 / WS-C2950G-24-EI / 2.2.5.5 :
      Fa0/3 [auto, auto] => Gi0/1 [auto, auto]
con1 / WS-C3550-12G / 2.2.1.1 :
      Gi0/1 [auto, auto] => Gi0/2 [auto, auto]
con2 / WS-C3550-24 / 2.2.2.2 :
      Gi0/2 [auto, auto] => Fa0/1 [auto, auto]
Destination 0000.0201.0201 found on con2[WS-C3550-24] (2.2.2.2)
Layer 2 trace completed.
Switch#
```

Configuring ICMP

Internet Control Message Protocol (ICMP) provides many services that control and manage IP connections. ICMP messages are sent by routers or access servers to hosts or other routers when a problem is discovered with the Internet header. For detailed information on ICMP, refer to RFC 792.

Enabling ICMP Protocol Unreachable Messages

If the Cisco IOS software receives a nonbroadcast packet that uses an unknown protocol, it sends an ICMP Protocol Unreachable message back to the source.

Similarly, if the software receives a packet that it is unable to deliver to the ultimate destination because it knows of no route to the destination address, it sends an ICMP Host Unreachable message to the source. This feature is enabled by default.

To enable the generation of ICMP Protocol Unreachable and Host Unreachable messages, enter the following command in interface configuration mode:

Command	Purpose
Switch (config-if)# [no] ip unreachable	Enables ICMP destination unreachable messages. Use the no keyword to disable the ICMP destination unreachable messages.



Caution

If you enter the **no ip unreachable** command, you will break the path MTU discovery functionality. Routers in the middle of the network might be forced to fragment packets.

To limit the rate that Internet Control Message Protocol (ICMP) destination unreachable messages are generated, enter the following command:

Command	Purpose
Switch (config)# [no] ip icmp rate-limit unreachable [df] milliseconds	Limits the rate that ICMP destination messages are generated. Use the no keyword to remove the rate limit and reduce the CPU usage.

Enabling ICMP Redirect Messages

Data routes are sometimes less than optimal. For example, it is possible for the router to be forced to resend a packet through the same interface on which it was received. If this occurs, the Cisco IOS software sends an ICMP Redirect message to the originator of the packet telling the originator that the router is on a subnet directly connected to the receiving device, and that it must forward the packet to another system on the same subnet. The software sends an ICMP Redirect message to the packet's originator because the originating host presumably could have sent that packet to the next hop without involving this device at all. The Redirect message instructs the sender to remove the receiving device from the route and substitute a specified device representing a more direct path. This feature is enabled by default.

However, when Hot Standby Router Protocol (HSRP) is configured on an interface, ICMP Redirect messages are disabled (by default) for the interface. For more information on HSRP, refer to the following URL:

http://www.cisco.com/en/US/docs/ios/ipapp/configuration/guide/ipapp_hsrp_ps6350_TSD_Products_Configuration_Guide_Chapter.html

To enable the sending of ICMP Redirect messages if the Cisco IOS software is forced to resend a packet through the same interface on which it was received, enter the following command in interface configuration mode:

Command	Purpose
Switch (config-if)# [no] ip redirects	Enables ICMP Redirect messages. Use the no keyword to disable the ICMP Redirect messages and reduce CPU usage.

Enabling ICMP Mask Reply Messages

Occasionally, network devices must know the subnet mask for a particular subnetwork in the internetwork. To obtain this information, devices can send ICMP Mask Request messages. These messages are responded to by ICMP Mask Reply messages from devices that have the requested information. The Cisco IOS software can respond to ICMP Mask Request messages if the ICMP Mask Reply function is enabled.

To have the Cisco IOS software respond to ICMP mask requests by sending ICMP Mask Reply messages, enter the following command:

Command	Purpose
Switch (config-if)# [no] ip mask-reply	Enables response to ICMP destination mask requests. Use the no keyword to disable this functionality.



Configuring Supervisor Engine Redundancy Using RPR and SSO on Supervisor Engine 6-E and Supervisor Engine 6L-E

Catalyst 4500 series switches allow a redundant supervisor engine to take over if the active supervisor engine fails. In software, supervisor engine redundancy is enabled by running the redundant supervisor engine in route processor redundancy (RPR) or stateful switchover (SSO) operating mode.



Note

The minimum ROMMON requirement for running SSO is Cisco IOS Release 12.1(20r)EW1 or Cisco IOS Release 12.2(20r)EW1.

This chapter describes how to configure supervisor engine redundancy on the Catalyst 4507R switch.



Note

For information on Cisco nonstop forwarding (NSF) with SSO, see [Chapter 12, “Configuring Cisco NSF with SSO Supervisor Engine Redundancy.”](#)

This chapter contains these major sections:

- [About Supervisor Engine Redundancy, page 11-2](#)
- [About Supervisor Engine Redundancy Synchronization, page 11-4](#)
- [Supervisor Engine Redundancy Guidelines and Restrictions, page 11-5](#)
- [Configuring Supervisor Engine Redundancy, page 11-7](#)
- [Performing a Manual Switchover, page 11-12](#)
- [Performing a Software Upgrade, page 11-13](#)
- [Manipulating Bootflash on the Redundant Supervisor Engine, page 11-14](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About Supervisor Engine Redundancy

These sections describe supervisor engine redundancy:

- [Overview, page 11-2](#)
- [RPR Operation, page 11-2](#)
- [SSO Operation, page 11-3](#)

Overview

With supervisor engine redundancy enabled, if the active supervisor engine fails or if a manual switchover is performed, the redundant supervisor engine becomes the active supervisor engine. The redundant supervisor engine has been automatically initialized with the startup configuration of the active supervisor engine, shortening the switchover time (30 seconds or longer in RPR mode, depending on the configuration; subsecond in SSO mode).

In addition to the reduced switchover time, supervisor engine redundancy supports the following:

- Online insertion and removal (OIR) of the redundant supervisor engine.
Supervisor engine redundancy allows OIR of the redundant supervisor engine for maintenance. When the redundant supervisor engine is inserted, the active supervisor engine detects its presence, and the redundant supervisor engine boots into a partially-initialized state in RPR mode and a fully-initialized state in SSO mode.
- Software upgrade. See the [“Performing a Software Upgrade” section on page 11-13](#).
To minimize down time during software changes on the supervisor engine, load the new image on the redundant supervisor engine, and conduct a switchover.

When power is first applied to a switch, the supervisor engine that boots first becomes the active supervisor engine and remains active until a switchover occurs.

A switchover occurs when one or more of the following events take place:

- The active supervisor engine fails (due to either hardware or software function) or is removed.
- A user forces a switchover.
- A user reloads the active supervisor engine.

RPR Operation

RPR is supported in Cisco IOS Release 12.2(12c)EW and later releases. When a redundant supervisor engine runs in RPR mode, it starts up in a partially-initialized state and is synchronized with the persistent configuration of the active supervisor engine.



Note

Persistent configuration includes the following components: startup-config, boot variables, config-register, and VLAN database.

The redundant supervisor engine pauses the startup sequence after basic system initialization, and in the event that the active supervisor engine fails, the redundant supervisor engine becomes the new active supervisor engine.

In a supervisor engine switchover, traffic is disrupted because in the RPR mode all of the physical ports restart since there is no state maintained between supervisor engines relating to module types and statuses. When the redundant supervisor engine completes its initialization, it reads hardware information directly from the module.

SSO Operation

SSO is supported in Cisco IOS Release 12.2(20)EWA and later releases. When a redundant supervisor engine runs in SSO mode, the redundant supervisor engine starts up in a fully-initialized state and synchronizes with the persistent configuration and the running configuration of the active supervisor engine. It subsequently maintains the state on the protocols listed below, and all changes in hardware and software states for features that support stateful switchover are kept in synchronization. Consequently, it offers zero interruption to Layer 2 sessions in a redundant supervisor engine configuration.

Because the redundant supervisor engine recognizes the hardware link status of every link, ports that were active before the switchover remain active, including the uplink ports. However, because uplink ports are physically on the supervisor engine, they will be disconnected if the supervisor engine is removed.

If the active supervisor engine fails, the redundant supervisor engine become active. This newly active supervisor engine uses existing Layer 2 switching information to continue forwarding traffic. Layer 3 forwarding is delayed until the routing tables have been repopulated in the newly active supervisor engine.

SSO supports stateful switchover of the following Layer 2 features. The state of these features is preserved between both the active and redundant supervisor engines:

- 802.3
- 802.3u
- 802.3x (Flow Control)
- 802.3ab (GE)
- 802.3z (Gigabit Ethernet including CWDM)
- 802.3ad (LACP)
- 802.1p (Layer 2 QoS)
- 802.1q
- 802.1X (Authentication)
- 802.1D (Spanning Tree Protocol)
- 802.3af (Inline power)
- PAgP
- VTP
- Dynamic ARP Inspection
- DHCP snooping
- IP source guard
- IGMP snooping (versions 1 and 2)
- DTP (802.1q and ISL)
- MST

- PVST+
- Rapid-PVST
- PortFast/UplinkFast/BackboneFast
- BPDU guard and filtering
- Voice VLAN
- Port security
- Unicast MAC filtering
- ACL (VACLs, PACLS, RACLs)
- QoS (DBL)
- Multicast storm control/broadcast storm control

SSO is compatible with the following list of features. However, the protocol database for these features is not synchronized between the redundant and active supervisor engines:

- 802.1Q tunneling with Layer 2 Protocol Tunneling (L2PT)
- Baby giants
- Jumbo frame support
- CDP
- Flood blocking
- UDLD
- SPAN/RSPAN
- NetFlow

The following features are learned on the redundant supervisor engine if the SSO feature is enabled:

- All Layer 3 protocols on Catalyst 4500 series switches (Switch Virtual Interfaces)

About Supervisor Engine Redundancy Synchronization

During normal operation, the persistent configuration (RPR and SSO) and the running configuration (SSO only) are synchronized by default between the two supervisor engines. In a switchover, the new active supervisor engine uses the current configuration.



Note You cannot enter CLI commands on the redundant supervisor engine console.

These sections describe supervisor engine redundancy synchronization:

- [RPR Supervisor Engine Configuration Synchronization, page 11-4](#)
- [SSO Supervisor Engine Configuration Synchronization, page 11-5](#)

RPR Supervisor Engine Configuration Synchronization

Because the redundant supervisor engine is only partially initialized in RPR mode, it interacts with the active supervisor engine only to receive configuration changes at startup and upon saving the configuration changes.

When a redundant supervisor engine is running in RPR mode, the following events trigger synchronization of the configuration information:

- When the redundant supervisor engine boots, the **auto-sync** command synchronizes the persistent configuration. This command is enabled by default. For details, refer to [“Synchronizing the Supervisor Engine Configurations” section on page 11-11](#).
- When the active supervisor engine detects the redundant supervisor engine, the configuration information is synchronized from the active supervisor engine to the redundant supervisor engine. This synchronization overwrites any existing startup configuration file on the redundant supervisor engine.
- When you make changes to the configuration, you must use the **write** command to save and synchronize the startup configuration of the redundant supervisor engine.

SSO Supervisor Engine Configuration Synchronization

When a redundant supervisor engine runs in SSO mode, the following events trigger synchronization of the configuration information:

- When the active supervisor detects the redundant supervisor engine, synchronization of the persistent and running configuration takes place, allowing the redundant supervisor engine to arrive at a fully-initiated state.
- When real-time changes occur, the active supervisor engine synchronizes the running-config and (or) the persistent configuration (if necessary) with the redundant supervisor engine.
- When you change the configuration, you must use the **write** command to allow the active supervisor engine to save and synchronize the startup configuration of the redundant supervisor engine.

Supervisor Engine Redundancy Guidelines and Restrictions

The following guidelines and restrictions apply to supervisor engine redundancy:

- If SSO mode cannot be established between the active and standby supervisor engines because of an incompatibility in the configuration file, a mismatched command list (MCL) is generated at the active supervisor engine and a reload into RPR mode is forced for the standby supervisor engine. Subsequent attempts to establish SSO, after removing the offending configuration and rebooting the standby supervisor engine with the exact same image, might cause the C4K_REDUNDANCY-2-IOS_VERSION_CHECK_FAIL and ISSU-3-PEER_IMAGE_INCOMPATIBLE messages to appear because the peer image is listed as incompatible. If the configuration problem can be corrected, you can clear the peer image from the incompatible list with the **redundancy config-sync ignore mismatched-commands EXEC** command while the peer is in a standby cold (RPR) state. This action allows the standby supervisor engine to boot in standby hot (SSO) state when it reloads.

Here are the steps:

-
- | | |
|---------------|--|
| Step 1 | Clear the offending configuration (that caused an MCL) while the standby supervisor engine is in standby cold (RPR) state. |
| Step 2 | Enter the redundancy config-sync ignore mismatched-commands EXEC command at the active standby supervisor engine. |
| Step 3 | Perform write memory . |

Step 4 Reload the standby supervisor engine with the **redundancy reload peer** command.

- If you configure Supervisor Engine V-10GE to use both Gigabit Ethernet and 10-Gigabit Ethernet uplinks without WS-X4302-GB in slot 10, module 10 is disabled and you cannot rollback the configuration to use gigabit ports.

Enter the following commands to recover:

```
config t
hw-module uplink select tengigabitethernet // This sets the switch back to default
mode
!
ctrl z
wr me
redundancy reload shelf // The switch will reload with all 10 modules working ok

//The switch reloads

config t
hw-module uplink select gigabitethernet // This sets the switch to the desired link
!
ctrl z
wr me
redundancy reload shelf // The switch reloads with module 10 active with the gigabit
ethernet port(s) ON and the ten gigabit ethernet port(s) Off
```

- RPR requires Cisco IOS Release 12.1(12c)EW, Release 12.1(19)E or later releases. SSO requires Cisco IOS Release 12.2(20)EWA or later releases.
- The Catalyst 4507R switch and the only Catalyst 4500 series switches that support supervisor engine redundancy.
- In Cisco IOS Release 12.2(25)SG and later releases on a Catalyst 4507R series switch, 10-Gigabit Ethernet and Gigabit Ethernet uplinks are concurrently usable.
- Redundancy requires both supervisor engines in the chassis to have the same components (model, memory, NFL daughter card), and to use the same Cisco IOS software image.
-
- The active and redundant supervisor engines in the chassis must be in slots 1 and 2.
- Each supervisor engine in the chassis must have its own flash device and console port connections to operate the switch on its own.
- Each supervisor engine must have a unique console connection. Do not connect a Y cable to the console ports.
- Supervisor engine redundancy does not provide supervisor engine load balancing.
- The Cisco Express Forwarding (CEF) table is cleared on a switchover. As a result, routed traffic is interrupted until route tables reconverge. This reconvergence time is minimal because the SSO feature reduces the supervisor engine redundancy switchover time from 30+ seconds to subsecond, so Layer 3 also has a faster failover time if the switch is configured for SSO.
- Static IP routes are maintained across a switchover because they are configured from entries in the configuration file.
- Information about Layer 3 dynamic states that is maintained on the active supervisor engine is not synchronized to the redundant supervisor engine and is lost on switchover.

- Starting with Cisco IOS Release 12.2, if an unsupported condition is detected (such as when the active supervisor engine is running Cisco IOS Release 12.2(20)EW and the redundant supervisor engine is running Cisco IOS Release 12.1(20)EW), the redundant supervisor engine is reset multiple times and then placed in ROMMON mode. It is important to follow the procedures outlined in the [“Performing a Software Upgrade” section on page 11-13](#).
- If you are running (or upgrading to) Cisco IOS Release 12.2(20)EWA or Cisco IOS Release 12.2(25)EW and are using a single supervisor engine in a redundant chassis (Catalyst 4507R or Catalyst 4510R series switch), and you intend to use routed ports, do one of the following:
 - Use SVIs instead of routed ports.
 - Change the redundancy mode from SSO to RPR.
- Configuration changes made to the redundant supervisor engine through SNMP synchronization and SNMP set operations in SSO mode are not synchronized to the redundant supervisor engine. Even though you can still perform SNMP set operations in SSO mode, you might experience unexpected behavior.

After you configure the switch through SNMP in SSO mode, copy the running-config file to the startup-config file on the active supervisor engine to trigger synchronization of the startup-config file on the redundant supervisor engine. Reload the redundant supervisor engine so that the new configuration is applied on the redundant supervisor engine.

- You cannot perform configuration changes during the startup (bulk) synchronization. If you attempt to make configuration changes during this process, the following message is generated:

```
Config mode locked out till standby initializes
```

- If configuration changes occur at the same time as a supervisor engine switchover, these configuration changes are lost.
- If you remove a line card from a redundant switch and initiate an SSO switchover, and then reinsert the line card, all interfaces are shutdown. The rest of the original line card configuration is preserved.

This situation only occurs if a switch had reached SSO before you removed the line card.

Configuring Supervisor Engine Redundancy

These sections describe how to configure supervisor engine redundancy:

- [Configuring Redundancy, page 11-8](#)
- [Virtual Console for Standby Supervisor Engine, page 11-10](#)
- [Synchronizing the Supervisor Engine Configurations, page 11-11](#)

Configuring Redundancy

To configure redundancy, perform this task:

	Command	Purpose
Step 1	Switch(config)# redundancy	Enters redundancy configuration mode.
Step 2	Switch(config-red)# mode {sso rpr}	Configures SSO or RPR. When this command is entered, the redundant supervisor engine is reloaded and begins to work in SSO or RPR mode.
Step 3	Switch# show running-config	Verifies that SSO or RPR is enabled.
Step 4	Switch# show redundancy [clients counters history states]	Displays the redundancy information (counter, state, and so on) for the active and redundant supervisor engines.

When configuring redundancy, note the following:

- The **sso** keyword is supported in Cisco IOS Release 12.2(20)EWA and later releases.
- The **rpr** keyword is supported in Cisco IOS Release 12.1(12c)EW and later releases.

This example shows how to configure the system for SSO and display the redundancy facility information:

```
Switch> enable
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# redundancy
Switch(config-red)# mode sso
Switch(config-red)# end
Switch# show redundancy
Redundant System Information :
-----
        Available system uptime = 2 days, 2 hours, 39 minutes
Switchovers system experienced = 0
        Standby failures = 0
        Last switchover reason = none

        Hardware Mode = Duplex
Configured Redundancy Mode = Stateful Switchover
Operating Redundancy Mode = Stateful Switchover
        Maintenance Mode = Disabled
        Communications = Up

Current Processor Information :
-----
        Active Location = slot 1
        Current Software state = ACTIVE
Uptime in current state = 2 days, 2 hours, 39 minutes
        Image Version = Cisco Internetwork Operating System Software
IOS (tm) Catalyst 4000 L3 Switch Software (cat4000-I5S-M), Version 12.2(20)EWA(3
.92), CISCO INTERNAL USE ONLY ENHANCED PRODUCTION VERSION
Copyright (c) 1986-2004 by cisco Systems, Inc.
Compiled Wed 14-Jul-04 04:42 by esi
        BOOT = bootflash:cat4000-i5s-mz.122_20_EWA_392,1
Configuration register = 0x2002
```

```

Peer Processor Information :
-----
        Standby Location = slot 2
        Current Software state = STANDBY HOT
        Uptime in current state = 2 days, 2 hours, 39 minutes
        Image Version = Cisco Internetwork Operating System Software
IOS (tm) Catalyst 4000 L3 Switch Software (cat4000-I5S-M), Version 12.2(20)EWA(3
.92), CISCO INTERNAL USE ONLY ENHANCED PRODUCTION VERSION
Copyright (c) 1986-2004 by cisco Systems, Inc.
Compiled Wed 14-Jul-04 0
        BOOT = bootflash:cat4000-i5s-mz.122_20_EWA_392,1
        Configuration register = 0x2002

Switch#

```

This example shows how to display redundancy facility state information:

```

Switch# show redundancy states
my state = 13 -ACTIVE
    peer state = 8 -STANDBY HOT
        Mode = Duplex
        Unit = Primary
        Unit ID = 2

Redundancy Mode (Operational) = Stateful Switchover
Redundancy Mode (Configured) = Stateful Switchover
Redundancy State = Stateful Switchover
Maintenance Mode = Disabled
    Manual Swact = enabled
    Communications = Up

client count = 21
    client_notification_TMR = 240000 milliseconds
        keep_alive TMR = 9000 milliseconds
        keep_alive count = 0
        keep_alive threshold = 18
        RF debug mask = 0x0

Switch#

```

This example shows how to change the system configuration from RPR to SSO mode:

```

Switch(config)# redundancy
Switch(config-red)# mode
Switch(config-red)# mode sso
Changing to sso mode will reset the standby. Do you want to continue?[confirm]
Switch(config-red)# end
Switch#
*Aug 1 13:11:16: %C4K_REDUNDANCY-3-COMMUNICATION: Communication with the peer Supervisor
has been lost
*Aug 1 13:11:16: %C4K_REDUNDANCY-3-SIMPLEX_MODE: The peer Supervisor has been lost

```

This example shows how to change the system configuration from SSO to RPR mode:

```

Switch(config)# redundancy
Switch(config-red)# mode rpr
Changing to rpr mode will reset the standby. Do you want to continue?[confirm]
Switch(config-red)# end
*Aug 1 13:11:16: %C4K_REDUNDANCY-3-COMMUNICATION: Communication with the peer Supervisor
has been lost
*Aug 1 13:11:16: %C4K_REDUNDANCY-3-SIMPLEX_MODE: The peer Supervisor has been lost

```

Virtual Console for Standby Supervisor Engine

Catalyst 4500 series switches can be configured with two supervisor engines to provide redundancy. When the switch is powered, one of the supervisor engines becomes active and remains active until a switchover occurs. The other supervisor engine remains in standby mode.

Each supervisor engine has its own console port. Access to the standby supervisor engine is possible only through the console port of the standby supervisor engine. You must connect to the standby console to access, monitor or debug the standby supervisor.

The virtual console for a standby supervisor Engine enables you to access the standby console from the active supervisor engine without requiring a physical connection to the standby console. It uses IPC over EOBC to communicate with the standby supervisor engine, which emulates the standby console on the active supervisor engine. Only one active standby console session is active at any time.

The virtual console for the standby supervisor engine allows users who are logged onto the active supervisor engine to remotely execute **show** commands on the standby supervisor engine and view the results on the active supervisor engine. Virtual console is available only from the active supervisor engine.

You can access the standby virtual console from the active supervisor engine with the **attach module**, **session module**, or **remote login** commands on the active supervisor engine. You must be in privilege EXEC mode (level 15) to run these commands to access the standby console.

Once you enter the standby virtual console, the terminal prompt automatically changes to *hostname-standby-console* where *hostname* is the configured name of the switch. The prompt is restored back to the original prompt when you exit the virtual console.

You exit the virtual console with the **exit** or **quit** commands. When the inactivity period of the terminal on the active supervisor engine where you logged in exceeds the configured idle time, you are automatically logged out of the terminal on the active supervisor engine. In this instance, the virtual console session is also terminated. Virtual console session is also automatically terminated when the standby is rebooted. After the standby boots up, you need to create another virtual console session.

To log in to the standby supervisor engine using a virtual console, enter the following command:

```
Switch# session module 2
Connecting to standby virtual console
Type "exit" or "quit" to end this session

Switch-standby-console# exit
Switch#
```

If the standby console is not enabled, the following message appears:

```
Switch-standby-console#
Standby console disabled.
Valid commands are: exit, logout
```



Note

The standby virtual console provides the standard features that are available from the supervisor console such as command history, command completion, command help and partial command keywords.

The following limitations apply to the standby virtual console:

- All commands on the virtual console run to completion. It does not provide the auto-more feature; it functions as if the **terminal length 0** command has been executed. It is also noninteractive. You cannot interrupt or abort an executing command by any key sequence on the active supervisor engine. If a command produces considerable output, the virtual console displays it on the supervisor engine screen.

- The virtual console is noninteractive. Because the virtual console does not detect the interactive nature of a command, any command that requires user interaction causes the virtual console to wait until the RPC timer aborts the command.

The virtual console timer is set to 60 seconds. The virtual console returns to its prompt after 60 seconds. During this time, you cannot abort the command from the key board. You must wait for the timer to expire before you continue.

- You cannot use virtual console to view debug and syslog messages that are being displayed on the standby supervisor engine. The virtual console only displays the output of commands that are executed from the virtual console. Other information that is displayed on the real standby console does not appear on the virtual console.

Synchronizing the Supervisor Engine Configurations

To manually synchronize the configurations used by the two supervisor engines, perform this task on the active supervisor engine:

	Command	Purpose
Step 1	Switch(config)# redundancy	Enters redundancy configuration mode.
Step 2	Switch(config-red)# main-cpu	Enters main-cpu configuration submode.
Step 3	Switch(config-r-mc)# auto-sync { startup-config config-register bootvar standard }	Synchronizes the configuration elements.
Step 4	Switch(config-r-mc)# end	Returns to privileged EXEC mode.
Step 5	Switch# copy running-config startup-config	Synchronizes the running configuration in dynamic random-access memory (DRAM) to the startup configuration file in NVRAM. Note This step is not required to synchronize the running configuration file in (DRAM).



Note

Configuration changes made to the active supervisor engine through SNMP are not synchronized to the redundant supervisor engine. For information on how to handle this situation, see the [“Supervisor Engine Redundancy Guidelines and Restrictions”](#) section on page 11-5.



Note

The **auto-sync** command controls the synchronization of the config-reg, bootvar, and startup/private configuration files only. The calendar and VLAN database files are always synchronized when they change. In SSO mode, the running-config is always synchronized.

This example shows how to reenable the default automatic synchronization feature using the **auto-sync standard** command to synchronize the startup-config and config-register configuration of the active supervisor engine with the redundant supervisor engine. Updates for the boot variables are automatic and cannot be disabled.

```
Switch(config)# redundancy
Switch(config-red)# main-cpu
Switch(config-r-mc)# auto-sync standard
Switch(config-r-mc)# end
Switch# copy running-config startup-config
```

**Note**

To manually synchronize individual elements of the standard auto-sync configuration, disable the default automatic synchronization feature.

**Note**

When you configure the auto-sync standard, the individual sync options such as no auto-sync startup-config are ignored.

This example shows how to disable default automatic synchronization and allow only automatic synchronization of the config-registers of the active supervisor engine to the redundant supervisor engine, while disallowing synchronization of the startup configuration:

```
Switch(config)# redundancy
Switch(config-red)# main-cpu
Switch(config-r-mc)# no auto-sync standard
Switch(config-r-mc)# auto-sync config-register
Switch(config-r-mc)# end
```

Performing a Manual Switchover

This section describes how to perform a manual switchover (from the active supervisor engine to the redundant supervisor engine) for test purposes. We recommend that you perform a manual switchover prior to deploying SSO in your production environment.

**Note**

This discussion assumes that SSO has been configured as the redundant mode.

To perform a manual switchover, perform this task on the active supervisor engine:

	Command	Purpose
Step 1	Switch# show redundancy	Verifies that the peer state is in the standby hot state. See the example of the show redundancy states command on page 6-10.
Step 2	Switch# redundancy force-switchover	Launches switchover from the active supervisor engine to the redundant supervisor engine. If the state of the redundant supervisor engine is not standby hot, this command does not execute.

Be aware of these usage guidelines:

- To force a switchover, the redundant supervisor engine must be in a standby hot state. You can verify the state with the **show redundancy** command. If the state is not standby hot, the **redundancy force-switchover** command does not execute.
- Use the **redundancy force-switchover** command, rather than the **reload** command, to initiate a switchover. The **redundancy force-switchover** command first verifies that the redundant supervisor engine is in the correct state. If you enter the **reload** command and the status is not standby hot, the **reload** command resets the current supervisor engine only.

After an initial switchover, there might be occasions when you want to make the supervisor engine in slot 1 of the chassis the active supervisor engine. If the image on supervisor engine 1 is the one you intend to run on both supervisor engines, it is not necessary to reboot the image on the supervisor engine in slot 1 to make it redundant. Instead, you can force another switchover. However, if you want a newer version of the image to run on both supervisor engines, follow the steps under “Performing a Software Upgrade” on page 13. Use the **show module** command to see which slot contains the active supervisor engine, and force another switchover if necessary.

Performing a Software Upgrade

The software upgrade procedure supported by supervisor engine redundancy allows you to reload the Cisco IOS software image on the redundant supervisor engine, and once complete, reload the active supervisor engine once.

The software upgrade procedure supported by supervisor engine redundancy allows you to reload the Cisco IOS software image on the redundant supervisor engine, and once complete, reloads the active supervisor engine once.

The following scenario is not supported: An active supervisor engine running Cisco IOS Release 12.1(x)E, and a standby supervisor engine running Cisco IOS Release 12.2(x)S. The standby supervisor engine resets repeatedly.

If you are trying to upgrade redundant supervisor engines from Cisco IOS Release 12.1(x)E to 12.2(x)S, this requires a full system reboot.

To perform a software upgrade, perform this task:

	Command	Purpose
Step 1	Switch# copy <i>source_device:source_filename</i> slot0:target_filename Or: Switch# copy <i>source_device:source_filename</i> bootflash:target_filename	Copies the new Cisco IOS software image to bootflash on the supervisor engine.
Step 2	Switch# copy <i>source_device:source_filename</i> slaveslot0:target_filename Or: Switch# copy <i>source_device:source_filename</i> slavebootflash:target_filename	Copies the new image to a slave device (such as slavebootflash and slaveslot0).
Step 3	Switch# config terminal Switch(config)# config-register 0x2 Switch(config)# boot system flash <i>device:file_name</i>	Configures the supervisor engines to boot the new image. If your system was configured to autoboot an earlier image, enter the following command string to boot the new image instead: no boot system flash <i>device:old_file_name</i>
Step 4	Switch(config)# redundancy	Enters redundancy configuration mode.
Step 5	Switch(config-red)# main-cpu	Enters main-cpu configuration submode.
Step 6	Switch(config-r-mc)# auto-syn standard	Synchronizes the configuration elements.
Step 7	Switch(config-r-mc)# end	Returns to privileged EXEC mode.

	Command	Purpose
Step 8	Switch# copy running-config start-config	Saves the configuration.
Step 9	Switch# redundancy reload peer	Reloads the redundant supervisor engine and brings it back online (using the new release of the Cisco IOS software). Note Before proceeding to Step 10, ensure that the switch is operating in RPR mode.
Step 10	Switch# redundancy force-switchover	Conducts a manual switchover to the redundant supervisor engine. The redundant supervisor engine becomes the new active supervisor engine using the new Cisco IOS software image. The old active supervisor engine reboots with the new image and becomes the redundant supervisor engine.

This example shows how to perform a software upgrade:

```
Switch# config terminal
Switch(config)# config-register 0x2
Switch(config)# boot system flash slot0:cat4000-i5s-mz.122-20.EWA
Switch(config)# redundancy
Switch(config-red)# main-cpu
Switch(config-r-mc)# auto-syn standard
Switch(config-r-mc)# end
Switch# copy running-config start-config
Switch# redundancy reload peer
Switch# redundancy force-switchover
Switch#
```

This example illustrates how to verify that the running configuration on the active supervisor engine has successfully synchronized with the redundant supervisor engine:

```
Switch# config terminal
Switch(config)# redundancy
Switch(config-red)# main-cpu
Switch(config-r-mc)# auto-sync standard
4d01h: %C4K_REDUNDANCY-5-CONFIGSYNC: The bootvar has been successfully synchronized to the
standby supervisor
4d01h: %C4K_REDUNDANCY-5-CONFIGSYNC: The config-reg has been successfully synchronized to
the standby supervisor
4d01h: %C4K_REDUNDANCY-5-CONFIGSYNC: The startup-config has been successfully synchronized
to the standby supervisor
4d01h: %C4K_REDUNDANCY-5-CONFIGSYNC: The private-config has been successfully synchronized
to the standby supervisor
```

The example above shows that the boot variable, the config-register, and the startup configuration from the active supervisor engine have successfully synchronized to the redundant supervisor engine.

Manipulating Bootflash on the Redundant Supervisor Engine



Note The console port on the redundant supervisor engine is not available.

To manipulate the redundant supervisor engine bootflash, perform one or more of the following commands:

Command	Purpose
Switch# dir slaveslot0: <i>target_filename</i> or Switch# dir slavebootflash: <i>target_filename</i>	Lists the contents of the slot0: device on the redundant supervisor engine. Lists the contents of the bootflash: device on the redundant supervisor engine.
Switch# delete slaveslot0: <i>target_filename</i> or Switch# delete slavebootflash: <i>target_filename</i>	Deletes specific files from the slot0: device on the redundant supervisor engine. Deletes specific files from the bootflash: device on the redundant supervisor engine.
Switch# squeeze slaveslot0: or Switch# squeeze slavebootflash:	Squeezes the slot0: device on the redundant supervisor engine. Squeezes the bootflash: device on the redundant supervisor engine.
Switch# format slaveslot0: or Switch# format slavebootflash:	Formats the slot0: device on the redundant supervisor engine. Formats the bootflash: device on the redundant supervisor engine.
Switch# copy source_device:source_filename slaveslot0:target_filename or Switch# copy source_device:source_filename slavebootflash:target_filename	Copies a file from the active supervisor engine to the slot0: device on the redundant supervisor engine. Copies a file to the bootflash: device on a redundant supervisor engine. Note Source could be the active supervisor engine or a TFTP server.



Configuring Supervisor Engine Redundancy Using RPR and SSO on Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, and 7-E

Catalyst 4500 series switches allow a standby supervisor engine to take over if the active supervisor engine fails. In software, supervisor engine redundancy is enabled by running the redundant supervisor engine in route processor redundancy (RPR) or stateful switchover (SSO) operating mode.



Note

The minimum ROMMON requirement for running SSO and RPR is 15.0(1r)SG1. VSS default mode is SSO and is not configurable. VSS does not support RPR mode.

This chapter describes how to configure supervisor engine redundancy on the Catalyst 4507R switch.



Note

For information on Cisco nonstop forwarding (NSF) with SSO, see [Chapter 13, “Configuring Cisco NSF with SSO Supervisor Engine Redundancy.”](#)

This chapter contains these major sections:

- [About Supervisor Engine Redundancy, page 12-2](#)
- [About Supervisor Engine Redundancy Synchronization, page 12-4](#)
- [Supervisor Engine Redundancy Guidelines and Restrictions, page 12-5](#)
- [Configuring Supervisor Engine Redundancy, page 12-7](#)
- [Performing a Manual Switchover, page 12-11](#)
- [Performing a Software Upgrade, page 12-12](#)
- [Manipulating Bootflash on the Standby Supervisor Engine, page 12-14](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About Supervisor Engine Redundancy

These sections describe supervisor engine redundancy:

- [Overview, page 12-2](#)
- [RPR Operation, page 12-2](#)
- [SSO Operation, page 12-3](#)

Overview

With supervisor engine redundancy enabled, if the active supervisor engine fails or if a manual switchover is performed, the standby supervisor engine becomes the “new” active supervisor engine. The standby supervisor engine has been automatically initialized with the startup configuration of the active supervisor engine, shortening the switchover time (30 seconds or longer in RPR mode, depending on the configuration; subsecond in SSO mode).

In addition to the reduced switchover time, supervisor engine redundancy supports the following:

- Online insertion and removal (OIR) of the supervisor engine.
Supervisor engine redundancy allows OIR of the redundant supervisor engine for maintenance. When the redundant supervisor engine is inserted, the active supervisor engine detects its presence, and the redundant supervisor engine boots into a partially-initialized state in RPR mode and a fully-initialized state in SSO mode.
- Software upgrade. (See the [“Performing a Software Upgrade” section on page 12-12.](#))
To minimize down time during software changes on the supervisor engine, load the new image on the standby supervisor engine, and conduct a switchover.

When power is first applied to a switch, the supervisor engine that boots first becomes the active supervisor engine and remains active until a switchover occurs.

A switchover will occur when one or more of the following events take place:

- The active supervisor engine fails (due to either hardware or software function) or is removed.
- A user forces a switchover.
- A user reloads the active supervisor engine.

[Table 12-1](#) provides information about chassis and supervisor engine support for redundancy.

Table 12-1 Chassis and Supervisor Support

Supported Switches	Supported Supervisor Engines
WS-C4507R-E, WS-C4510R-E, WS-C4507R+E, WS-C4510R+E	Supervisor Engine 9-E (WS-X45-SUP9-E), Supervisor Engine 8L-E (WS-X45-SUP8L-E), Supervisor Engine 8-E (WS-X45-SUP8-E), (WS-X45-SUP7L-E), (WS-X45-SUP7-E),

RPR Operation

RPR is supported in Cisco IOS-XE Release 3.1.0SG and later releases. When a standby supervisor engine runs in RPR mode, it starts up in a partially-initialized state and is synchronized with the persistent configuration of the active supervisor engine.

**Note**

Persistent configuration includes the following components: startup-config, boot variables, config-register, and VLAN database.

The standby supervisor engine pauses the startup sequence after basic system initialization, and in the event that the active supervisor engine fails, the standby supervisor engine becomes the new active supervisor engine.

In a supervisor engine switchover, traffic is disrupted because in the RPR mode all of the physical ports restart since there is no state maintained between supervisor engines relating to module types and statuses. Upon switchover, when the standby supervisor engine completes its initialization, it will read hardware information directly from the module and become the active supervisor engine.

SSO Operation

SSO is supported in Cisco IOS-XE Release 3.1.0SG and later releases. When a standby supervisor engine runs in SSO mode, the standby supervisor engine starts up in a fully-initialized state and synchronizes with the persistent configuration and the running configuration of the active supervisor engine. It subsequently maintains the state on the protocols listed below, and all changes in hardware and software states for features that support stateful switchover are kept in synchronization. Consequently, it offers zero interruption to Layer 2 sessions in a redundant supervisor engine configuration.

Because the standby supervisor engine recognizes the hardware link status of every link, ports that were active before the switchover will remain active, including the uplink ports. However, because uplink ports are physically on the supervisor engine, they will be disconnected if the supervisor engine is removed.

If the active supervisor engine fails, the standby supervisor engine become active. This newly active supervisor engine uses existing Layer 2 switching information to continue forwarding traffic. Layer 3 forwarding will be delayed until the routing tables have been repopulated in the newly active supervisor engine.

After a switchover, Access Point (AP) connected ports take some time to converge. For faster STP convergence of AP connected ports, enable the **spanning-tree portfast edge** interface configuration command on all interfaces that are connected to APs. This enables the workstation or server to allow those devices to connect to the network immediately, rather than waiting for spanning tree to converge.

SSO supports stateful switchover of the following Layer 2 features.

The state of these features is preserved between both the active and standby supervisor engines:

- 802.3
- 802.3u
- 802.3x (Flow Control)
- 802.3ab (GE)
- 802.3z (Gigabit Ethernet including CWDm)
- 802.3ad (LACP)
- 802.1p (Layer 2 QoS)
- 802.1q
- 802.1X (Authentication)

- 802.1D (Spanning Tree Protocol)
- 802.3af (Inline power)
- PAgP
- VTP
- Dynamic ARP Inspection
- DHCP snooping
- IP source guard
- IGMP snooping (versions 1 and 2)
- DTP (802.1q and ISL)
- MST
- PVST+
- Rapid-PVST
- PortFast/UplinkFast/BackboneFast
- BPDU guard and filtering
- Voice VLAN
- Port security
- Unicast MAC filtering
- ACL (VACLs, PACLS, RACLs)
- QoS (DBL)
- Multicast storm control/broadcast storm control

SSO is compatible with the following list of features. However, the protocol database for these features is not synchronized between the standby and active supervisor engines:

- 802.1Q tunneling with Layer 2 Protocol Tunneling (L2PT)
- Baby giants
- Jumbo frame support
- CDP
- Flood blocking
- UDLD
- SPAN/RSPAN
- NetFlow

The following features are learned on the standby supervisor engine if the SSO feature is enabled:

- All Layer 3 protocols on Catalyst 4500 series switches (Switch Virtual Interfaces)

About Supervisor Engine Redundancy Synchronization

During normal operation, the persistent configuration (RPR and SSO) and the running configuration (SSO only) are synchronized by default between the two supervisor engines. In a switchover, the new active supervisor engine uses the current configuration.

**Note**

You cannot enter CLI commands on the standby supervisor engine console.

These sections describe supervisor engine redundancy synchronization:

- [RPR Supervisor Engine Configuration Synchronization, page 12-5](#)
- [SSO Supervisor Engine Configuration Synchronization, page 12-5](#)

RPR Supervisor Engine Configuration Synchronization

Because the standby supervisor engine is only partially initialized in RPR mode, it interacts with the active supervisor engine only to receive configuration changes at startup and upon saving the configuration changes.

When a standby supervisor engine is running in RPR mode, the following events trigger synchronization of the configuration information:

- When the standby supervisor engine boots, the **auto-sync** command synchronizes the persistent configuration. This command is enabled by default. For details, refer to “[Synchronizing the Supervisor Engine Configurations](#)” section on page 12-10.
- When the active supervisor engine detects the standby supervisor engine, the configuration information is synchronized from the active supervisor engine to the standby supervisor engine. This synchronization overwrites any existing startup configuration file on the standby supervisor engine.
- When you make changes to the configuration, you must use the **write** command to save and synchronize the startup configuration to the standby supervisor engine.

SSO Supervisor Engine Configuration Synchronization

When a standby supervisor engine runs in SSO mode, the following events trigger synchronization of the configuration information:

- When the active supervisor detects the standby supervisor engine, synchronization of the persistent and running configuration takes place, allowing the standby supervisor engine to arrive at a fully-initiated state.
- When real-time changes occur, the active supervisor engine synchronizes the running-config and (or) the persistent configuration (if necessary) with the standby supervisor engine.
- When you change the configuration, you must use the **write** command to allow the active supervisor engine to save and synchronize the startup configuration to the standby supervisor engine.

Supervisor Engine Redundancy Guidelines and Restrictions

The following guidelines and restrictions apply to supervisor engine redundancy:

- If SSO mode cannot be established between the active and standby supervisor engines because of an incompatibility in the configuration file, a mismatched command list (MCL) is generated at the active supervisor engine and a reload into RPR mode is forced for the standby supervisor engine. Subsequent attempts to establish SSO, after removing the offending configuration and rebooting the standby supervisor engine with the exact same image, might cause the C4K_REDUNDANCY-2-IOS_VERSION_CHECK_FAIL and

ISSU-3-PEER_IMAGE_INCOMPATIBLE messages to appear because the peer image is listed as incompatible. If the configuration problem can be corrected, you can clear the peer image from the incompatible list with the **redundancy config-sync ignore mismatched-commands EXEC** command while the peer is in a standby cold (RPR) state. This action allows the standby supervisor engine to boot in standby hot (SSO) state when it reloads.

Here are the steps:

-
- | | |
|---------------|--|
| Step 1 | Clear the offending configuration (that caused an MCL) while the standby supervisor engine is in standby cold (RPR) state. |
| Step 2 | Enter the redundancy config-sync ignore mismatched-commands EXEC command at the active standby supervisor engine. |
| Step 3 | Perform write memory . |
| Step 4 | Reload the standby supervisor engine with the redundancy reload peer command. |
-

- RPR and SSO requires Cisco IOS-XE Release 3.1.0SG and later releases.
- WS-C4507R-E, WS-C4510R-E, WS-C4507R+E, and WS-C4510R+E the only Catalyst 4500 series switches that support Supervisor Engine 7-E, Supervisor Engine 7L-E, and Supervisor Engine 8-E redundancy.
- In RPR or SSO mode,
 - with Supervisor Engine 7-E and 7L-E, only the first two uplinks on each supervisor engine are available. The second two uplinks are unavailable.
 - with Supervisor Engine 8-E, only the first four uplinks on each supervisor engine are available provided a 47xx line card is inserted on slot 10. The second set of four uplinks are unavailable. Only the first two uplinks on each supervisor are available unless the requirement is met (i.e., 47xx linecard in 4510 chassis).
- SSO requires both supervisor engines in the chassis to have the same components (model and memory), and to use the same Cisco IOS XE software image.
- The active and standby supervisor engines in the chassis must be in slots 3 and 4 for 7-slot chassis and slot 5 and 6 for 10-slot chassis.
- Each supervisor engine in the chassis must have its own flash device and console port connections to operate the switch on its own.
- Each supervisor engine must have a unique console connection. Do not connect a Y cable to the console ports.
- Supervisor engine redundancy does not provide supervisor engine load balancing.
- The Cisco Express Forwarding (CEF) table is cleared on a switchover. As a result, routed traffic is interrupted until route tables reconverge. This reconvergence time is minimal because the SSO feature reduces the supervisor engine redundancy switchover time from 30+ seconds to subsecond, so Layer 3 also has a faster failover time if the switch is configured for SSO.
- Static IP routes are maintained across a switchover because they are configured from entries in the configuration file.
- Information about Layer 3 dynamic states that is maintained on the active supervisor engine is not synchronized to the standby supervisor engine and is lost on switchover.

- If configuration changes on a redundant switch are made through SNMP set operations, the changes are not synchronized to the standby supervisor engine even in SSO mode. You might experience unexpected behavior.
- After you configure the switch through SNMP in SSO mode, copy the running-config file to the startup-config file on the active supervisor engine to trigger synchronization of the startup-config file to the standby supervisor engine. Then, reload the standby supervisor engine so that the new configuration is applied on the standby supervisor engine.
- You cannot perform configuration changes during the startup (bulk) synchronization. If you attempt to make configuration changes during this process, the following message is generated:

```
Config mode locked out till standby initializes
```

- If configuration changes occur at the same time as a supervisor engine switchover, these configuration changes are lost.
- If you remove a line card from a redundant switch and initiate an SSO switchover, then reinsert the line card, and all interfaces are shutdown. The rest of the original line card configuration is preserved.

This situation only occurs if a switch had reached SSO before you removed the line card.

Configuring Supervisor Engine Redundancy

These sections describe how to configure supervisor engine redundancy:

- [Configuring Redundancy, page 12-7](#)
- [Virtual Console for Standby Supervisor Engine, page 12-9](#)
- [Synchronizing the Supervisor Engine Configurations, page 12-10](#)

Configuring Redundancy



Note

IOS XE software can be booted at three different levels (Enterprise Services, IP Base, and LAN Base), based on the licenses available on the supervisor engine.

To configure redundancy, perform this task:

	Command	Purpose
Step 1	Switch(config)# redundancy	Enters redundancy configuration mode.
Step 2	Switch(config-red)# mode {sso rpr}	Configures SSO or RPR. When this command is entered, the standby supervisor engine is reloaded and begins to work in SSO or RPR mode.
Step 3	Switch# show running-config	Verifies that SSO or RPR is configured.
Step 4	Switch# show redundancy [clients counters history states]	Displays the redundancy information (counter, state, and so on) for the active and standby supervisor engines.

This example shows how to configure the system for SSO and display the redundancy facility information:

```

Switch> enable
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# redundancy
Switch(config-red)# mode sso
Switch(config-red)# end

Switch# show redundancy
Redundant System Information :

-----
        Available system uptime = 10 minutes Switchovers system experienced = 0
        Standby failures = 1
        Last switchover reason = none

        Hardware Mode = Duplex
        Configured Redundancy Mode = Stateful Switchover
        Operating Redundancy Mode = Stateful Switchover
        Maintenance Mode = Disabled
        Communications = Up

Current Processor Information :
-----
        Active Location = slot 3
        Current Software state = ACTIVE
        Uptime in current state = 9 minutes
        Image Version = Cisco IOS Software, Catalyst 4500 L3 Switch Software
        (cat4500e-UNIVERSALK9-M), Version 15.0(100)XO(1.42), INTERIM SOFTWARE Copyright (c)
        1986-2010 by Cisco Systems, Inc.
        Compiled Sun 01-Aug-10 04:12 by gsbuprod
        Configuration register = 0x920

Peer Processor Information :
-----
        Standby Location = slot 4
        Current Software state = STANDBY HOT
        Uptime in current state = 0 minute
        Image Version = Cisco IOS Software, Catalyst 4500 L3 Switch Software
        (cat4500e-UNIVERSALK9-M), Version 15.0(100)XO(1.42), INTERIM SOFTWARE Copyright (c)
        1986-2010 by Cisco Systems, Inc.
        Compiled Sun 01-Aug-10 04:12 by gsbuprod
        Configuration register = 0x920

```

This example shows how to display redundancy facility state information:

```

Switch# show redundancy states
        my state = 13 -ACTIVE
        peer state = 8 -STANDBY HOT
        Mode = Duplex
        Unit = Primary
        Unit ID = 3

Redundancy Mode (Operational) = Stateful Switchover
Redundancy Mode (Configured) = Stateful Switchover
Redundancy State = Stateful Switchover
Manual Swact = enabled

Communications = Up

        client count = 64
        client_notification_TMR = 240000 milliseconds
        keep_alive TMR = 9000 milliseconds
        keep_alive count = 1
        keep_alive threshold = 18

```

```
RF debug mask = 0
```

This example shows how to change the system configuration from RPR to SSO mode:

```
Switch(config)# redundancy
Switch(config-red)# mode
Switch(config-red)# mode sso
Changing to sso mode will reset the standby. Do you want to continue?[confirm]
Switch(config-red)# end
Switch#
*Aug 1 13:11:16: %C4K_REDUNDANCY-3-COMMUNICATION: Communication with the peer Supervisor
has been lost
*Aug 1 13:11:16: %C4K_REDUNDANCY-3-SIMPLEX_MODE: The peer Supervisor has been lost
```

This example shows how to change the system configuration from SSO to RPR mode:

```
Switch(config)# redundancy
Switch(config-red)# mode rpr
Changing to rpr mode will reset the standby. Do you want to continue?[confirm]
Switch(config-red)# end
*Aug 1 13:11:16: %C4K_REDUNDANCY-3-COMMUNICATION: Communication with the peer Supervisor
has been lost
*Aug 1 13:11:16: %C4K_REDUNDANCY-3-SIMPLEX_MODE: The peer Supervisor has been lost
```

Virtual Console for Standby Supervisor Engine

Catalyst 4500 series switches can be configured with 2 supervisor engines to provide redundancy. When the switch is powered, one of the supervisor engines becomes active and remains active until a switchover occurs. The other supervisor engine remains in standby mode.

Each supervisor engine has its own console port. Access to the standby supervisor engine is possible only through the console port of the standby supervisor engine. Therefore, you must connect to the standby console to access, monitor or debug the standby supervisor.

Virtual Console for Standby Supervisor Engine enables you to access the standby console from the active supervisor engine without requiring a physical connection to the standby console. It uses IPC over EOBC to communicate with the standby supervisor engine and thus emulate the standby console on the active supervisor engine. Only one standby console session is active at any time.

The Virtual Console for Standby Supervisor Engine allows users who are logged onto the active supervisor engine to remotely execute **show** commands on the standby supervisor engine and view the results on the active supervisor engine. Virtual Console is available only from the active supervisor engine.

You can access the standby virtual console from the active supervisor engine with the **attach module**, **session module**, or **remote login** commands on the active supervisor engine. You must be in privilege EXEC mode (level 15) to run these commands to access the standby console.

Once you enter the standby virtual console, the terminal prompt automatically changes to *hostname-standby-console* where *hostname* is the configured name of the switch. The prompt is restored to the original setting when you exit the virtual console.

You exit the virtual console with the **exit** or **quit** commands. When the inactivity period of the terminal on the active supervisor engine where you logged in exceeds the configured idle time, you are automatically logged out of the terminal on the active supervisor engine. In this instance, the virtual console session is also terminated. Virtual console session is also automatically terminated when the standby is rebooted. After the standby boots up, you need to create another virtual console session.

To log in to the standby supervisor engine using a virtual console, do the following:

```
Switch# session module 4
```

```
Connecting to standby virtual console
Type "exit" or "quit" to end this session
```

```
Switch-standby-console# exit
Switch#
```

If the standby console is not enabled, the following message appears:

```
Switch-standby-console#
Standby console disabled.
Valid commands are: exit, logout
```



Note

The standby virtual console provides the standard features that are available from the supervisor console such as command history, command completion, command help and partial command keywords.

The following limitations apply to the standby virtual console:

- All commands on the virtual console run to completion. It does not provide the auto-more feature; it behaves as if the **terminal length 0** command has been executed. It is also non-interactive. Therefore, a running command cannot be interrupted or aborted by any key sequence on the active supervisor engine. If a command produces considerable output, the virtual console displays it on the supervisor engine screen.
- The virtual console is non-interactive. Because the virtual console does not detect the interactive nature of a command, any command that requires user interaction causes the virtual console to wait until the RPC timer aborts the command.

The virtual console timer is set to 60 seconds. The virtual console returns to its prompt after 60 seconds. During this time, you cannot abort the command from the key board. You must wait for the timer to expire before you continue.

- You cannot use virtual console to view debug and syslog messages that are being displayed on the standby supervisor engine. The virtual console only displays the output of commands that are executed from the virtual console. Other information that is displayed on the real standby console does not appear on the virtual console.

Synchronizing the Supervisor Engine Configurations

To manually synchronize the configurations used by the two supervisor engines, perform this task on the active supervisor engine:

	Command	Purpose
Step 1	Switch(config)# redundancy	Enters redundancy configuration mode.
Step 2	Switch(config-red)# main-cpu	Enters main-cpu configuration submode.
Step 3	Switch(config-r-mc)# auto-sync {startup-config config-register bootvar standard}	Synchronizes the configuration elements.
Step 4	Switch(config-r-mc)# end	Returns to privileged EXEC mode.
Step 5	Switch# copy running-config startup-config	Synchronizes the running configuration in dynamic random-access memory (DRAM) to the startup configuration file in NVRAM.
		Note This step is not required to synchronize the running configuration file in (DRAM).

**Note**

Configuration changes made to the active supervisor engine through SNMP are not synchronized to the redundant supervisor engine. For information on how to handle this situation, see the [“Supervisor Engine Redundancy Guidelines and Restrictions”](#) section on page 12-5.

**Note**

The **auto-sync** command controls the synchronization of the config-reg, bootvar, and startup/private configuration files only. The calendar and VLAN database files are always synchronized when they change. In SSO mode, the running-config is always synchronized.

This example shows how to reenabling the default automatic synchronization feature using the **auto-sync standard** command to synchronize the startup-config and config-register configuration of the active supervisor engine with the standby supervisor engine. Updates for the boot variables are automatic and cannot be disabled.

```
Switch(config)# redundancy
Switch(config-red)# main-cpu
Switch(config-r-mc)# auto-sync standard
Switch(config-r-mc)# end
Switch# copy running-config startup-config
```

**Note**

To manually synchronize individual elements of the standard auto-sync configuration, disable the default automatic synchronization feature.

**Note**

When you configure the auto-sync standard, the individual sync options such as no auto-sync startup-config are ignored.

This example shows how to disable default automatic synchronization and allow only automatic synchronization of the config-registers of the active supervisor engine to the standby supervisor engine, while disallowing synchronization of the startup configuration:

```
Switch(config)# redundancy
Switch(config-red)# main-cpu
Switch(config-r-mc)# no auto-sync standard
Switch(config-r-mc)# auto-sync config-register
Switch(config-r-mc)# end
```

Performing a Manual Switchover

This section describes how to perform a manual switchover (from the active supervisor engine to the standby supervisor engine) for test purposes. We recommend that you perform a manual switchover prior to deploying SSO in your production environment.

**Note**

This discussion assumes that SSO has been configured as the redundant mode.

To perform a manual switchover, perform this task on the active supervisor engine:

	Command	Purpose
Step 1	Switch# show redundancy	Verifies that the peer state is in the standby hot state. See the example of the show redundancy states command on page 6-10.
Step 2	Switch# redundancy force-switchover	Launches switchover from the active supervisor engine to the standby supervisor engine.

Be aware of these usage guidelines:

- force a switchover, the redundant supervisor engine must be in a standby hot (SSO) or standby cold (RPR) state. You can verify the state with the **show redundancy** command. If the state is not standby hot or standby cold, the **redundancy force-switchover** command will not execute.
- Use the **redundancy force-switchover** command, rather than the **reload** command, to initiate a switchover. The **redundancy force-switchover** command will first check that the redundant supervisor engine is in the correct state. If you issue the **reload** command and the status is not standby hot or standby cold, the **reload** command will reset the current supervisor engine and the peer supervisor may not be able to take over because it was not in a terminal state (standby hot or cold).

After a normal switchover, you might want to make the supervisor engine in a lower slot number of the chassis the active supervisor engine. Use the **show module** command to see which slot contains the active supervisor engine, and force another switchover if necessary.

Performing a Software Upgrade

This is useful only if IOS -XE software is running in LAN Base mode. For Enterprise Services or IP Base mode, use ISSU to upgrade software for both RPR and SSO redundant mode.

The software upgrade procedure supported by supervisor engine redundancy allows you to reload the Cisco IOS software image on the redundant supervisor engine, and once complete, reload the active supervisor engine once.

To perform a software upgrade, perform this task:

	Command	Purpose
Step 1	Switch# copy <i>source_device:source_filename</i> <i>slot0:target_filename</i> Or: Switch# copy <i>source_device:source_filename</i> bootflash: <i>target_filename</i>	Copies the new Cisco IOS-XE software image to bootflash on the supervisor engine.
Step 2	Switch# copy <i>source_device:source_filename</i> slaveslot0: <i>target_filename</i> Or: Switch# copy <i>source_device:source_filename</i> slavebootflash: <i>target_filename</i>	Copies the new image to a device on the standby supervisor engine (such as slavebootflash and slaveslot0).

	Command	Purpose
Step 3	Switch# config terminal Switch(config)# config-register 0x2 Switch(config)# boot system flash <i>device:file_name</i>	Configures the supervisor engines to boot the new image. If your system was configured to autoboot an earlier image, issue the following command string to boot the new image instead: no boot system flash device:old_file_name
Step 4	Switch(config)# redundancy	Enters redundancy configuration mode.
Step 5	Switch(config-red)# main-cpu	Enters main-cpu configuration submode.
Step 6	Switch(config-r-mc)# auto-syn standard	Synchronizes the configuration elements.
Step 7	Switch(config-r-mc)# end	Returns to privileged EXEC mode.
Step 8	Switch# copy running-config start-config	Saves the configuration.
Step 9	Switch# redundancy reload peer	Reloads the standby supervisor engine and brings it back online (using the new release of the Cisco IOS-XE software).
Step 10	Switch# redundancy force-switchover	Conducts a manual switchover to the standby supervisor engine. The standby supervisor engine becomes the new active supervisor engine using the new Cisco IOS-XE software image. The old active supervisor engine reboots with the new image and becomes the standby supervisor engine.

This example shows how to perform a software upgrade:

```
Switch# config terminal
Switch(config)# config-register 0x2
Switch(config)# boot system flash
bootflash0:cat4500e-universalk9.SSA.03.01.00.150.1.XO.bin
Switch(config)# redundancy
Switch(config-red)# main-cpu
Switch(config-r-mc)# auto-syn standard
Switch(config-r-mc)# end
Switch# copy running-config start-config
Switch# redundancy reload peer
Switch# redundancy force-switchover
Switch#
```

This example illustrates how to verify that the running configuration on the active supervisor engine has successfully synchronized with the redundant supervisor engine:

```
Switch# config terminal
Switch(config)# redundancy
Switch(config-red)# main-cpu
Switch(config-r-mc)# auto-sync standard
4d01h: %C4K_REDUNDANCY-5-CONFIGSYNC: The bootvar has been successfully synchronized to the
standby supervisor
4d01h: %C4K_REDUNDANCY-5-CONFIGSYNC: The config-reg has been successfully synchronized to
the standby supervisor
4d01h: %C4K_REDUNDANCY-5-CONFIGSYNC: The startup-config has been successfully synchronized
to the standby supervisor
4d01h: %C4K_REDUNDANCY-5-CONFIGSYNC: The private-config has been successfully synchronized
to the standby supervisor
```

The example above shows that the boot variable, the config-register, and the startup configuration from the active supervisor engine have successfully synchronized to the redundant supervisor engine.

Manipulating Bootflash on the Standby Supervisor Engine



Note

The console port on the standby supervisor engine is not available.

To manipulate the standby supervisor engine bootflash, perform one or more of the following tasks:

Command	Purpose
Switch# dir slaveslot0: <i>target_filename</i>	Lists the contents of the slot0: device on the standby supervisor engine.
or Switch# dir slavebootflash: <i>target_filename</i>	Lists the contents of the bootflash: device on the standby supervisor engine.
Switch# delete slaveslot0: <i>target_filename</i>	Deletes specific files from the slot0: device on the standby supervisor engine.
or Switch# delete slavebootflash: <i>target_filename</i>	Deletes specific files from the bootflash: device on the standby supervisor engine.
Switch# squeeze slaveslot0: <i>target_filename</i>	Squeezes the slot0: device on the standby supervisor engine.
or Switch# squeeze slavebootflash: <i>target_filename</i>	Squeezes the bootflash: device on the standby supervisor engine.
Switch# format slaveslot0: <i>target_filename</i>	Formats the slot0: device on the standby supervisor engine.
or Switch# format slavebootflash: <i>target_filename</i>	Formats the bootflash: device on the standby supervisor engine.
Switch# copy source_device:source_filename slaveslot0: <i>target_filename</i>	Copies a file from the active supervisor engine to the slot0: device on the standby supervisor engine.
or Switch# copy source_device:source_filename slavebootflash: <i>target_filename</i>	Copies a file to the bootflash: device on a standby supervisor engine.
	Note Source could be the active supervisor engine or a TFTP server.



Configuring Cisco NSF with SSO Supervisor Engine Redundancy

This chapter describes how to configure supervisor engine redundancy using Cisco nonstop forwarding (NSF) with stateful switchover (SSO).



Note

Starting with Cisco IOS Release 12.2(52)SG, the Catalyst 4500 switch supports VRF lite NSF support with routing protocols OSPF/EIGRP/BGP.

This chapter consists of these sections:

- [About NSF with SSO Supervisor Engine Redundancy, page 13-1](#)
- [Configuring NSF with SSO Supervisor Engine Redundancy, page 13-9](#)
- [Cisco High Availability Features in Cisco IOS XE 3.1.0SG, page 13-17](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About NSF with SSO Supervisor Engine Redundancy

These sections describe supervisor engine redundancy using NSF with SSO:

- [About Cisco IOS NSF-Aware and NSF-Capable Support, page 13-2](#)
- [NSF with SSO Supervisor Engine Redundancy Overview, page 13-3](#)
- [SSO Operation, page 13-4](#)
- [NSF Operation, page 13-4](#)
- [Cisco Express Forwarding, page 13-5](#)
- [Routing Protocols, page 13-5](#)
- [NSF Guidelines and Restrictions, page 13-9](#)

About Cisco IOS NSF-Aware and NSF-Capable Support

Cisco IOS Nonstop Forwarding (NSF) has two primary components:

- NSF-awareness—If neighboring router devices detect that an NSF router can still forward packets when a supervisor engine switchover happens, this capability is referred to as NSF-awareness. Cisco IOS enhancements to the Layer 3 routing protocols (OSPF, BGP, EIGRP and IS-IS) are designed to prevent route-flapping so that the CEF routing table does not time out or the NSF router does not drop routes. An NSF-aware router helps to send routing protocol information to the neighboring NSF router.



Note

OSPF Version2 fast hellos generate false alarms. We recommend that you use Bidirectional Forwarding Detection (BFD) instead.

- NSF-capability—NSF works with SSO to minimize the amount of time that a Layer 3 network is unavailable following a supervisor engine switchover by continuing to forward IP packets. Reconvergence of Layer 3 routing protocols (BGP, EIGRP, OSPF v2, and IS-IS) is transparent to the user and happens automatically in the background. The routing protocols recover routing information from neighbor devices and rebuild the Cisco Express Forwarding (CEF) table.



Note

NSF does not support IPv6.

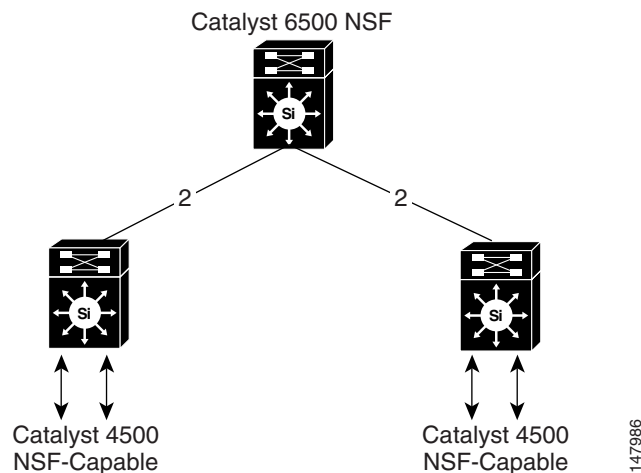


Note

NSF-capable devices include Catalyst 4500 series switches, Catalyst 6500 series switches, Cisco 7500 series routers, Cisco 10000 series routers, and Cisco 12000 series routers.

A typical topology for NSF and NSF-aware routers is given below.

Figure 13-1 Topology for NSF and NSF-Capable Switches



Starting with Cisco IOS Release 12.2(20)EWA, the Catalyst 4500 series switch supported NSF-awareness for the EIGRP, IS-IS, OSPF, and BGP protocols. Starting with Cisco IOS Release 12.2(31)SG, the Catalyst 4500 series switch supported NSF-awareness for the EIGRP-stub in IP Base image for all supervisor engines. NSF-awareness is turned on by default for EIGRP-stub, EIGRP, IS-IS, and OSPF protocols. You need to turn BGP on manually.

If the supervisor engine is configured for BGP (with the **graceful-restart** command), EIGRP, OSPF, or IS-IS routing protocols, routing updates are automatically sent during the supervisor engine switchover of a neighboring NSF-capable switch (typically a Catalyst 6500 series switch).

Starting with Cisco IOS Release 12.2(31)SG, the Catalyst 4500 series switch supports NSF-capability. [Table 13-1](#) lists the supervisor engines and the associated Catalyst 4500 series switches that are NSF-capable.

Table 13-1 NSF-Capable Supervisor Engines

NSF-Capable Supervisor Engines	Switch Support
Supervisor Engine 6-E (Supervisor Engine 6L-E (WS-X45-SUP7-E), (WS-X45-SUP7L-E), and Supervisor Engine 8-E (WS-X45-SUP8-E)	Catalyst 4500 E-series switch WS-C4507R-E, WS-C4510R-E, WS-C4507R+E, WS-C4510R+E
Supervisor Engine 9-E (WS-X45-SUP9-E)	WS-C4507R+E, WS-C4510R+E

NSF with SSO Supervisor Engine Redundancy Overview

Catalyst 4500 Series Switches support fault resistance by allowing a redundant supervisor engine to take over if the primary supervisor engine fails. NSF works with SSO to minimize the amount of time a network is unavailable to its users following a switchover.

NSF provides these benefits:

- Improved network availability

NSF continues forwarding network traffic and application state information so that user session information is maintained after a switchover.

- Overall network stability

Network stability may be improved with the reduction in the number of route flaps, which were created when routers in the network failed and lost their routing tables.

- Neighboring routers do not detect a link flap

Because the interfaces remain up during a switchover, neighboring routers do not detect a link flap (the link does not go down and come back up).

- Prevents routing flaps

Because SSO continues forwarding network traffic during a switchover, routing flaps are avoided.

- Maintains user sessions established prior to the switchover

Catalyst 4500 series switches also support route processor redundancy (RPR). For information about these redundancy modes, see [Chapter 10, “Configuring Supervisor Engine Redundancy Using RPR and SSO on Supervisor Engine 6-E and Supervisor Engine 6L-E.”](#)

SSO Operation

SSO establishes one of the supervisor engines as active while the other supervisor engine is designated as standby, and then SSO synchronizes information between them. A switchover from the active to the redundant supervisor engine occurs when the active supervisor engine fails, or is removed from the switch, or is manually shut down for maintenance.

In networking devices running SSO, both supervisor engines must be running the same Cisco IOS software version and ROMMON version so that the redundant supervisor engine is always ready to assume control following a fault on the active supervisor engine. SSO switchover also preserves FIB and adjacency entries and can forward Layer 3 traffic after a switchover. Configuration information and data structures are synchronized from the active to the redundant supervisor engine at startup and whenever changes to the active supervisor engine configuration occur. Following an initial synchronization between the two supervisor engines, SSO maintains state information between them, including forwarding information.

During switchover, system control and routing protocol execution is transferred from the active supervisor engine to the redundant supervisor engine.



Note

Use the **[no] service slave-log** configuration command to forward all error messages from the standby supervisor engine to the active engine. By default, this capability is enabled. For details, refer to the *Catalyst 4500 Series Switch Cisco IOS System Error Message Guide*, Release 12.2(37)SG.

NSF Operation

NSF always runs with SSO and provides redundancy for Layer 3 traffic. NSF is supported by the BGP, OSPF, IS-IS, and EIGRP routing protocols and is supported by Cisco Express Forwarding (CEF) for forwarding. The routing protocols have been enhanced with NSF-capability and awareness, which means that routers running these protocols can detect a switchover and take the necessary actions to continue forwarding network traffic and to recover route information from the peer devices. The IS-IS protocol can be configured to use state information that has been synchronized between the active and the redundant supervisor engine to recover route information following a switchover instead of information received from peer devices.

A networking device is NSF-aware if it is running NSF-compatible software. A device is NSF-capable if it has been configured to support NSF; it rebuilds routing information from NSF-aware or NSF-capable neighbors.

Each protocol depends on CEF to continue forwarding packets during switchover while the routing protocols rebuild the Routing Information Base (RIB) tables. After the routing protocols have converged, CEF updates the FIB table and removes stale route entries. CEF then updates the line cards with the new FIB information.

Cisco Express Forwarding

A key element of NSF is packet forwarding. In a Cisco networking device, packet forwarding is provided by Cisco Express Forwarding (CEF). CEF maintains the FIB and uses the FIB information that was current at the time of the switchover to continue forwarding packets during a switchover. This feature reduces traffic interruption during the switchover.

During normal NSF operation, CEF on the active supervisor engine synchronizes its current FIB and adjacency databases with the FIB and adjacency databases on the redundant supervisor engine. Upon switchover of the active supervisor engine, the redundant supervisor engine initially has FIB and adjacency databases that are mirror images of those that were current on the active supervisor engine. For platforms with forwarding engines, CEF keeps the forwarding engine on the redundant supervisor engine current with changes that are sent to it by CEF on the active supervisor engine. The forwarding engine can continue forwarding after a switchover as soon as the interfaces and a data path are available.

As the routing protocols start to repopulate the RIB on a prefix-by-prefix basis, the updates cause prefix-by-prefix updates to CEF, which it uses to update the FIB and adjacency databases. Existing and new entries receive the new version (“epoch”) number, indicating that they have been refreshed. The forwarding information is updated on the forwarding engine during convergence. The supervisor engine signals when the RIB has converged. The software removes all FIB and adjacency entries that have an epoch older than the current switchover epoch. The FIB now represents the newest routing protocol forwarding information.

Routing Protocols

**Note**

Use of the routing protocols require the Enterprise Services Cisco IOS Software image for the Catalyst 4500 series switch.

The routing protocols run only on the active supervisor engine, and they receive routing updates from their neighbor routers. Routing protocols do not run on the standby supervisor engine. Following a switchover, the routing protocols request that the NSF-aware neighbor devices send state information to help rebuild the routing tables. Alternately, the IS-IS protocol can be configured to synchronize state information from the active to the redundant supervisor engine to help rebuild the routing table on the NSF-capable device in environments where neighbor devices are not NSF-aware. NSF supports the BGP, OSPF, IS-IS, and EIGRP protocols.

**Note**

For NSF operation, the routing protocols depend on CEF to continue forwarding packets while the routing protocols rebuild the routing information.

BGP Operation

When an NSF-capable router begins a BGP session with a BGP peer, it sends an OPEN message to the peer. Included in the message is a statement that the NSF-capable device has “graceful” restart capability. Graceful restart is the mechanism by which BGP routing peers avoid a routing flap following a switchover. If the BGP peer has received this capability, it is aware that the device sending the message is NSF-capable. Both the NSF-capable router and its BGP peers need to exchange the graceful restart capability in their OPEN messages at the time of session establishment. If both peers do not exchange the graceful restart capability, the session will not be capable of a graceful restart.

If the BGP session is lost during the supervisor engine switchover, the NSF-aware BGP peer marks all the routes associated with the NSF-capable router as stale; however, it continues to use these routes to make forwarding decisions for a set period of time. This functionality prevents packets from being lost while the newly active supervisor engine is waiting for convergence of the routing information with the BGP peers.

After a supervisor engine switchover occurs, the NSF-capable router reestablishes the session with the BGP peer. In establishing the new session, it sends a new graceful restart message that identifies the NSF-capable router as having restarted.

At this point, the routing information is exchanged between the two BGP peers. After this exchange is complete, the NSF-capable device uses the routing information to update the RIB and the FIB with the new forwarding information. The NSF-aware device uses the network information to remove stale routes from its BGP table; the BGP protocol then is fully converged.

If a BGP peer does not support the graceful restart capability, it ignores the graceful restart capability in an OPEN message but establishes a BGP session with the NSF-capable device. This function allows interoperability with non-NSF-aware BGP peers (and without NSF functionality), but the BGP session with non-NSF-aware BGP peers is not capable of a graceful restart.

**Note**

BGP support in NSF requires that neighbor networking devices be NSF-aware; that is, the devices must have the graceful restart capability and advertise that capability in their OPEN message during session establishment. If an NSF-capable router discovers that a particular BGP neighbor does not have graceful restart capability, it does not establish an NSF-capable session with that neighbor. All other neighbors that have graceful restart capability continue to have NSF-capable sessions with this NSF-capable networking device.

OSPF Operation

**Note**

OSPF Version2 fast hellos generate false alarms. We recommend that you use Bidirectional Forwarding Detection (BFD) instead.

When an OSPF NSF-capable router performs a supervisor engine switchover, it must perform the following tasks in order to resynchronize its link state database with its OSPF neighbors:

- Relearn the available OSPF neighbors on the network without causing a reset of the neighbor relationship
- Reacquire the contents of the link state database for the network

As quickly as possible after a supervisor engine switchover, the NSF-capable router sends an OSPF NSF signal to neighboring NSF-aware devices. Neighbor networking devices recognize this signal as an indicator that the neighbor relationship with this router should not be reset. As the NSF-capable router receives signals from other routers on the network, it can begin to rebuild its neighbor list.

After neighbor relationships are reestablished, the NSF-capable router begins to resynchronize its database with all of its NSF-aware neighbors. At this point, the routing information is exchanged between the OSPF neighbors. Once this exchange is complete, the NSF-capable device uses the routing information to remove stale routes, update the RIB, and update the FIB with the new forwarding information. The OSPF protocols are then fully converged.

**Note**

OSPF support in NSF requires that all neighbor networking devices be NSF-aware. If an NSF-capable router discovers that it has non-NSF -aware neighbors on a particular network segment, it disables NSF capabilities for that segment. Other network segments composed entirely of NSF-capable or NSF-aware routers continue to provide NSF capabilities.

IS-IS Operation

When an IS-IS NSF-capable router performs a supervisor engine switchover, it must perform the following tasks in order to resynchronize its link state database with its IS-IS neighbors:

- Relearn the available IS-IS neighbors on the network without causing a reset of the neighbor relationship
- Reacquire the contents of the link state database for the network

The IS-IS NSF feature offers two options when you configure NSF:

- Internet Engineering Task Force (IETF) IS-IS
- Cisco IS-IS

If neighbor routers on a network segment are running a software version that supports the IETF Internet draft for router restartability, they assist an IETF NSF router that is restarting. With IETF, neighbor routers provide adjacency and link-state information to help rebuild the routing information following a switchover. A benefit of IETF IS-IS configuration is operation between peer devices based on a proposed standard.

**Note**

If you configure IETF on the networking device, but neighbor routers are not IETF-compatible, NSF aborts following a switchover.

If the neighbor routers on a network segment are not NSF-aware, you must use the Cisco configuration option. The Cisco IS-IS configuration transfers both protocol adjacency and link-state information from the active to the redundant supervisor engine. An advantage of Cisco configuration is that it does not rely on NSF-aware neighbors.

IETF IS-IS Configuration

As quickly as possible after a supervisor engine switchover, the NSF-capable router sends IS-IS NSF restart requests to neighboring NSF-aware devices using the IETF IS-IS configuration. Neighbor networking devices recognize this restart request as an indicator that the neighbor relationship with this router should not be reset, but that they should initiate database resynchronization with the restarting router. As the restarting router receives restart request responses from routers on the network, it can begin to rebuild its neighbor list.

After this exchange is complete, the NSF-capable device uses the link-state information to remove stale routes, update the RIB, and update the FIB with the new forwarding information; IS-IS is then fully converged.

The switchover from one supervisor engine to the other happens within seconds. IS-IS reestablishes its routing table and resynchronizes with the network within a few additional seconds. At this point, IS-IS waits for a specified interval before it attempts a second NSF restart. During this time, the new redundant supervisor engine boots up and synchronizes its configuration with the active supervisor engine. The

IS-IS NSF operation waits for a specified interval to ensure that connections are stable before attempting another restart of IS-IS NSF. This functionality prevents IS-IS from attempting back-to-back NSF restarts with stale information.

Cisco IS-IS Configuration

Using the Cisco configuration option, full adjacency and LSP information is saved, or *checkpointed*, to the redundant supervisor engine. Following a switchover, the newly active supervisor engine maintains its adjacencies using the check-pointed data, and can quickly rebuild its routing tables.



Note

Following a switchover, Cisco IS-IS NSF has complete neighbor adjacency and LSP information; however, it must wait for all interfaces to come on line that had adjacencies prior to the switchover. If an interface does not come on line within the allocated interface wait time, the routes learned from these neighbor devices are not considered in routing table recalculation. IS-IS NSF provides a command to extend the wait time for interfaces that, for whatever reason, do not come on line in a timely fashion.

The switchover from one supervisor engine to the other happens within seconds. IS-IS reestablishes its routing table and resynchronizes with the network within a few additional seconds. At this point, IS-IS waits for a specified interval before it attempts a second NSF restart. During this time, the new redundant supervisor engine boots up and synchronizes its configuration with the active supervisor engine. After this synchronization is completed, IS-IS adjacency and LSP data is check-pointed to the redundant supervisor engine; however, a new NSF restart is not attempted by IS-IS until the interval time expires. This functionality prevents IS-IS from attempting back-to-back NSF restarts.

EIGRP Operation

When an EIGRP NSF-capable router initially re-boots after an NSF restart, it has no neighbor and its topology table is empty. The router is notified by the redundant (now active) supervisor engine when it needs to bring up the interfaces, reacquire neighbors, and rebuild the topology and routing tables. The restarting router and its peers must accomplish these tasks without interrupting the data traffic directed toward the restarting router. EIGRP peer routers maintain the routes learned from the restarting router and continue forwarding traffic through the NSF restart process.

To prevent an adjacency reset by the neighbors, the restarting router uses a new Restart (RS) bit in the EIGRP packet header to indicate a restart. The RS bit is set in the hello packets and in the initial INIT update packets during the NSF restart period. The RS bit in the hello packets allows the neighbors to be quickly notified of the NSF restart. Without seeing the RS bit, the neighbor can only detect an adjacency reset by receiving an INIT update or by the expiration of the hello hold timer. Without the RS bit, a neighbor does not know if the adjacency reset should be handled using NSF or the normal startup method.

When the neighbor receives the restart indication, either by receiving the hello packet or the INIT packet, it recognizes the restarting peer in its peer list and maintains the adjacency with the restarting router. The neighbor then sends it topology table to the restarting router with the RS bit set in the first update packet indicating that it is NSF-aware and is helping out the restarting router. The neighbor does not set the RS bit in their hello packets, unless it is also a NSF restarting neighbor.



Note

A router may be NSF-aware but may not be helping the NSF restarting neighbor because booting from a cold start.

If at least one of the peer routers is NSF-aware, the restarting router then receives updates and rebuilds its database. The restarting router must then find out if it had converged so that it can notify the routing information base (RIB). Each NSF-aware router is required to send an end of table (EOT) marker in the last update packet to indicate the end of the table content. The restarting router knows it has converged when it receives the EOT marker. The restarting router can then begin sending updates.

An NSF-aware peer knows when the restarting router had converged when it receives an EOT indication from the restarting router. The peer then scans its topology table to search for the routes with the restarted neighbor as the source. The peer compares the route timestamp with the restart event timestamp to determine if the route is still available. The peer then goes active to find alternate paths for the routes that are no longer available through the restarted router.

When the restarting router has received all EOT indications from its neighbors or when the NSF converge timer expires, EIGRP notifies the RIB of convergence. EIGRP waits for the RIB convergence signal and then floods its topology table to all awaiting NSF-aware peers.

NSF Guidelines and Restrictions

NSF with SSO has these restrictions:

- With aggressive protocol timers (such as, when the default exceeds the timer value), upon switchover, the protocol software running on the new active supervisor engine might not initialize in time to send "hello" packets to its neighboring switches or routers. If the protocol takes longer time to initialize because of other CPU-demanding tasks, then the protocol encounters state transitions and causes a loss in traffic on the order of seconds. We recommend that you do not configure aggressive timers in conjunction with SSO/NSF.
- For NSF operation, you must have SSO configured on the device.
- NSF with SSO supports IP Version 4 traffic and protocols only; NSF with SSO does not support IPv6 traffic.
- The Virtual Redundancy Routing Protocols (VRRP) is not SSO-aware, meaning state information is not maintained between the active and standby supervisor engine during normal operation. VRRP and SSO can coexist but both features work independently. Traffic that relies on VRRP may switch to the VRRP standby in the event of a supervisor engine switchover.
- All neighboring devices participating in BGP NSF must be NSF-capable and configured for BGP graceful restart.
- OSPF NSF for virtual links is not supported.
- All OSPF networking devices on the same network segment must be NSF-aware (running an NSF software image).
- For IETF IS-IS, all neighboring devices must be running an NSF-aware software image.

Configuring NSF with SSO Supervisor Engine Redundancy

The following sections describe the configuration tasks for the NSF feature:

- [Configuring SSO, page 13-10](#)
- [Configuring CEF NSF, page 13-11](#)
- [Verifying CEF NSF, page 13-11](#)
- [Configuring BGP NSF, page 13-11](#)

- [Verifying BGP NSF, page 13-12](#)
- [Configuring OSPF NSF, page 13-13](#)
- [Verifying OSPF NSF, page 13-13](#)
- [Configuring IS-IS NSF, page 13-14](#)
- [Verifying IS-IS NSF, page 13-15](#)
- [Configuring EIGRP NSF, page 13-16](#)
- [Verifying EIGRP NSF, page 13-16](#)

Configuring SSO

You must configure SSO in order to use NSF with any supported protocol.

To configure SSO, perform this task:

	Command	Purpose
Step 1	Switch(config)# redundancy	Enters redundancy configuration mode.
Step 2	Switch(config-red)# mode sso	Configures SSO. When this command is entered, the redundant supervisor engine is reloaded and begins to work in SSO mode.
Step 3	Switch(config-red)# end	Returns to EXEC mode.
Step 4	Switch# show running-config	Verifies that SSO is enabled.
Step 5	Switch# show redundancy states	Displays the operating redundancy mode.



Note

The **sso** keyword is supported in Cisco IOS Release 12.2(20)EWA and later releases.

This example shows how to configure the system for SSO and display the redundancy state:

```
Switch> enable
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# redundancy
Switch(config-red)# mode sso
Switch(config-red)# end
Switch# show redundancy states
my state = 13 -ACTIVE
    peer state = 8  -STANDBY HOT
        Mode = Duplex
        Unit = Primary
        Unit ID = 5

Redundancy Mode (Operational) = sso
Redundancy Mode (Configured)  = sso
    Split Mode = Disabled
    Manual Swact = Enabled
    Communications = Up

    client count = 29
    client_notification_TMR = 30000 milliseconds
        keep_alive TMR = 9000 milliseconds
        keep_alive count = 1
```

```

        keep_alive threshold = 18
        RF debug mask = 0x0
Switch#

```

Configuring CEF NSF

The CEF NSF feature operates by default while the networking device is running in SSO mode. No configuration is necessary.

Verifying CEF NSF

To verify that CEF is NSF-capable, enter the **show cef state** command:

```

Switch# show cef state

CEF Status [RP]
CEF enabled/running
dCEF enabled/running
CEF switching enabled/running
CEF default capabilities:
Always FIB switching:      yes
Default CEF switching:     yes
Default dCEF switching:    yes
Update HWIDB counters:    no
Drop multicast packets:    no
.
.
.
CEF NSF capable:           yes
IPC delayed func on SSO:   no
RRP state:
I am standby RRP:          no
My logical slot:           0
RF PeerComm:               no

```

Configuring BGP NSF



Note

You must configure BGP graceful restart on all peer devices participating in BGP NSF.

To configure BGP for NSF, perform this task (repeat this procedure on each of the BGP NSF peer devices):

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.

	Command	Purpose
Step 2	Switch(config)# router bgp <i>as-number</i>	Enables a BGP routing process, which places the switch in switch configuration mode.
Step 3	Switch(config-router)# bgp graceful-restart	<p>Enables the BGP graceful restart capability, starting BGP NSF.</p> <p>If you enter this command after the BGP session has been established, you must restart the session for the capability to be exchanged with the BGP neighbor.</p> <p>Use this command on the restarting switch and all of its peers.</p>

Verifying BGP NSF

To verify BGP NSF, you must check that BGP graceful restart is configured on the SSO-enabled networking device and on the neighbor devices. To verify, follow these steps:

- Step 1** Verify that “bgp graceful-restart” appears in the BGP configuration of the SSO-enabled switch by entering the **show running-config** command:
- ```
Switch# show running-config

.
.
.
router bgp 120
.
.
.
bgp graceful-restart
neighbor 10.2.2.2 remote-as 300
.
.
.
```
- Step 2** Repeat Step 1 on each of the BGP neighbors.
- Step 3** On the SSO device and the neighbor device, verify that the graceful restart function is shown as both advertised and received, and confirm the address families that have the graceful restart capability. If no address families are listed, BGP NSF does not occur either:
- ```
Switch# show ip bgp neighbors x.x.x.x

BGP neighbor is 192.168.2.2, remote AS YY, external link
  BGP version 4, remote router ID 192.168.2.2
  BGP state = Established, up for 00:01:18
  Last read 00:00:17, hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh:advertised and received(new)
    Address family IPv4 Unicast:advertised and received
    Address family IPv4 Multicast:advertised and received
    Graceful Restart Capabilty:advertised and received
    Remote Restart timer is 120 seconds
    Address families preserved by peer:
      IPv4 Unicast, IPv4 Multicast
  Received 1539 messages, 0 notifications, 0 in queue
```



```
Sent 1544 messages, 0 notifications, 0 in queue
Default minimum time between advertisement runs is 30 seconds
```

Configuring OSPF NSF



Note

OSPF Version2 fast hellos generate false alarms. We recommend that you use Bidirectional Forwarding Detection (BFD) instead.

All peer devices participating in OSPF NSF must be made OSPF NSF-aware, which happens automatically when you install an NSF software image on the device.

To configure OSPF NSF, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# router ospf <i>processID</i>	Enables an OSPF routing process, which places the switch in router configuration mode.
Step 3	Switch(config-router)# nsf	Enables NSF operations for OSPF.

Verifying OSPF NSF

To verify OSPF NSF, you must check that the NSF function is configured on the SSO-enabled networking device. To verify OSPF NSF, follow these steps:

- Step 1** Verify that 'nsf' appears in the OSPF configuration of the SSO-enabled device by entering the **show running-config** command:

```
Switch# show running-config

route ospf 120
log-adjacency-changes
nsf
network 192.168.20.0 0.0.0.255 area 0
network 192.168.30.0 0.0.0.255 area 1
network 192.168.40.0 0.0.0.255 area 2
.
.
.
```

- Step 2** Enter the **show ip ospf** command to verify that NSF is enabled on the device:

```
Switch> show ip ospf

Routing Process "ospf 1" with ID 192.168.2.1 and Domain ID 0.0.0.1
Supports only single TOS(TOS0) routes
Supports opaque LSA
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
Number of external LSA 0. Checksum Sum 0x0
Number of opaque AS LSA 0. Checksum Sum 0x0
```

```

Number of DCbitless external and opaque AS LSA 0
Number of DoNotAge external and opaque AS LSA 0
Number of areas in this router is 1. 1 normal 0 stub 0 nssa
External flood list length 0
Non-Stop Forwarding enabled, last NSF restart 00:02:06 ago (took 44 secs)
Area BACKBONE(0)
Number of interfaces in this area is 1 (0 loopback)
Area has no authentication
SPF algorithm executed 3 times

```

Configuring IS-IS NSF

To configure IS-IS NSF, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# router isis [<i>tag</i>]	Enables an IS-IS routing process, which places the switch in router configuration mode.
Step 3	Switch(config-router)# nsf [cisco ietf]	Enables NSF operation for IS-IS. Enter the ietf keyword to enable IS-IS in a homogeneous network where adjacencies with networking devices supporting IETF draft-based restartability is guaranteed. Enter the cisco keyword to run IS-IS in heterogeneous networks that might not have adjacencies with NSF-aware networking devices.
Step 4	Switch(config-router)# nsf interval [<i>minutes</i>]	(Optional) Specifies the minimum time between NSF restart attempts. The default time between <i>consecutive</i> NSF restart attempts is 5 minutes.
Step 5	Switch(config-router)# nsf t3 { manual [<i>seconds</i>] adjacency }	(Optional) Specifies the time IS-IS waits for the IS-IS database to synchronize before generating overloaded link-state information for itself and flooding that information out to its neighbors. The t3 keyword applies only if you selected IETF operation. When you specify adjacency , the switch that is restarting obtains its wait time from neighboring devices.
Step 6	Switch(config-router)# nsf interface wait <i>seconds</i>	(Optional) Specifies how long an IS-IS NSF restart waits for all interfaces with IS-IS adjacencies to come up before completing the restart. The default is 10 seconds.

Verifying IS-IS NSF

To verify IS-IS NSF, you must check that the NSF function is configured on the SSO-enabled networking device. To verify IS-IS NSF, follow these steps:

- Step 1** Verify that “nsf” appears in the IS-IS configuration of the SSO-enabled device by entering the **show running-config** command. The display shows either the Cisco IS-IS or the IETF IS-IS configuration. The following display indicates that the device uses the Cisco implementation of IS-IS NSF:

```
Switch# show running-config
<...Output Truncated...>
router isis
nsf cisco
<...Output Truncated...>
```

- Step 2** If the NSF configuration is set to **cisco**, enter the **show isis nsf** command to verify that NSF is enabled on the device. Using the Cisco configuration, the display output differs on the active and redundant RPs. The following display shows sample output for the Cisco configuration on the active RP. In this example, note the presence of “NSF restart enabled”:

```
Switch# show isis nsf

NSF is ENABLED, mode 'cisco'

RP is ACTIVE, standby ready, bulk sync complete
NSF interval timer expired (NSF restart enabled)
Checkpointing enabled, no errors
Local state:ACTIVE, Peer state:STANDBY HOT, Mode:SSO
```

The following display shows sample output for the Cisco configuration on the standby RP. In this example, note the presence of “NSF restart enabled”:

```
Switch# show isis nsf

NSF enabled, mode 'cisco'
RP is STANDBY, chkpt msg receive count:ADJ 2, LSP 7
NSF interval timer notification received (NSF restart enabled)
Checkpointing enabled, no errors
Local state:STANDBY HOT, Peer state:ACTIVE, Mode:SSO
```

- Step 3** If the NSF configuration is set to **ietf**, enter the **show isis nsf** command to verify that NSF is enabled on the device. The following display shows sample output for the IETF IS-IS configuration on the networking device:

```
Switch# show isis nsf

NSF is ENABLED, mode IETF
NSF pdb state:Inactive
NSF L1 active interfaces:0
NSF L1 active LSPs:0
NSF interfaces awaiting L1 CSNP:0
Awaiting L1 LSPs:
NSF L2 active interfaces:0
NSF L2 active LSPs:0
NSF interfaces awaiting L2 CSNP:0
Awaiting L2 LSPs:
Interface:Serial3/0/2
  NSF L1 Restart state:Running
  NSF p2p Restart retransmissions:0
  Maximum L1 NSF Restart retransmissions:3
```

```
L1 NSF ACK requested:FALSE
L1 NSF CSNP requested:FALSE
NSF L2 Restart state:Running
NSF p2p Restart retransmissions:0
Maximum L2 NSF Restart retransmissions:3
L2 NSF ACK requested:FALSE
Interface:GigabitEthernet2/0/0
  NSF L1 Restart state:Running
  NSF L1 Restart retransmissions:0
  Maximum L1 NSF Restart retransmissions:3
  L1 NSF ACK requested:FALSE
  L1 NSF CSNP requested:FALSE
  NSF L2 Restart state:Running
  NSF L2 Restart retransmissions:0
  Maximum L2 NSF Restart retransmissions:3
  L2 NSF ACK requested:FALSE
  L2 NSF CSNP requested:FALSE
Interface:Loopback1
  NSF L1 Restart state:Running
  NSF L1 Restart retransmissions:0
  Maximum L1 NSF Restart retransmissions:3
  L1 NSF ACK requested:FALSE
  L1 NSF CSNP requested:FALSE
  NSF L2 Restart state:Running
  NSF L2 Restart retransmissions:0
  Maximum L2 NSF Restart retransmissions:3
  L2 NSF ACK requested:FALSE
  L2 NSF CSNP requested:FALSE
```

Configuring EIGRP NSF

To configure EIGRP NSF, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# router eigrp <i>as-number</i>	Enables an EIGRP routing process, which places the switch in router configuration mode.
Step 3	Switch(config-router)# nsf	Enables EIGRP NSF. Use this command on the “restarting” switch and all of its peers.

Verifying EIGRP NSF

To verify EIGRP NSF, you must check that the NSF function is configured on the SSO-enabled networking device. To verify EIGRP NSF, follow these steps:

- Step 1
- Verify that “nsf” appears in the EIGRP configuration of the SSO-enabled device by entering the **show running-config** command:

```
Switch# show running-config
.
```

```
.
router eigrp 100
  auto-summary
  nsf
..
.
```

Step 2 Enter the **show ip protocols** command to verify that NSF is enabled on the device:

```
Switch# show ip protocols
*** IP Routing is NSF aware ***
Routing Protocol is "eigrp 100"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 100
  EIGRP NSF-aware route hold timer is 240s
  EIGRP NSF enabled
    NSF signal timer is 20s
    NSF converge timer is 120s
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
  Routing Information Sources:
    Gateway         Distance      Last Update
  Distance: internal 90 external 170
```

Cisco High Availability Features in Cisco IOS XE 3.1.0SG

This section provides a list of High Availability software features that are supported in Cisco IOS XE 3.1.0SG. Links to the feature documentation are included.

Feature guides may contain information about more than one feature. To find information about a specific feature within a feature guide, see the Feature Information table at the end of the guide.

Feature guides document features that are supported on many different software releases and platforms. Your Cisco software release or platform may not support all the features documented in a feature guide. See the Feature Information table at the end of the feature guide for information about which features in that guide are supported in your software release. Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Enhanced High System Availability

<http://www.cisco.com/en/US/docs/ios-xml/ios/ha/configuration/xe-3s/ha-config-stateful-switchover.html>

NSF - Graceful Restart (GR) and Non Stop Routing (NSR) for IS-IS

<http://www.cisco.com/en/US/docs/ios-xml/ios/ha/configuration/xe-3s/ha-config-nonstop-forwarding.html>

NSF - OSPF

<http://www.cisco.com/en/US/docs/ios-xml/ios/ha/configuration/xe-3s/ha-config-nonstop-forwarding.html>

NSF/SSO (Nonstop Forwarding with Stateful Switchover)

<http://www.cisco.com/en/US/docs/ios-xml/ios/ha/configuration/xe-3s/ha-config-nonstop-forwarding.html>

SSO - HDLC

<http://www.cisco.com/en/US/docs/ios-xml/ios/ha/configuration/xe-3s/ha-config-stateful-switchover.html>

SSO - HSRP

<http://www.cisco.com/en/US/docs/ios-xml/ios/ha/configuration/xe-3s/ha-config-stateful-switchover.html>

SSO - Multilink PPP (MLP)

<http://www.cisco.com/en/US/docs/ios-xml/ios/ha/configuration/xe-3s/ha-config-stateful-switchover.html>

SSO - PPP

<http://www.cisco.com/en/US/docs/ios-xml/ios/ha/configuration/xe-3s/ha-config-stateful-switchover.html>



Environmental Monitoring and Power Management



Note

Before reading this chapter, read the “Preparing for Installation” section of the *Catalyst 4500 Series Installation Guide*. It is important to ensure that your installation site has enough power and cooling to accommodate the additional electrical load and heat introduced by Power over Ethernet (PoE).

This chapter describes power management and environmental monitoring features in the Catalyst 4500 series switches. It provides guidelines, procedures, and configuration examples.

This chapter consists of the following major sections:

- [About Environmental Monitoring, page 14-1](#)
- [Power Management, page 14-7](#)
- [IEEE 802.3az Energy Efficient Ethernet, page 14-23](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About Environmental Monitoring

This section contains the following subsections:

- [Using CLI Commands to Monitor your Environment, page 14-2](#)
- [Displaying Environment Conditions, page 14-2](#)
- [Displaying On Board Failure Logging \(OBFL\) information for 9000W AC, page 14-4](#)
- [Emergency Actions, page 14-5](#)
- [System Alarms, page 14-6](#)

Environmental monitoring of chassis components provides early warning indications of possible component failure. This warning helps you to ensure the safe and reliable operation of your system and avoid network interruptions.

This section describes how to monitor critical system components so that you can identify and rapidly correct hardware-related problems.

Using CLI Commands to Monitor your Environment

Use the **show environment** CLI command to monitor the system. This section gives a basic overview of the command and keywords you need.

Enter the **show environment [alarm | status | temperature]** command to display system status information. Keyword descriptions are listed in [Table 14-1](#).

Table 14-1 *show environment Keyword Descriptions*

Keyword	Purpose
alarm	Displays environmental alarms for the system.
status	Displays field-replaceable unit (FRU) operational status and power and power supply fan sensor information.
temperature	Displays temperature of the chassis.

Displaying Environment Conditions

Supervisor Engine 6-E, Supervisor Engine 6L-E, Supervisor Engine 7-E, and Supervisor Engine 7L-E and their associated line cards support multiple temperature sensors per card. The environment condition output includes the temperature reading from each sensor and the temperature thresholds for each sensor. These line cards support three thresholds: warning, critical, and shutdown.

The following example illustrates how to display the environment condition on a Supervisor Engine 6-E and Supervisor 6L-E. The thresholds appear within parentheses.

```
Switch# show environment
no temperature alarms
```

Module	Sensor	Temperature	Status
2	air inlet	23C (51C, 65C, 68C)	ok
2	air outlet	29C (69C, 83C, 86C)	ok
5	air inlet	38C (51C, 65C, 68C)	ok
5	air outlet	38C (69C, 83C, 86C)	ok
6	air inlet	34C (51C, 65C, 68C)	ok
6	air outlet	37C (69C, 83C, 86C)	ok

Power Supply	Model No	Type	Status	Fan Sensor	Inline Status
PS1	PWR-C45-2800AC	AC 2800W	good	good	good
PS2	none	--	--	--	--

```
Power supplies needed by system : 1
Power supplies currently available : 1
```

```
Chassis Type : WS-C4510R-E
```


Power consumed by backplane : 40 Watts

Switch Bandwidth Utilization : 0%

Supervisor Led Color : Green

Module 2 Status Led Color : Green

Module 5 Status Led Color : Green

Module 6 Status Led Color : Orange

Module 10 Status Led Color : Green

Fantray : Good

Power consumed by Fantray : 80 Watts

The following example illustrates how to display the environment condition on WS-C4506-E with a Supervisor Engine 7-E. The thresholds appear within parentheses.

Switch# **show environment**

Module	Sensor	Temperature	Status
1	Xgstub A	39C (48C,62C,65C)	ok
1	Xgstub B	32C (45C,60C,63C)	ok
1	XPP	47C (62C,75C,78C)	ok
1	VFE2	59C (74C,85C,88C)	ok
1	NFE	44C (63C,75C,78C)	ok
1	CPU	55C (57C,72C,75C)	ok
1	FPGA	37C (52C,66C,69C)	ok
4	Power macro	30C (56C,68C,71C)	ok
4	Air inlet	27C (46C,59C,62C)	ok
4	Xgstub	31C (66C,76C,79C)	ok
4	Air outlet	30C (60C,71C,74C)	ok

Power Supply	Model No	Type	Status	Fan Sensor	Inline Status
PS1	PWR-C45-1300ACV	AC 1300W	good	good	good
PS2	none	--	--	--	--

Power supplies needed by system : 1
Power supplies currently available : 1

Chassis Type : WS-C4506-E

Power consumed by backplane : 0 Watts

Switch Bandwidth Utilization : 0%

Supervisor Led Color : Green

Module 1 Status Led Color : Green

Module 4 Status Led Color : Green

Module 6 Status Led Color : Green

PoE Led Color : Green

PoE Led Color : Green

Fantray : Good

Fantray removal timeout : 30

Power consumed by Fantray : 120 Watts

The following example illustrates how to display the environment condition on WS-C4500X-32 with a Supervisor Engine 7-E. The thresholds appear within parentheses.

```
Switch> show environment
no temperature alarms
```

Module	Sensor	Temperature	Status
1	XPP	42C (80C,90C,100C)	ok
1	VFE	47C (80C,90C,100C)	ok
1	CPU	46C (80C,90C,100C)	ok
1	FPGA	50C (80C,90C,100C)	ok
1	Stub A	34C (80C,90C,100C)	ok
1	Stub B	35C (80C,90C,100C)	ok
2	Air inlet	31C (80C,90C,100C)	ok
2	Air outlet	36C (80C,90C,100C)	ok

Power Supply	Model No	Type	Status	Fan Sensor	Inline Status
PS1	none	--	--	--	--
PS2	PWR-C49X-750AC-R	AC 750W	good	good	n.a.

```
Power supplies needed by system      : 1
Power supplies currently available   : 1
```

```
Chassis Type : WS-C4500X-32
```

```
Power consumed by backplane : 0 Watts
```

```
Switch Bandwidth Utilization : 0%
```

```
Supervisor Led Color : Green
```

```
Module 1 Status Led Color : Green
Module 2 Status Led Color : Green
```

```
Beacon Led Status : off
```

```
Fan trays needed by system      : 4
Fan trays currently available   : 5
Chassis fan tray direction     : FrontToBack
```

```
Fantray 1 : dir : FrontToBack    status : Good
Fantray 2 : dir : FrontToBack    status : Good
Fantray 3 : dir : FrontToBack    status : Good
Fantray 4 : dir : FrontToBack    status : Good
Fantray 5 : dir : FrontToBack    status : Good
```

```
Fantray removal timeout : 30
```

```
Power consumed by Fantray : 30 Watts
```

```
Switch>
```

Displaying On Board Failure Logging (OBFL) information for 9000W AC

9000W AC power supplies support logging of voltage sag (input voltage drops below a certain input threshold) and voltage surge (input voltage spikes above a certain input threshold) events.

If a 9000W power supply is installed in the left bay, the **show logging onboard subslot 0 detail** command displays logging information for that power supply.

If a 9000W power supply is installed in the right bay, enter the **show logging onboard subslot 1 detail** command, as follows:

```
Switch# show logging onboard subslot 0 detail
PID: WS-C4506-E , VID: 4 , SN: FOX1347GR85

-----
ERROR MESSAGE SUMMARY INFORMATION
-----
Facility-Sev-Name      | Count | Persistence Flag
MM/DD/YYYY HH:MM:SS
-----
No historical data to display
-----

-----
ERROR MESSAGE CONTINUOUS INFORMATION
-----
MM/DD/YYYY HH:MM:SS Facility-Sev-Name
-----
08/20/2012 08:06:23 %CAT4K-2-POWER_SAG : 5 Power Sag event(s) detected: Slot 1,
Input 1
08/20/2012 08:06:23 %CAT4K-2-POWER_SURGE : 5 Power Surge event(s) detected: Slot 1, Input 1
09/09/2012 21:12:28 %CAT4K-2-POWER_SURGE : 5 Power Surge event(s) detected: Slot 1, Input 3
09/09/2012 21:22:30 %CAT4K-2-POWER_SURGE : 5 Power Surge event(s) detected: Slot 1, Input 3
09/09/2012 21:32:32 %CAT4K-2-POWER_SURGE : 5 Power Surge event(s) detected: Slot 1, Input 3
09/10/2012 00:05:49 %CAT4K-2-POWER_SAG : 1 Power Sag event(s) detected: Slot 1, Input 1
09/10/2012 00:05:49 %CAT4K-2-POWER_SURGE : 1 Power Surge event(s) detected: Slot 1, Input 1
-----
```

Emergency Actions

Catalyst 4500 chassis can power down a single card, providing a detailed response to over-temperature conditions on line cards. However, the Catalyst 4500 chassis *cannot* safely operate when the temperature of the supervisor itself exceeds the critical threshold. The supervisor engine turns off the chassis' power supplies to protect itself from overheating. When this happens, you can recover the switch only by cycling the power on and off switches on the power supplies or by cycling the AC or DC inputs to the power supplies.

Critical and shutdown temperature emergencies trigger the same action. [Table 14-2](#) lists temperature emergencies but does not distinguish between critical and shutdown emergencies.

Table 14-2 Emergency and Action

Case 1. Complete fan failure emergency.	Power down the chassis.
Case 2. Temperature emergency on a line card.	Power down the line card.
Case 3. Temperature emergency on the standby supervisor engine.	Power down the standby supervisor engine.

Table 14-2 **Emergency and Action**

Case 1. Complete fan failure emergency.	Power down the chassis.
Case 4. Temperature emergency on the active supervisor engine with the standby supervisor engine in the hot standby or cold standby redundancy state.	Reset the active supervisor engine.
Case 5. Temperature emergency on the active supervisor engine with no standby supervisor engine or with a standby supervisor engine that is not in hot standby or cold standby redundancy state.	Power down the chassis.

In Case 4, the standby supervisor engine takes over when the active engine resets itself. If the temperature emergency remains, the newly active supervisor engine resets the standby supervisor engine.

Case 5 applies to nonredundant chassis and to chassis with a standby supervisor engine that has been shutdown or which has not fully booted.

System Alarms

Any system has two types of alarms: major and minor. A major alarm indicates a critical problem that could lead to system shutdown. A minor alarm is informational—it alerts you to a problem that could become critical if corrective action is not taken.

[Table 14-3](#) lists the possible environment alarms.

Table 14-3 **Possible Environmental Alarms**

A temperature sensor over its warning threshold	minor
A temperature sensor over its critical threshold	major
A temperature sensor over its shutdown threshold	major
A partial fan failure	minor
A complete fan failure	major

Fan failure alarms are issued as soon as the fan failure condition is detected and are canceled when the fan failure condition clears. Temperature alarms are issued as soon as the temperature reaches the threshold temperature and are canceled when the temperature drops more than 5 degree C below the threshold. 5 degree C is a hysteresis value designed to prevent toggling alarms.

An LED on the supervisor engine indicates whether an alarm has been issued.

When the system issues a major alarm, it starts a timer whose duration depends on the alarm. If the alarm is not canceled before the timer expires, the system takes emergency action to protect itself from the effects of overheating. The timer values and the emergency actions depend on the type of supervisor engine.



Note

Refer to the *Catalyst 4500 Series Switch Module Installation Guide* for information on LEDs, including the startup behavior of the supervisor engine system LED.

Table 14-4 describes the alarms.

Table 14-4 Alarms on Supervisor Engine 6-E, Supervisor Engine 6L-E, and Supervisor Engine 7-E

Event	Alarm Type	Supervisor LED Color	Timeout	Description and Action
Card temperature exceeds the critical threshold.	Major	Red	15 min	Syslog message displays when the alarm is issued. See Table 14-2 for the action on timeout.
Card temperature exceeds the shutdown threshold.	Major	Red	30 sec	Syslog message displays when the alarm is issued. See Table 14-2 for the action on timeout.
Supervisor engine fails power-on self-test (POST).	Major	Red	—	Syslog message displays. Supervisor engine fails to come up.
Chassis fan tray fails.	Major	Red	30 sec	Syslog message displays when the alarm is issued. See Table 14-2 for the action on timeout.
Chassis temperature exceeds the warning threshold.	Minor	Orange	—	Syslog message when the alarm is issued.
Chassis fan tray experiences partial failure.	Minor	Orange	—	Syslog message when the alarm is issued.

Power Management

This section describes the power management feature in the Catalyst 4500 series switches. It includes the following topics:

- [Power Management for the Catalyst 4500 series switches, page 14-7](#)
- [Powering Down a Module, page 14-22](#)



Note

For power consumption of all Catalyst 4000/4500 family modules, see [“Appendix A, Specifications,” in the Catalyst 4500 Series Module Installation Guide](#). Enter the **show power** command to display the current power redundancy and the current system power usage.

Power Management for the Catalyst 4500 series switches

This section includes the following subsections:

- [Supported Power Supplies, page 14-8](#)
- [Power Management Modes for the Catalyst 4500 Switch, page 14-9](#)
- [Selecting a Power Management Mode, page 14-10](#)
- [Power Management Limitations in Catalyst 4500 series switches, page 14-10](#)
- [Available Power for Catalyst 4500 Series Switches Power Supplies, page 14-14](#)
- [Special Considerations for the 4200 W AC and 6000 W AC Power Supplies, page 14-15](#)

- [Combined Mode Power Resiliency, page 14-19](#)
- [Special Considerations for the 1400 W DC Power Supply, page 14-21](#)
- [Special Considerations for the 1400 W DC SP Triple Input Power Supply, page 14-22](#)
- [IEEE 802.3az Energy Efficient Ethernet, page 14-23](#)

Supported Power Supplies

You can select from several different power supplies to ensure that you have enough power for the modules installed in your switch.



Note

You should select a power supply based on the modules and the amount of PoE desired using the Cisco Power Calculator:

<http://tools.cisco.com/cpc/>

The choice between 1000 AC and 1400 AC should depend on the type of line cards that the customer plans to use in the chassis.

The Catalyst 4500 series switches support the following power supplies:

- Fixed Wattage—These power supplies always deliver a fixed amount of PoE and system power.
 - 1000 W AC—Supports up to 1050 W of system power. (Not recommended on the Catalyst 4510R switch, PoE not supported)
 - 1400 W AC—Supports up to 1400 W system power. (PoE not supported)
 - 2800 W AC—Supports up to 1400 W of system power and up to 1400 W of PoE.
- Variable Wattage—These power supplies automatically adjust the wattage to accommodate PoE and system power requirements.
 - 1300 W AC—Supports up to 1050 W of system power and 800 W of PoE, limited to a total of 1300 W.
 - 1400 W DC—Supports up to 1400 W of system power and variable amounts of PoE, depending on the input feed to the power supply. See [“Special Considerations for the 1400 W DC Power Supply” section on page 14-21](#) for more information.
 - 1400 W DC Service Provider—Uses up to three lines (12.5 A, 15 A, 15 A) of DC input and delivers varying amounts of system power ranging from 400 W to 1400 W depending on the lines powered. See [“Special Considerations for the 1400 W DC SP Triple Input Power Supply” section on page 14-22](#) for more information. (PoE not supported)
 - 4200 W AC, 6000 W AC, and 9000W AC—Supports varying amounts of system power and PoE depending on the number of inputs powered and input voltage.



Note

All Catalyst 4500 series switch AC-input power supplies require single-phase source AC. The source AC can be out of phase between multiple power supplies or multiple AC-power plugs on the same power supply because all AC power supply inputs are isolated. Each chassis power supply should ideally have its own dedicated branch circuit sized to local and national codes.

When you insert power supplies in your switch, use power supplies that are of the same wattage. Multi-input power supplies such as 1400 W DC triple-input, 4200 W AC, 6000 W AC, and 9000W AC have additional restrictions. Read the sections on special considerations for these power supplies. If you mix power supplies, the switch uses the one with the higher wattage and ignores the other power supply. The power supply status displays as err-disable and the summary displays as all zeros (0) for wattage values in the output for the **show power** command.

The following example shows the output for the **show power** command for mixed power supplies:

```
Switch# show power
Power
Supply  Model No          Type      Status      Fan      Inline
        Sensor      Status
-----  -
PS1     PWR-C45-2800AC         AC 2800W   good       good     good
→ PS2     PWR-C45-1000AC         AC 1000W   err-disable good     n.a.

*** Power Supplies of different type have been detected***

Power supplies needed by system      :1
Power supplies currently available :1

Power Summary
(in Watts)          Used      Maximum
-----  -
System Power (12V)    328      1360
Inline Power (-50V)    0        1400
Backplane Power (3.3V) 10        40
-----  -
Total Used            338 (not to exceed Total Maximum Available = 750)
Switch#
```

Power Management Modes for the Catalyst 4500 Switch

The Catalyst 4500 series switches support two power management modes:

- **Redundant mode**—Redundant mode uses one power supply as a primary power supply and the second power supply as a back-up. If the primary power supply fails, the second power supply immediately supports the switch without any disruption in the network. Both power supplies must be the same wattage. A single power supply must have enough power to support the switch configuration.
- **Combined mode**—Combined mode uses the power from all installed power supplies to support the switch configuration power requirements. However, combined mode has no power redundancy. If a power supply fails, one or more modules might shut down.



Note On the Catalyst 4510R switch, the 1000 W AC power supply is not enough to support redundant mode for all possible configurations. It is able to support redundant mode for limited configurations that require less than 1050 W.



Note The 1400 W DC power supply supports combined mode for data power. It does not support combined mode for PoE power.

Selecting a Power Management Mode

By default, a switch is set to redundant mode. In the **show power** command, if the **power supplies needed by system** is 1, the switch is in redundant mode; if the **power supplies needed by system** is 2, the switch is in combined mode.

Your switch hardware configuration dictates which power supply or supplies you should use. For example, if your switch configuration requires more power than a single power supply provides, use the combined mode. In combined mode, however, the switch has no power redundancy. Consider the following possibilities:

- The supervisor engine consumes 110 W, the fan boxes for the Catalyst 4503 switch consume 30 W each, the fan boxes for the Catalyst 4506 and Catalyst 4507 switches consume 50 W each, the backplane for the Catalyst 4503 and Catalyst 4506 switches consumes 10 W, and the backplane for the Catalyst 4507 switch consumes 40 W.
- 1000 W can support a fully loaded Catalyst 4503 switch with no powered device support.
- 1300 W can support a fully loaded Catalyst 4503 switch with Cisco powered devices.
- Each PoE port on a WS-X4148-RJ45V module requires 6.3 W. Five fully loaded WS-X4148-RJ45V modules in a switch comprise 240 ports. This configuration requires 1512 W of PoE, plus 300 W for the modules.

Power Management Limitations in Catalyst 4500 series switches

Limitation 1

It is possible to configure a switch that requires more power than the power supplies provide. The two ways you could configure a switch to exceed the power capabilities are as follows:

- The power requirements for the installed modules exceed the power provided by the power supplies. If you insert a single power supply and then set the switch to combined mode, the switch displays this error message:

```
Insufficient power supplies present for specified configuration.
```

This error message also displays in the output for the **show power** command. This error message displays because, by definition, combined mode requires that two working power supplies be installed in your switch.

If the power requirements for the installed modules exceeds the power provided by the power supplies, the switch displays this error message:

```
Insufficient power available for the current chassis configuration.
```

This error message also appears in the **show power** command output.

If you attempt to insert additional modules into your switch and exceed the power supply, the switch immediately places the newly inserted module into reset mode, and the switch displays these error messages:

```
Module has been inserted
Insufficient power supplies operating.
```

Additionally, if you power down a functioning switch and insert an additional module or change the module configuration so that the power requirements exceed the available power, one or more modules enter reset mode when you power on the switch again.

- The power requirements for the PoE exceed the PoE provided by the power supplies.

If you have too many IP phones drawing power from the system, power to IP phones is cut, and some phones may be powered down to reduce the power requirements to match the power supplies.

In the first scenario (power requirements exceed the power supplied), the system attempts to resolve this power usage limitation by evaluating the type and number of modules installed. During the evaluation cycle, beginning from the bottom of the chassis, the system puts the modules that it is unable to support (for lack of power) into reset mode. The supervisor engine and modules for which there is adequate power always remain enabled, with no disruption of network connectivity. Modules placed in reset mode still consume some power and can be removed from the chassis to further reduce power requirements. If you configure the chassis correctly, the system does not enter the evaluation cycle.

A module in reset mode continues to draw power as long as it is installed in the chassis; use the **show power module** command to determine how much power is required to bring the module online.

To compute the power requirements for your system and verify that your system has enough power, add the power consumed by the supervisor engine module(s), the fan box(es), and the installed modules (including PoE). For PoE, total the requirements for all the phones. See the [“Powering Down a Module” section on page 14-22](#) for more information on the power consumption for the various components of your switch.

The 802.3af-compliant PoE modules can consume up to 20 W of PoE to power FPGAs and other hardware components on the module. Be sure to add at least 20 W to your PoE requirements for each 802.3af-compliant PoE module to ensure that the system has adequate power for the PDs connected to the switch.

On the WS-X4148-RJ45V PoE module, PoE consumption cannot be measured. For all PoE calculations, the PoE consumption on this module is presumed to be equal to its administrative PoE.

Use the **show module** command to verify which modules are active and which, if any, have been placed in reset.

The following example shows the **show module** command output for a system with inadequate power for all installed modules. The system does not have enough power for Module 5; the *Status* displays it as *PwrDeny*.

If the PoE that is consumed by the module is more than 50 W above the PoE you allocated using the **power inline consumption default** command, the Status displays as *PwrOver*. If the PoE consumed by the module is more than 50 W above the PoE module limit, the Status displays as *PwrFault*.

```
Switch# show module
Mod  Ports Card Type                               Model                Serial No.
-----+-----+-----+-----+-----+-----+-----+-----+-----+
1      2  1000BaseX (GBIC) Supervisor(active)  WS-X4014             JAB054109GH
2      6  1000BaseX (GBIC)                               WS-X4306             00000110
3     18  1000BaseX (GBIC)                               WS-X4418             JAB025104WK
→ 5      0  Not enough power for module          WS-X4148-FX-MT       00000000000
6     48  10/100BaseTX (RJ45)                     WS-X4148             JAB023402RP

M MAC addresses                               Hw  Fw  Sw  Status
-----+-----+-----+-----+-----+-----+-----+-----+
1 005c.9d1a.f9d0 to 005c.9d1a.f9df 0.5 12.1 (11br) EW 12.1 (20020313:00) Ok
2 0010.7bab.9920 to 0010.7bab.9925 0.2                               Ok
3 0050.7356.2b36 to 0050.7356.2b47 1.0                               Ok
→ 5 0001.64fe.a930 to 0001.64fe.a95f 0.0                               PwrDeny
6 0050.0f10.28b0 to 0050.0f10.28df 1.0                               Ok
Switch#
```

Limitation 2

Certain configurations on the Catalyst 4507R and Catalyst 4510R chassis exceeds the maximum amount of data power available. These configurations include the combination of the follow PIDs:

- 7-slot configuration
- Chassis: WS-C4507R-E, WS-C4510R-E
- Dual supervisor engines: WS-X45-Sup6-E and WS-X45-Sup6L-E
- One or more: WS-X4448-GB-RJ45 or WS-X4148-FX-MT

To maximize the 10/100/1000 port density of 7- and 10- slot chassis, install WS-X4548-GB-RJ45 line cards instead of WS-X4448-GB-RJ45 line cards. If WS-X4448-GB-RJ45 line cards are required, two options are available:

- Option 1
Only four line card slots can be used on the Catalyst 4507R and six line card slots on the Catalyst 4510R chassis.
- Option 2
When all slots are required, only one WS-X4448-GB-RJ45 line card can be used.

To maximize the 100-BASE-FX port density of 7- and 10- slot chassis, install WS-X4248-FE-SFP line cards with FX optics instead of WS-X4148-FX-MT line cards. If WS-X4148-FX-MT line cards are required, two options are available:

- Option 1
Only four line card slots can be used on the Cat4507R and six line card slots on the Catalyst 4510R chassis.
- Option 2
When all slots are required only one WS-X4448-GB-RJ45 line card can be used.

Configuring Redundant Mode on Catalyst 4500 series switches

By default, the power supplies in a Catalyst 4500 series switch are set to operate in redundant mode. To effectively use redundant mode, follow these guidelines:

- Use two power supplies of the same type.
- If you have the power management mode set to redundant mode and only one power supply installed, your switch accepts the configuration but operates without redundancy.



Caution

If you have power supplies with different types or different wattages installed in your switch, the switch does not recognize one of the power supplies and does not have power redundancy.

- For fixed power supplies, choose a power supply that is powerful enough to support the switch configuration.
- For variable power supplies, choose a power supply that provides enough power so that the chassis and PoE requirements are less than the maximum available power. Variable power supplies automatically adjust the power resources at startup to accommodate the chassis and PoE requirements. Modules are brought up first, followed by IP phones.
- The maximum available power for chassis and PoE for each power supply are listed in [Table 14-5 on page 14-14](#).

To configure redundant mode on your Catalyst 4500 series switch, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters configuration mode.
Step 2	Switch(config)# power redundancy-mode redundant	Sets the power management mode to redundant mode.
Step 3	Switch(config)# end	Exits configuration mode.
Step 4	Switch# show power supplies	Verifies the power redundancy mode for the switch.

The following example shows how to set the power management mode to redundant mode:

```
Switch (config)# power redundancy-mode redundant
Switch (config)# end
Switch#
```

The following example shows how to display the current power redundancy mode. The power supplies needed by system:1 indicates that the switch is in redundant mode.

```
Switch# show power supplies
Power supplies needed by system:1
Switch#
```

An option in the combined mode provides a form of redundancy available with only the 4200 W AC and 6000 W AC power supplies. Refer to the section “Combined Mode Power Resiliency” on page 19.

Configuring Combined Mode on Catalyst 4500 series switches

If your switch configuration requires more power than a single power supply can provide, set the power management mode to combined mode. Combined mode utilizes the available power for both power supplies; however, your switch has no power redundancy.

To effectively use combined mode, follow these guidelines:

- Use power supplies of the same type and wattage (fixed or variable and AC or DC).
- If you use power supplies with different types or wattages, the switch utilizes only one of the power supplies.
- For variable power supplies, choose a power supply that provides enough power so that the chassis and PoE requirements are less than the maximum available power. Variable power supplies automatically adjust the power resources at startup to accommodate the chassis and PoE requirements.
- If you have the power management mode set to combined mode and only one power supply installed, your switch accepts the configuration, but power is available from only one power supply.
- When your switch is configured to combined mode, the total available power is not the mathematical sum of the individual power supplies. The power supplies have a predetermined current sharing ratio. See [Table 14-5 on page 14-14](#) for more information.
- The maximum available power for chassis and PoE for each power supply are listed in [Table 14-5 on page 14-14](#).

To configure combined mode on your Catalyst 4500 series switch, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters configuration mode.
Step 2	Switch(config)# power redundancy-mode combined	Sets the power management mode to combined mode.
Step 3	Switch(config)# end	Exits configuration mode.
Step 4	Switch# show power supplies	Verifies the power redundancy mode for the switch.

The following example shows how to set the power management mode to combined mode:

```
Switch(config)# power redundancy-mode combined
Switch(config)# end
Switch#
```

The following example shows how to display the current power redundancy mode. The power supplies needed by system: 2 indicates that the switch is in combined mode.

```
Switch# show power supplies
Power supplies needed by system:2
Switch#
```

Available Power for Catalyst 4500 Series Switches Power Supplies

Table 14-5 lists the power available for use in the various Catalyst 4500 series switches power supplies. When your switch is configured to combined mode, the total available power is not the mathematical sum of the individual power supplies. The power supplies have a sharing ratio predetermined by the hardware. In combined mode, the total power available is $P + (P * \text{sharing-ratio})$, where P is the amount of power in the power supply.

Table 14-5 Available Power for Switch Power Supplies

Power Supply	Redundant Mode (W)	Combined Mode (W)	Sharing Ratio
1000 W AC	Chassis ¹ = 1050 PoE = 0	Chassis = 1667 PoE = 0	2/3
1300 W AC	Chassis (max) = 1050 PoE (max) = 800 Chassis + PoE + Backplane \leq 1300	Chassis (min) = 767 PoE (max) = 1333 Chassis (max) = 1667 PoE (min) = 533 Chassis + PoE + Backplane \leq 2200	2/3
1400 W DC	Chassis (min) = 200 Chassis (max) = 1360 $\text{PoE (max)}^2 = (\text{DC Input}^3 - [\text{Chassis (min)} + \text{Backplane}] / 0.75) * 0.96$	Chassis = 2267 ⁴ PoE ⁵	Chassis—2/3 PoE—0

Table 14-5 Available Power for Switch Power Supplies (continued)

Power Supply	Redundant Mode (W)	Combined Mode (W)	Sharing Ratio
1400 W AC	Chassis = 1360 PoE = 0 ⁶	Chassis = 2473 PoE = 0	9/11
2800 W AC	Chassis = 1360 PoE = 1400	Chassis = 2473 PoE = 2333	Chassis ⁷ —9/11 PoE ⁸ —2/3

1. Chassis power includes power for the supervisor engine(s), all line cards, and the fan tray.
2. The efficiency for the 1400 W DC power supply is 0.75, and 0.96 is applied to PoE.
3. DC input can vary for the 1400 W DC power supply and is configurable. For more information, see “Special Considerations for the 1400 W DC Power Supply” on page 21.
4. Not available for PoE.
5. Not available for PoE.
6. No voice power.
7. Data-only.
8. Inline power.

Special Considerations for the 4200 W AC and 6000 W AC Power Supplies

The 4200 W AC and 6000 W AC power supply has two inputs: each can be powered at 110 or 220 V.

The 9000 W AC power supply has three inputs: each can be powered at 110 or 220V.

The output of the **show power** command for the 4200 W AC, 6000 W AC, and 9000W AC power supplies are similar to that of 1400 W DC triple-input power supply (that is, the status of the submodules (multiple inputs) is displayed). With these power supplies, you can distinguish submodule “failed” versus “off,” and the status of the submodules (good, bad, or off):

```
Switch# show power
Power
Supply  Model No          Type      Status      Fan      Inline
        -----
PS1     PWR-C45-4200ACV      AC 4200W  good        good      good
PS1-1                   220V      good
PS1-2                   off
PS2     PWR-C45-4200ACV      AC 4200W  bad/off     good      bad/off
PS2-1                   220V      good
PS2-2                   220V      bad

Power supplies needed by system      : 1
Power supplies currently available   : 2

Power Summary
(in Watts)      Used      Maximum
-----
System Power (12V)      140      1360
Inline Power (-50V)      0      1850
Backplane Power (3.3V)   0      40
-----
Total              140 (not to exceed Total Maximum Available = 2100)
Switch#
```

```
Switch# show power
Power
Supply  Model No          Type      Status      Fan      Inline
        -----
PS1     PWR-C45-4200ACV      AC 4200W  good        good      good
PS1-1                   220V      good
PS1-2                   off
PS2     PWR-C45-4200ACV      AC 4200W  bad/off     good      bad/off
PS2-1                   220V      good
PS2-2                   220V      bad
```

```

PS1      PWR-C45-9000ACV  AC 9000W  good      good      good
PS1-1    220V  good
PS1-2    220V  good
PS1-3    220V  good
PS2      PWR-C45-9000ACV  AC 9000W  good      good      good
PS2-1    220V  good
PS2-2    220V  good
PS2-3    220V  good

```

```

Power supplies needed by system      : 2 Maximum Inputs = 3
Power supplies currently available   : 2

```

Power Summary (in Watts)	Used	Maximum Available
-----	----	-----
System Power (12V)	1323	2646
Inline Power (-50V)	0	6022
Backplane Power (3.3V)	40	67
-----	----	-----

As with other power supplies, the two power supplies must be of the same type (6000 W AC or 4200 W AC or 1400 W DC). Otherwise, the right power supply is put in err-disable state and the left one is selected. In addition, all the inputs to the chassis must be at the same voltage. In redundant mode, the inputs to the left and right power supplies must be identical. If the left and right power supplies are powered in redundant mode, the power values is based on the power supply with the higher output wattage.

**Note**

When the system is powered with a 4200 W, 6000 W, or 9000W power supply either in 110 V or 220 V combined mode operation, the available power is determined by the configuration of the system (the type of line cards, the number of line cards, number of ports consuming inline power, etc.) and does not reflect the absolute maximum power.

**Note**

In a matched redundant power supply configuration, if a power supply submodule fails, the other (good) power supply provides power to its full capability.

Table 14-6 illustrates how the 4200 W AC power supply is evaluated in redundant mode.

Table 14-6 Output Power in Redundant Mode for the 4200 W AC Power Supply

Power Supply	12 V (data) (W)	-50V (PoE) (W)	Total Power (W) ¹
110 V AC	660	922	1050
110 V AC + 110 V AC	1460	2000	2100
220 V AC	1460	2500	2100
220 V AC + 220 V AC	1960	5000	4200

1. Power supply outputs' drawing should not exceed the total power.

In combined mode, all the inputs to the chassis must be at the same voltage.

Table 14-7 illustrates how the 4200 W AC power supply is evaluated in combined mode.

Table 14-7 Output Power in Combined Mode for the 4200 W AC Power Supply

Power Supply	12 V (data) (W)	-50 V (PoE) (W)	Total Power (W) ¹
Both sides at 110 V AC	1188	1531	1700
Both sides at 110 V AC + 110 V AC	2448	3071	3570
One side at 110 V AC + 110 V AC, the other at 110 V AC	1818	2301	2660
Both sides at 220 V AC	2448	3071	3570
Both sides at 220 V AC + 220 V AC	2448	6142	7070
Both sides at 220 V AC + 220 V AC, the other at 220 V AC	2447	4607	5320

1. Power supply outputs' drawing should not exceed the total power.

Table 14-8 illustrates how the 6000 W AC power supply is evaluated in redundant mode.

Table 14-8 Output Power in Redundant Mode for the 6000 W AC Power Supply

Power Supply	12 V (data) (W)	-50V (PoE) (W)	Total Power (W) ¹
110 V AC	850	922	1050
110 V AC + 110 V AC	1700	1850	2100
220 V AC	2200	2400	3000
220 V AC + 220 V AC	2200	4800	6000

1. Power supply outputs' drawing should not exceed the total power.

In combined mode, all the inputs to the chassis must be at the same voltage.

Table 14-9 illustrates how the 6000 W AC power supply is evaluated in combined mode.

Table 14-9 Combined Mode Output for the 6000 W AC Power Supply

Power Supply	12 V (data) (W)	-50 V (PoE) (W)	Total Power (W) ¹
Both sides at 110 V AC	1530	1531	1710
Both sides at 110 V AC + 110 V AC	3060	3071	3590
One side at 110 V AC + 110 V AC, the other at 110 V AC	2295	2301	2680
Both sides at 220 V AC	3960	3984	5140
Both sides at 220 V AC + 220 V AC	3960	7968	10170
Both sides at 220 V AC + 220 V AC, the other at 220 V AC	2970	5976	7610

1. Power supply outputs' drawing should not exceed the total power.

Table 14-10 illustrates how the 9000 W AC power supply is evaluated in redundant mode.

Table 14-10 Power Output in Redundant Mode for the 9000 W AC Power Supply

Power Supply	12V (data) (W)	-50V (PoE) (W)	¹ Total Power (W)
110VAC	960	1000	1100
110VAC + 110 VAC	1460	2000	2200
110VAC + 110 V AC+ 110VAC	1460	2500	3300
220VAC	1460	2500	3000
220VAC + 220VAC	1960	5000	6000
220VAC + 220VAC + 220VAC	1960	7500	9000

1. Power supply output drawings should not exceed the total power.

Table 14-11 illustrates how the 9000 W AC power supply is evaluated in combined mode.

Table 14-11 Power Output in Combined Mode for the 9000 W AC Power Supply

Power Supply	12V (data) (W)	-50V (PoE) (W)	¹ Total Power (W)
Both sides at 110 VAC	1594	1660	1790
Both sides at 110VAC + 110VAC	2424	3320	3610
Both sides at 110VAC + 110VAC + 110VAC	2424	4150	5420
One side at 110VAC + 110VAC + 110VAC, the other at 110VAC + 110VAC	2020	3458	4510
One side at 110VAC + 110VAC + 110VAC, the other at 110VAC	1615	2767	3600
One side at 110VAC + 110VAC, the other at 110VAC	1818	2490	2700
Both sides at 220VAC	2424	4150	4930
Both sides at 220VAC + 220VAC	3763	8300	10140
Both sides at 220VAC + 220VAC + 220VAC	3763	14400	17210
One side at 220VAC + 220VAC + 220VAC, the other at 220VAC + 220VAC	2940	11250	13430
One side at 220VAC + 220VAC + 220VAC, the other at 220VAC	2169	8300	9900
One side at 220VAC + 220VAC, the other at 220VAC	2646	6225	7410

1. Power supply output drawings should not exceed the total power.

Combined Mode Power Resiliency



Note

This feature only applies in combined mode when both power supply bays contain the 4200 W AC or 6000 W AC power supply.

Using the combined mode power resiliency feature, you can limit the power usage to a maximum of two or three (configurable) inputs for 4000W and 6000W power supplies. For 9000W power supplies, you can limit the power usage to a maximum of 2 to 5 inputs, since the 9000W is a triple input power supply.

With two 4200 W AC or 6000 W AC power supplies, a maximum of four inputs are available. With two 9000W, a maximum of six inputs are available. This feature allows you to cap the power usage to that of two/three inputs or four/five inputs. If one of the power supplies fails, no loss of power occurs because you have capped its usage to a smaller number of inputs.

To configure the combined mode resiliency feature, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters configuration mode.
Step 2	Switch(config)# power redundancy combined max inputs {2 3}	If a 9000W AC power supply is detected, this limits the power usage to four or five inputs.
	OR	
	Switch(config)# power redundancy combined max inputs {2 3}	If a 9000W AC power supply is not detected, limits the power usage to two or three inputs.
Step 3	Switch(config)# end	Exits configuration mode.

If you have **max inputs 3** configured with four “good” (220 V) inputs and you limit the user to 5500 W instead of 7600 W and one subunit fails or is powered off, you have three quality inputs providing 5500 W and the chassis is powered at the same rate as it was prior to the failure event:

```
Switch# configuration terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# power redundancy combined max inputs 3
Switch(config)# end
Switch#
14:32:01: %SYS-5-CONFIG_I: Configured from console by console
```

Here is the output of the **show power** command prior to invoking this feature:

```
Switch# show power
sh power
Power
Supply  Model No          Type      Status      Fan      Inline
-----  -
PS1      PWR-C45-4200ACV    AC 4200W  good        good      good
PS1-1    110V               good
PS1-2    110V               good
```

```

PS2      PWR-C45-4200ACV  AC 4200W  good      good      good
PS2-1                    110V  good
PS2-2                    110V  good

```

```

Power supplies needed by system : 1
Power supplies currently available : 2

```

```

Power Summary
(in Watts)      Used      Maximum
-----
System Power (12V)  140      1360
Inline Power (-50V)   0      1850
Backplane Power (3.3V) 0       40
-----
Total            140 (not to exceed Total Maximum Available = 2100)

```

Here is the output after invoking this feature. The combined mode was indicated before **Power supplies needed = 2** in the output of the **show power** command, combined mode is now indicated by the phrase **Power supplies needed by system: 2 Maximum Inputs = 3**.

```
Switch# show power
```

```

sh power
Power
Supply Model No      Type      Status      Fan      Inline
-----
PS1      PWR-C45-4200ACV  AC 4200W  good      good      good
PS1-1                    110V  good
PS1-2                    110V  good
PS2      PWR-C45-4200ACV  AC 4200W  good      good      good
PS2-1                    110V  good
PS2-2                    110V  good

```

```

Power supplies needed by system : 2 Maximum Inputs = 3
Power supplies currently available : 2

```

```

Power Summary
(in Watts)      Used      Maximum
-----
System Power (12V)  140      2400
Inline Power (-50V)   0      2000
Backplane Power (3.3V) 0       40
-----
Total            140 (not to exceed Total Maximum Available = 2728)

```

```
Switch#
```

Here's another example of combined mode resiliency with 9000W power supply with a maximum of six active inputs, limited to 3 inputs:

```
Switch# show power
```

```

Power
Supply Model No      Type      Status      Fan      Inline
-----
PS1      PWR-C45-9000ACV  AC 9000W  good      good      good
PS1-1                    220V  good
PS1-2                    220V  good
PS1-3                    220V  good
PS2      PWR-C45-9000ACV  AC 9000W  good      good      good
PS2-1                    220V  good
PS2-2                    220V  good
PS2-3                    220V  good

```

```

Power supplies needed by system : 2 Maximum Inputs = 3

```

Power supplies currently available : 2

Power Summary (in Watts)	Used	Maximum Available
-----	----	-----
System Power (12V)	1323	2646
Inline Power (-50V)	0	6022
Backplane Power (3.3V)	40	67
-----	----	-----
Total	1363	(not to exceed Total Maximum Available = 7412)

Special Considerations for the 1400 W DC Power Supply



Caution

Do not mix the 1400 W DC power supply with any other power supply, even for a hot swap or other short-term emergency. Doing so can seriously damage your switch.

Keep in mind the following guidelines when using a 1400 W DC power supply with your Catalyst 4500 series switch:

- The 1400 W DC power supply works with a variety of DC sources. The DC input can vary from 300 W to 7500 W. Refer to the power supply documentation for additional information.
- The supervisor engine cannot detect the DC source plugged into the 1400 W DC power supply. If you are using the 1400 W DC power supply, use the **power dc input** command to set the DC input power. For more information on this command, see the [“Configuring the DC Input for a Power Supply” section on page 14-21](#).
- The software automatically adjusts between system power (for modules, backplane, and fans) and PoE. Although PoE is 96 percent efficient, system power has only 75 percent efficiency. For example, each 120 W of system power requires 160 W from the DC input. This requirement is reflected in the **“Power Used”** column of the output for the **show power available** command.
- The 1400 W DC power supply has a separate power on or off switch for PoE. The power supply fan status and main power supply status are tied together. If either of them fails, both the power supply and its fan report as bad/off. You should verify that the main power is on before turning on the power for the inline switch. In addition, you should verify that the power for the inline switch is off before turning off the main power.

Configuring the DC Input for a Power Supply

To configure the DC input power for the 1400 W DC power supply or a power shelf, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters configuration mode.
Step 2	Switch(config)# power dc input watts	Sets the capacity of the DC input source.
Step 3	Switch(config)# end	Exits configuration mode.

The same configuration is applied to both power slots. For example, if you set the **dc power input** to 1000 W, the switch expects 1000 W as the external DC source for both slot 1 and slot 2 (if present).

The following example shows how to set the external DC power source to 1000 W:

```
Switch# configure terminal
Switch (config)# power dc input 1000
```

```
Switch (config)# end
Switch#
```

If you use the 1400 W DC SP power supply in combined mode, the inputs do not have to match.

Special Considerations for the 1400 W DC SP Triple Input Power Supply

Unlike the 1400 W DC power supply, the 1400 W DC SP power supply has submodules (multiple inputs) that can be powered on or off. With Cisco IOS Release 12.2(25)EW, the output of the **show power** command is modified to display the status of these submodules:

```
Switch# show power
Power
Supply Model No          Type      Status      Fan      Inline
-----
PS1      PWR-C45-1400DC        DCSP1400W good        good      n.a.
PS1-1    12.5A                 good
PS1-2    15.0A                 bad
PS1-3    15.0A                 off

PS2      none                  --        --        --        --
```

Observer the following guidelines when using a 1400 W DC SP power supply with your Catalyst 4500 series switch:

- When you use two 48 V power rails to drive two power supplies, you might use cross-wiring to connect the power supplies (to rails) to minimize the inrush current drawn during an initial power up. In this situation, you should configure the switch in combined mode before you take a rail down for maintenance.
- Ordinarily, when configured for redundancy, two power supplies must be matched (have identical inputs). For example, you might provide power to inputs 1 and 3 on both PS1 and PS2. If power supplies are mismatched upon bootup, the right (second) power supply is in err-disable state.

In a matched redundant power supply configuration, if a power supply submodule fails, the other (good) power supply provides power to its full capability.

Powering Down a Module

If your system does not have enough power for all modules installed in the switch, you can power down a module, and place it in low-power mode. To power down a module, perform this task:

Command	Purpose
Switch(config)# no hw-module module num power	Turns power down to the specified module by placing it in low power mode.

To power on a module that has been powered down, perform this task:

Command	Purpose
Switch(config)# hw-module module num power	Turns power on to the specified module.

This example shows how to power down module 6:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# no hw-module module 6 power
Switch(config)# end
Switch#
```

**Note**

After you enter **no hw-mod mod x power** command and OIR the linecard, the configuration persists and is valid for any slot in the chassis it is applied to. You observe the same behavior in the active and standby supervisor engines

IEEE 802.3az Energy Efficient Ethernet

**Note**

EEE is supported on WS-X4748-UPOE+E and WS-X4748-RJ45-E.

Energy Efficient Ethernet is an extension of the IEEE 802.3 standard that provides a mechanism and a standard for reducing energy usage without reducing the vital function of network interfaces. EEE defines the signaling necessary for energy savings during periods where no data is sent on the interface.

EEE defines support for physical layer devices (PHYs) to operate in Low Power Idle (LPI) mode. When enabled, EEE supports QUIET times during low link utilization allowing both sides of a link to disable portions of each PHY's operating circuitry and save power. This functionality is provided per port and is not enabled by default. To avoid issues with EEE functionality on any port during run-time, Cisco provides the **power efficient-ethernet auto** command to enable or disable EEE.

Because EEE relies on Auto Negotiation pulse to determine whether to activate EEE, the port must initially enable auto negotiation. Furthermore, EEE is the correct action provided the speed is auto 100M, auto 1000M, or auto 100M and 1000M. 10M (either auto or forced mode) does not require EEE for power saving.

For more details, see the URL:

http://www.cisco.com/en/US/prod/collateral/switches/ps5718/ps4324/white_paper_c11-676336.pdf

Sections include:

- [Determining EEE Capability, page 14-23](#)
- [Enabling EEE, page 14-24](#)
- [Determining EEE Status, page 14-24](#)

Determining EEE Capability

To determine whether a line card supports EEE, use the **show interface capabilities module *module*** command, as follows:

```
Switch# show interface capabilities module 3
GigabitEthernet3/1
Model: WS-X4748-NGPOE+E-RJ-45
Type: 10/100/1000-TX
Speed: 10,100,1000,auto
Duplex: half,full,auto
Auto-MDIX: yes
EEE: yes ( 100-Tx and 1000-T auto mode )
```

Enabling EEE

To enable EEE on a given port, use the **power efficient-ethernet auto** command.

	Command	Purpose
Step 1	Switch# configure terminal	Enters configuration mode.
Step 2	Switch(config)# interface <i>interface</i>	Enters interface configuration mode and specifies the port to be configured.
Step 3	Switch(config-if)# power efficient-ethernet auto	Enables EEE.
Step 4	Switch(config-if)# exit	Exits global configuration mode.

The following example shows how to enable EEE:

```
Switch# conf t
Switch(config)# interface gigabitethernet 1/1
Switch(config-if)# power efficient-ethernet auto
Switch(config-if)# exit
```

Determining EEE Status

To determine EEE status use the **show platform software interface *interface* status** command:

The following example determines EEE status:

```
Switch(config)# show platform software interface g2/1 status
Switch Phyport Gi2/1 Software Status
EEE: Disabled
```

EEE status can have the following values:

- EEE: N/A**—The port is not capable of EEE.
- EEE: Disabled**—The port EEE is disabled.
- EEE: Disagreed**—The port EEE is not set because a remote link partner might be incompatible with EEE; either it is not EEE capable, or it’s EEE setting is incompatible.
- EEE: Operational**—The port EEE is enabled and operating.



Configuring Power over Ethernet



Note

Before reading this chapter, read the “Preparing for Installation” section of the *Catalyst 4500 Series Installation Guide*. You must ensure that your installation site has enough power and cooling to accommodate the additional electrical load and heat introduced by PoE.

This chapter describes how to configure Power over Ethernet (PoE) on the Catalyst 4500 series switch.

This chapter contains the following sections:

- [About Power over Ethernet, page 15-2](#)
- [Power Management Modes, page 15-3](#)
- [Configuring Power Consumption for Powered Devices on an Interface, page 15-5](#)
- [Displaying the Operational Status for an Interface, page 15-7](#)
- [Displaying all PoE Detection and Removal Events, page 15-8](#)
- [Displaying the PoE Consumed by a Module, page 15-8](#)
- [PoE Policing and Monitoring, page 15-12](#)
- [The WS-X4648-RJ45V-E, WS-X4648-RJ45V+E, and WS-X4548-RJ45V+ switching modules support IEEE 802.3af PoE as well as the Cisco proprietary Inline Power standard. With Cisco IOS Release 12.2\(44\)SG, the WS-X4648-RJ45V+E line card can also support the IEEE 802.3at standard with up to 30 W available per-port. The WS-X4648-RJ45V-E line card also supports up to 20 W. The WS-X4548-RJ45V+ switching module is supported with Cisco IOS Release 12.2\(50\)SG and can provide up to 30 W of inline power per-port., page 15-16](#)
- [Additional References for Power over Ethernet, page 15-18](#)
- [Feature Information for Power over Ethernet, page 15-18](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About Power over Ethernet

The Catalyst 4500 series switch provides Power over Ethernet (PoE) support for both Cisco Prestandard PoE and the IEEE 802.3af standard (ratified in 2003). PoE is supported by all Catalyst 4500 series chassis and requires a PoE module and power supply. The amount of PoE power available depends on the PoE capabilities of individual power supplies. Support for PoE enables the system to power inline devices, such as IP phones, IP video phones, and wireless access points over standard copper cabling (Category 5, 5e, or 6 cabling).

In addition, with PoE, you do not need to provide wall power for each PoE enabled device. This eliminates the cost for additional electrical cabling that is otherwise necessary for connected devices. Moreover, PoE enables you to isolate critical devices on a single power system, enabling the entire system to be supported by UPS backup.

You typically deploy a Catalyst 4500 series switch in one of two deployment scenarios. The first scenario is data-only, which requires power to operate the switch and the associated modules. The second scenario supports data and PoE (also termed “inline power”) for deployments where the attached device derives power from the Ethernet port.

Catalyst 4500 series switches can sense if a powered device is connected to a PoE module. They can supply PoE to the powered device if there is no power on the circuit. (If there is power on the circuit, the switch does not supply it.) The powered device can also be connected to an AC power source and supply its own power to the voice circuit.


Note

You should select the amount of PoE desired using the Cisco Power Calculator:

<http://tools.cisco.com/cpc/>

Hardware Requirements

To power a device using PoE, your chassis must use at least one of the power supplies listed in [Table 15-1](#), and connect the device to at least one of the switching modules listed in [Table 15-1](#).

Table 15-1 **Hardware Requirements**

Switching Modules	Power Supplies
WS-X4148-RJ45V	PWR-C45-1300ACV=
WS-X4224-RJ45V	PWR-C45-1400DCV=
WS-X4248-RJ21V	PWR-C45-2800ACV=
WS-X4248-RJ45V	PWR-C45-4200ACV=
WS-X4506-GB-T	
WS-X4524-GB-RJ45V	
WS-X4548-RJ45V+	
WS-X4548-GB-RJ45V	
WS-X4648-RJ45V-E	
WS-X4648-RJ45V+E	

Power Management Modes

If your switch has a module capable of providing PoE to end stations, you can set each interface on the module to automatically detect and apply PoE if the end station requires power.

The Catalyst 4500 series switch has three PoE modes:

- **auto**—PoE interface. The supervisor engine directs the switching module to power up the interface *only* if the switching module discovers the phone and the switch has enough power. You can specify the maximum wattage that is allowed on the interface. If you do not specify a wattage, then the switch delivers no more than the hardware-supported maximum value. This mode has no effect if the interface is not capable of providing PoE.
- **static**—High priority PoE interface. The supervisor engine preallocates power to the interface, even when nothing is connected, guaranteeing that power exists for the interface. You can specify the maximum wattage that is allowed on the interface. If you do not specify a wattage, then the switch preallocates the hardware-supported maximum value. If the switch does not have enough power for the allocation, the command fails. The supervisor engine directs the switching module to power up the interface *only* if the switching module discovers the powered device.
- **never**—Data interface only. The supervisor engine never powers up the interface, even if an unpowered phone is connected. This mode is only needed when you want to make sure power is never applied to a PoE-capable interface.

The switch can measure the actual PoE consumption for an 802.3af-compliant PoE module, and displays this in the **show power module** command.

PoE consumption cannot be measured on the WS-X4148-RJ45V PoE module. For all PoE calculations, the PoE consumption on this module is presumed to be equal to its administrative PoE.

For more information, see the [“Displaying the PoE Consumed by a Module” section on page 15-8](#).

For most users, the default configuration of “auto” works well, providing plug-and-play capability. No further configuration is required. However, to make an interface higher priority or data only, or to specify a maximum wattage, perform this task:

	Command	Purpose
Step 1	Switch(config)# interface { fastethernet gigabitethernet } <i>slot/port</i>	Selects the interface to configure.
Step 2	Switch(config-if)# power inline { auto [max <i>milli-watts</i>] never static [max <i>milli-watts</i>]}	<p>The auto keyword sets the interface to automatically detect and supply power to the powered device. it is the default configuration.</p> <p>The static keyword sets the interface to higher priority than auto.</p> <p>If necessary, use the max keyword to specify the maximum wattage allowed on the interface (4000 to 15400 milliwatts for most switching modules. As of Cisco IOS Release 12.2(44)SG, the WS-X4648-RJ45V+E can support up to 30 W available per-port and the WS-X4648-RJ45V-E supports up to 20 W. For more information, see “The WS-X4648-RJ45V-E, WS-X4648-RJ45V+E, and WS-X4548-RJ45V+ switching modules support IEEE 802.3af PoE as well as the Cisco proprietary Inline Power standard. With Cisco IOS Release 12.2(44)SG, the WS-X4648-RJ45V+E line card can also support the IEEE 802.3at standard with up to 30 W available per-port. The WS-X4648-RJ45V-E line card also supports up to 20 W. The WS-X4548-RJ45V+ switching module is supported with Cisco IOS Release 12.2(50)SG and can provide up to 30 W of inline power per-port.” on page 16).</p> <p>Use the never keyword to disable detection and power for the PoE capable interface.</p>
Step 3	Switch(config-if)# end	Exits configuration mode.
Step 4	Switch# show power inline { fastethernet gigabitethernet } <i>slot/port</i>	Displays the PoE state for the switch.



Note

If you set a non-PoE-capable interface to automatically detect and apply power, an error message indicates that the configuration is not valid.

The following example shows how to set the Fast Ethernet interface 4/1 to automatically detect PoE and send power through that interface, and to verify the PoE configuration:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface fastethernet 4/1
Switch(config-if)# power inline auto
Switch(config-if)# end
Switch# show power inline fastethernet 4/1
Available:677(w) Used:11(w) Remaining:666(w)
```

Interface	Admin	Oper	Power(Watts)		Device	Class
			From PS	To Device		
Fa4/1	auto	on	11.2	10.0	Ieee PD	0

Interface	AdminPowerMax (Watts)	AdminConsumption (Watts)
-----------	--------------------------	-----------------------------

```
-----  
Fa4/1                15.4                10.0  
Switch#
```

The following example shows how to configure an interface so that it never supplies power through the interface:

```
Switch# configure terminal  
Enter configuration commands, one per line. End with CNTL/Z.  
Switch(config)# interface fastethernet 5/2  
Switch(config-if)# power inline never  
Switch(config-if)# end  
Switch#
```

Intelligent Power Management

All Catalyst 4500 PoE-capable modules use Intelligent Power Management to provide power on each interface. When a powered device (PD) is attached to a PoE-capable port, the port detects the PD and provision power accordingly. If a Cisco PD is used, the switch and PD negotiate power using CDP packets to determine the precise amount of power needed by the PD. If the PD is 802.3af compatible, the difference between what is mandated by the 802.3af class and what is actually needed by the PD is returned to the power budget for use by additional devices. In this way, power negotiation enables customers to stretch their power budget and use it more effectively.

Power negotiation also enables the interoperability of newer Cisco powered devices with older legacy PoE-capable ports from Cisco. Newer Cisco PDs do not consume more than what the switch port can provide.

Configuring Power Consumption for Powered Devices on an Interface

By default, when the switch detects a powered device on an interface, it assumes the powered device consumes the maximum the port can provide (7 W on a legacy PoE module and 15.4W on the IEEE PoE modules introduced in Cisco IOS Release 12.2(18)EW). When the switch receives a CDP packet from the powered device, the wattage automatically adjusts downward to the specific amount required by that device. Normally, this automatic adjustment works well, and no further configuration is required or recommended. However, you can specify the powered device's consumption for a particular interface to provide extra functionality from your switch. This behavior is useful when CDP is disabled or not available.



Note

When manually configuring the consumption for powered devices, you need to account for the power loss over the cable between the switch and the powered device.



Note

The **inline power consumption** command overrides the power allocated to the port through IEEE/Cisco phone discovery and CDP/LLDP power negotiation. To guarantee safe operation of the system, ensure that the value configured here is no less than the actual power requirement of the attached device. If the power drawn by the inline powered devices exceeds the capability of the power supply, it could trip the power supply.

To change the power consumption of a single powered device, perform this task:

	Command	Purpose
Step 1	Switch(config)# interface { fastethernet gigabitethernet } <i>slot/port</i>	Selects the interface to configure.
Step 2	Switch(config-if)# [no] power inline consumption <i>milli-watts</i>	Sets the PoE consumption (in milliwatts) of the powered device connected to a specific interface. The power consumption can range from 4000 to 15,400. To reenable the automatic adjustment of consumption, either use the no keyword or specify 15,400 milliwatts.
Step 3	Switch(config-if)# end	Exits configuration mode.
Step 4	Switch# show power inline consumption { fastethernet gigabitethernet } <i>slot/port</i>	Displays the PoE consumption for the interface.

This example shows how to set the PoE consumption to 5000 milliwatts for interface gi 7/1 regardless what is mandated by the 802.3af class of the discovered device, or by any CDP packet received from the powered device. This example also verifies the PoE consumption on interface gi 7/1.

The following output displays the initial power consumption of the interface:

```
Switch# show power inline gi 7/1
Available:627(w)  Used:267(w)  Remaining:360(w)

Interface Admin  Oper          Power(Watts)      Device           Class
          From PS    To Device
-----
Gi7/1      auto    on           7.9              7.0             IP Phone 7941    3

Interface  AdminPowerMax  AdminConsumption
          (Watts)      (Watts)
-----
Gi7/1                        15.4              15.4

Switch# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# int gi 7/1
Switch(config-if)# power inline consumption 5000
Switch(config-if)# exit
Switch(config)# exit
```

The following output displays the power consumption after entering the **power inline consumption** command on the interface:

```
Switch# show power inline gi 7/1
Available:627(w)  Used:265(w)  Remaining:362(w)

Interface Admin  Oper          Power(Watts)      Device           Class
          From PS    To Device
-----
Gi7/1      auto    on           5.6              5.0             Ieee PD          3
```

Interface	AdminPowerMax (Watts)	AdminConsumption (Watts)
-----	-----	-----
Gi7/1	15.4	5.0

Displaying the Operational Status for an Interface

Each interface has an operational status which reflects the PoE status for an interface. The operational status for an interface is defined as one of the following:

- on—Power is supplied by the port.
- off—Power is not supplied by the port. If a powered device is connected to an interface with external power, the switch does not recognize the powered device. The “Device” column in the **show power inline** command displays as n/a.
- Power-deny—The supervisor engine does not have enough power to allocate to the port, or the power that is configured for the port is less than the power required by the port; power is not being supplied by the port.
- err-disable—The port is unable to provide power to the connected device that is configured in static mode.
- faulty—The port failed diagnostics tests.

To view the operational status for an interface, use the **show power inline** command.

This example shows how to display the operational status for all interfaces on module 3:

```
Switch# show power inline module 3
Available:677(w)  Used:117(w)  Remaining:560(w)
```

Interface	Admin	Oper	Power(Watts)		Device	Class
			From PS	To Device		
-----	-----	-----	-----	-----	-----	-----
Fa3/1	auto	on	17.3	15.4	Ieee PD	0
Fa3/2	auto	on	4.5	4.0	Ieee PD	1
Fa3/3	auto	on	7.1	6.3	Cisco IP Phone 7960	0
Fa3/4	auto	on	7.1	6.3	Cisco IP Phone 7960	n/a
Fa3/5	auto	on	17.3	15.4	Ieee PD	0
Fa3/6	auto	on	17.3	15.4	Ieee PD	0
Fa3/7	auto	on	4.5	4.0	Ieee PD	1
Fa3/8	auto	on	7.9	7.0	Ieee PD	2
Fa3/9	auto	on	17.3	15.4	Ieee PD	3
Fa3/10	auto	on	17.3	15.4	Ieee PD	4
Fa3/11	auto	off	0	0	n/a	n/a
Fa3/12	auto	off	0	0	n/a	n/a
Fa3/13	auto	off	0	0	n/a	n/a
Fa3/14	auto	off	0	0	n/a	n/a
Fa3/15	auto	off	0	0	n/a	n/a
Fa3/16	auto	off	0	0	n/a	n/a
Fa3/17	auto	off	0	0	n/a	n/a
Fa3/18	auto	off	0	0	n/a	n/a
-----	-----	-----	-----	-----	-----	-----
Totals:		10 on	117.5	104.6		

```
Switch#
```

This example shows how to display the operational status for Fast Ethernet interface 4/1:

```
Switch# show power inline fa4/1
Available:677(w)  Used:11(w)  Remaining:666(w)
```

Interface	Admin	Oper	Power(Watts)		Device	Class
			From PS	To Device		
Fa4/1	auto	on	11.2	10.0	Ieee PD	0

Interface	AdminPowerMax (Watts)	AdminConsumption (Watts)
Fa4/1	15.4	10.0

```
Switch#
```

Displaying all PoE Detection and Removal Events

Starting with Cisco IOS Release 15.0(2)SG2/XE 3.2.2SG, a Catalyst 4500 series switch can display all PoE detection and removal events.

To enable PoE event logging, you use the **power inline logging global** command:

```
Switch# conf terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# power inline logging global
Switch(config)# int gigabitEthernet 5/5
Switch(config-if)# shut
Switch(config-if)#
*Oct 17 12:02:48.407: %ILPOWER-5-IEEE_DISCONNECT: Interface Gi5/5: PD removed
Switch(config-if)# no shut
Switch(config-if)#
*Oct 17 12:02:54.915: %ILPOWER-7-DETECT: Interface Gi5/5: Power Device detected: IEEE PD
```

Displaying the PoE Consumed by a Module

A Catalyst 4500 series switch can measure the actual PoE consumption for an 802.3af-compliant PoE module. You can observe this consumption by using **show power module** and **show power detail** commands. For all PoE calculations, presume that the PoE consumption on the WS-X4148-RJ45V module equals its administrative PoE.

The 802.3af-compliant PoE modules can consume up to 20 W of PoE to power FPGAs and other hardware components on the module. To ensure that the system has sufficient power for the PDs connected to the switch, add at least 20 W to your PoE requirements for each 802.3af-compliant PoE module.

The following example uses the **show power module** command to display the PoE consumption for an 802.3af-compliant module:

Switch# **show power module**

Watts Used of System Power (12V)						
Mod	Model	currently	out of reset	in reset		
1	WS-X4013+TS	330	330	330		
2	WS-X4548-GB-RJ45V	60	60	20		
3	WS-X4548-GB-RJ45V	60	60	20		
--	Fan Tray	30	--	--		
Total		480	450	370		

Watts used of Chassis Inline Power (-50V)						
Mod	Model	PS	Device	PS	Device	Efficiency
2	WS-X4548-GB-RJ45V	138	123	73	65	89
3	WS-X4548-GB-RJ45V	0	0	22	20	89
Total		138	123	95	85	

Watts used of Module Inline Power (12V -> -50V)						
Mod	Model	PS	Device	PS	Device	Efficiency
1	WS-X4013+TS	128	128	63	63	100

Switch#

The Inline Power Oper column displays the amount of PoE consumed by the powered devices that are attached to the module, in addition to the PoE consumed by the FPGAs and other hardware components on the module.

The Inline Power Admin column displays only the amount of PoE allocated by the powered devices attached to the module.



Note

The operating PoE consumption for an 802.3af-compliant module can be non-zero, even when no powered devices are attached to the module, because of the PoE consumed by FPGAs and other hardware components on the module. In addition, the operating PoE can vary because of fluctuations in the PoE consumed by the hardware components.

The following example uses the **show power detail** and **show power inline** commands to display the PoE consumption for an 802.3af-compliant module:

Switch# **show power detail**

Power Supply	Model No	Type	Status	Fan Sensor	Inline Status
PS1	PWR-C45-1300ACV	AC 1300W	good	good	good
PS2	none	--	--	--	--

Power supplies needed by system : 1

Power supplies currently available : 1

■ Displaying the PoE Consumed by a Module

```

Power Summary
(in Watts)
-----
System Power (12V)      480      1000
Inline Power (-50V)     138      800
Backplane Power (3.3V)   0        0
-----
Total                   618 (not to exceed Total Maximum Available = 1300)

```

```

Module Inline Power Summary (Watts)
(12V -> -48V on board conversion)
-----

```

```

Mod      Used      Maximum
-----
1        128      158
-----

```

```

Mod      Model      Watts Used of System Power (12V)
              currently out of reset in reset
-----
1      WS-X4013+TS      330      330      330
2      WS-X4548-GB-RJ45V  60      60      20
3      WS-X4548-GB-RJ45V  60      60      20
--      Fan Tray        30      --      --
-----
Total      480      450      370

```

```

Watts used of Chassis Inline Power (-50V)
Mod      Model      Inline Power Admin PS Device      Inline Power Oper PS Device      Efficiency
-----
2      WS-X4548-GB-RJ45V  138      123      73      65      89
3      WS-X4548-GB-RJ45V   0        0       22      20      89
-----
Total      138      123      95      85

```

```

Watts used of Module Inline Power (12V -> -50V)
Mod      Model      Inline Power Admin PS Device      Inline Power Oper PS Device      Efficiency
-----
1      WS-X4013+TS      128      128      64      64      100
-----

```

```
Switch# show power inline g1/1
```

```
Module 1 Inline Power Supply: Available:158(w) Used:128(w) Remaining:30(w)
```

```

Interface Admin Oper      Power(Watts)      Device      Class
              From PS      To Device
-----
Gi1/1      auto   on        10.3      10.3      CNU Platform      3

Interface      AdminPowerMax      AdminConsumption
              (Watts)      (Watts)
-----
Gi1/1              15.4      15.4

```



```
switch# show power inline g2/1
Chassis Inline Power Supply: Available:800(w) Used:138(w) Remaining:662(w)
```

Interface	Admin	Oper	Power(Watts)		Device	Class
			From PS	To Device		
Gi2/1	auto	on	11.5	10.2	CNU Platform	n/a

Interface	AdminPowerMax (Watts)	AdminConsumption (Watts)
Gi2/1	15.4	15.4

```
Switch# show power inline module 1
Module 1 Inline Power Supply: Available:158(w) Used:128(w) Remaining:30(w)
```

Interface	Admin	Oper	Power(Watts)		Device	Class
			From PS	To Device		
Gi1/1	auto	on	10.3	10.3	CNU Platform	3
Gi1/2	auto	on	10.3	10.3	CNU Platform	3
Gi1/3	auto	on	10.3	10.3	CNU Platform	3
Gi1/4	auto	on	10.3	10.3	CNU Platform	3
Gi1/5	auto	on	10.3	10.3	CNU Platform	3
Gi1/6	auto	on	10.3	10.3	CNU Platform	3
Gi1/7	auto	on	10.3	10.3	CNU Platform	3
Gi1/8	auto	on	10.3	10.3	CNU Platform	3
Gi1/9	auto	on	10.3	10.3	CNU Platform	3
Gi1/10	auto	on	15.4	15.4	Cisco/Ieee PD	3
Gi1/11	auto	on	10.3	10.3	CNU Platform	3
Gi1/12	auto	on	10.3	10.3	CNU Platform	3
Totals:	12	on	128.2	128.2		

```
switch# show power inline module 2
Chassis Inline Power Supply: Available:800(w) Used:138(w) Remaining:662(w)
```

Interface	Admin	Oper	Power(Watts)		Device	Class
			From PS	To Device		
Gi2/1	auto	on	11.5	10.2	CNU Platform	n/a
Gi2/2	auto	on	11.5	10.2	CNU Platform	n/a
Gi2/3	auto	on	11.5	10.2	CNU Platform	n/a
Gi2/4	auto	on	11.5	10.2	CNU Platform	n/a
Gi2/5	auto	off	0.0	0.0	n/a	n/a
Gi2/6	auto	off	0.0	0.0	n/a	n/a
Gi2/7	auto	off	0.0	0.0	n/a	n/a
Gi2/8	auto	off	0.0	0.0	n/a	n/a
Gi2/9	auto	on	11.5	10.2	CNU Platform	3
Gi2/10	auto	on	11.5	10.2	CNU Platform	n/a
Gi2/11	auto	on	11.5	10.2	CNU Platform	n/a
Gi2/12	auto	on	11.5	10.2	CNU Platform	n/a
Gi2/13	auto	on	11.5	10.2	CNU Platform	3
Gi2/14	auto	on	11.5	10.2	CNU Platform	3
Gi2/15	auto	on	11.5	10.2	CNU Platform	3

Gi2/16	auto	on	11.5	10.2	CNU Platform	3
Gi2/17	auto	off	0.0	0.0	n/a	n/a
Gi2/18	auto	off	0.0	0.0	n/a	n/a
Interface	Admin	Oper	Power (Watts)		Device	Class
			From PS	To Device		

Gi2/19	auto	off	0.0	0.0	n/a	n/a
Gi2/20	auto	off	0.0	0.0	n/a	n/a
Gi2/21	auto	off	0.0	0.0	n/a	n/a
Gi2/22	auto	off	0.0	0.0	n/a	n/a
Gi2/23	auto	off	0.0	0.0	n/a	n/a
Gi2/24	auto	off	0.0	0.0	n/a	n/a
Gi2/25	auto	off	0.0	0.0	n/a	n/a
Gi2/26	auto	off	0.0	0.0	n/a	n/a
Gi2/27	auto	off	0.0	0.0	n/a	n/a
Gi2/28	auto	off	0.0	0.0	n/a	n/a
Gi2/29	auto	off	0.0	0.0	n/a	n/a
Gi2/30	auto	off	0.0	0.0	n/a	n/a
Gi2/31	auto	off	0.0	0.0	n/a	n/a
Gi2/32	auto	off	0.0	0.0	n/a	n/a
Gi2/33	auto	off	0.0	0.0	n/a	n/a
Gi2/34	auto	off	0.0	0.0	n/a	n/a
Gi2/35	auto	off	0.0	0.0	n/a	n/a
Gi2/36	auto	off	0.0	0.0	n/a	n/a
Gi2/37	auto	off	0.0	0.0	n/a	n/a
Gi2/38	auto	off	0.0	0.0	n/a	n/a
Gi2/39	auto	off	0.0	0.0	n/a	n/a
Gi2/40	auto	off	0.0	0.0	n/a	n/a
Interface	Admin	Oper	Power (Watts)		Device	Class
			From PS	To Device		

Gi2/41	auto	off	0.0	0.0	n/a	n/a
Gi2/42	auto	off	0.0	0.0	n/a	n/a
Gi2/43	auto	off	0.0	0.0	n/a	n/a
Gi2/44	auto	off	0.0	0.0	n/a	n/a
Gi2/45	auto	off	0.0	0.0	n/a	n/a
Gi2/46	auto	off	0.0	0.0	n/a	n/a
Gi2/47	auto	off	0.0	0.0	n/a	n/a
Gi2/48	auto	off	0.0	0.0	n/a	n/a

Totals:		12 on	138.2	123.0		
Switch#						

PoE Policing and Monitoring



Note

This functionality is supported on the WS-X4548-RJ45V+, WS-X4648-RJ45V-E, and WS-X4648-RJ45V+E line cards.

PoE policing protects a switch from faulty inline powered devices that may draw more current than they were designed for. When a device is connected to a port, a line card detects the type of device connected and allocates the appropriate amount of power. It sets a PoE policing threshold to a value 5 percent greater than the allocated power. If the device consumes more power than specified by the policing threshold for a more than 1 second, the port shuts down. Depending on the policing action configured, the port may then be error-disabled, or a message might be logged to the console and the port restarted.

PoE monitoring lets you display the true power consumption of inline powered devices attached to the switch, allowing you determine your actual power consumption.

This section includes these topics:

- [PoE Policing Modes, page 15-13](#)
- [Configuring Power Policing on an Interface, page 15-13](#)
- [Displaying Power Policing on an Interface, page 15-14](#)
- [Configuring Errdisable Recovery, page 15-15](#)

PoE Policing Modes

PoE policing comprises two modes, which determine the action to take on the interface after a port shuts down because of an inline-power policing violation:

- **Logging** — An error message is logged to the console and the interface restarts; the device powers up.
- **Errdisable** (Default) — In addition to logging an error message to the console, the interface is placed in an errdisable state so that the device attached to the port does not receive inline-power until you restart the port or configure an errdisable autorecovery mechanism.



Note

After an inline-power policing violation occurs and the port shuts down, PoE policing automatically turns on again when the port restarts. If the connected device exceeds its allocated power again, the port once again shuts down.

Configuring Power Policing on an Interface

The default policing levels are determined by the discovery and power allocation methods (listed in order of priority):

- Configured consumption values, in case any exist
- CDP allocated values (for Cisco devices using CDP)
- Allocated power from IEEE discovery (for devices using this mechanism)

To activate default PoE policing, enter the following:

```
Switch# conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# int g2/1
Switch(config-if)# power inline police
Switch(config-if)# end
Switch# show power inline police g2/1
Available:800(w) Used:32(w) Remaining:768(w)
```

Interface	Admin State	Oper State	Admin Police	Oper Police	Cutoff Power	Oper Power
-----	-----	-----	-----	-----	-----	-----
Gi2/1	auto	on	errdisable	ok	17.2	16.7

The default action for power policing is to set the port to errdisable; the **power inline police** command is equivalent to the **power inline police action errdisable** command, as the above example illustrates. The following example illustrates how to configure the logging policing action:

```
Switch# conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# int g2/1
Switch(config-if)# power inline police action log
Switch(config-if)# end
Switch# show power inline police g2/1
Available:800(w) Used:32(w) Remaining:768(w)
```

Interface	Admin State	Oper State	Admin Police	Oper Police	Cutoff Power	Oper Power
-----	-----	-----	-----	-----	-----	-----
Gi2/1	auto	on	log	ok	17.2	16.7

When a PD consumes more than its allocated power, the port shuts down and a warning message similar to the following appears on the console.

For the WS-X4648-GB-RJ45V and WS-X4648-GB-RJ45V+:

```
*Sep 12 09:15:28.583: %C4K_ETHPORTMAN-3-INLINEPOWEROVERDRAWN: Inline powered device
connected on port Gi3/25 exceeded its policed threshold.
```

For the WS-X4548-RJ45V+:

```
*Sep 26 09:23:21.355: %C4K_SWITCHMANAGER-3-INLINEPOWEROVERDRAWN: Inline powered device
connected on port Gi2/1 exceeded its policed threshold.
```

For actions of Log type, the port restarts itself and the device reboots. In contrast, when the action is to set the port in an errdisable state, a log message similar to the following appears:

```
*Sep 26 09:30:20.463: %PM-4-ERR_DISABLE: inline-power error detected on Gi2/1, putting
Gi2/1 in err-disable state
```

```
Switch# show power inline police g2/1
Available:800(w) Used:16(w) Remaining:784(w)
```

Interface	Admin State	Oper State	Admin Police	Oper Police	Cutoff Power	Oper Power
-----	-----	-----	-----	-----	-----	-----
Gi2/1	auto	errdisable	errdisable	overdrawn	0.0	0.0

Displaying Power Policing on an Interface

You can display power policing on an interface, on a module, or for all the PoE-capable line cards in a chassis.

The following example shows output for the **show power inline police** command:

```
Switch# show power inline police
Available:623(w) Used:6(w) Remaining:617(w)
```

Interface	Admin State	Oper State	Admin Police	Oper Police	Cutoff Power	Oper Power
-----	-----	-----	-----	-----	-----	-----
Gi2/1	auto	off	none	n/a	n/a	0.0
Gi2/2	auto	on	none	n/a	n/a	16.7
Gi2/3	auto	off	errdisable	n/a	0.0	0.0
Gi2/4	auto	on	errdisable	ok	16.6	11.4
Gi2/5	auto	on	log	ok	16.6	11.2
Gi2/6	auto	on	errdisable	overdrawn	0.0	0.0

The following table lists the interface and the status.

Interface Configuration	State
Gi2/1	No PD connected, no policing configured.
Gi2/2	PD connected, no policing configured.
Gi2/3	No PD connected, policing configured (is enabled when PD is connected). Policing action is errdisable.
Gi2/4	PD connected, policing configured. Configured policing action is errdisable. Port is currently operating within policing limits.
Gi2/5	PD connected, policing configured. Configured policing action is log. Port is currently operating within policing limits.
Gi2/6	PD connected, policing configured. Configured policing action is errdisable. Port is currently in errdisable state as it has overdrawn its policed power level.

If you enter the **show power inline** command at the global level (**show power inline police**), the last line of the output under the Oper Power field displays the total of true inline power consumption of all devices connected to the switch.

Configuring Errdisable Recovery

By default, errdisable auto recovery for inline-power is disabled; when an interface is placed in an errdisable state because of an inline-power policing violation, it remains in that state. You must enter **shut** and then **no shut** on the affected interface to revive it.

The errdisable autorecovery mechanism allows you to configure a timer for errdisable recovery so that when an interface enters errdisable state (after the timer expires), the interface returns from the errdisable state.

errdisable detection

By default, errdisable detection for inline-power is enabled, as the following example illustrates:

```
Switch# show errdisable detect
ErrDisable Reason    Detection    Mode
-----
inline-power         Enabled      port
```



Note

If detection is disabled (through the **errdisable detect cause inline-power** command), the port is not placed in errdisable state when it exceeds its power policing threshold.

errdisable recovery

By default, errdisable recovery for inline-power is disabled. To enable errdisable recovery, enter the **errdisable detect cause line-power** command:

```
Switch# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# errdisable detect cause inline-power
Switch(config)# end
```

```
Switch# show errdisable recovery
ErrDisable Reason          Timer Status
-----
inline-power              Enabled
```

The WS-X4648-RJ45V-E, WS-X4648-RJ45V+E, and WS-X4548-RJ45V+ switching modules support IEEE 802.3af PoE as well as the Cisco proprietary Inline Power standard. With Cisco IOS Release 12.2(44)SG, the WS-X4648-RJ45V+E line card can also support the IEEE 802.3at standard with up to 30 W available per-port. The WS-X4648-RJ45V-E line card also supports up to 20 W. The WS-X4548-RJ45V+ switching module is supported with Cisco IOS Release 12.2(50)SG and can provide up to 30 W of inline power per-port.

For these switching modules, the valid milliwatt ranges for the **power inline** command have been increased appropriately for the module, as the following table illustrates:

Line card	Standard	Max Power/Port	Cisco IOS Release
WS-X4648-RJ45V-E	IEEE 802.3af	20 W	12.2(44)SG
	IEEE 802.3at		
WS-X4648-RJ45V+E	IEEE 802.3af	30 W	12.2(44)SG
	IEEE 802.3at		
WS-X4548-RJ45V+	IEEE 802.3af	30 W	12.2(50)SG
	IEEE 802.3at		

The default power inline configurations usually are sufficient; no additional configuration is required even for high power-consumption Cisco powered devices (for example, a Cisco AP1250 Wireless Access Point). When a high-power consumption device is attached to a port on a WS-X4648-RJ45V-E or WS-X4648-RJ45V+E line card, the switch and device negotiate power using CDP packets to automatically determine the extended amount of power needed by the device.

Depending on the deployment requirements and design, you specify a specific configuration with the **power inline** command.

The following example shows how to pre-allocate PoE allocation to 16500 mW for Gi 2/1, regardless of what is mandated either by the 802.3af class of the discovered device or by any CDP packet that is received from the powered device:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface gigabitethernet 2/1
Switch(config-if)# power inline static max 16500
Switch(config-if)# end
Switch#
```

Configuring Universal PoE



Note

This feature is only available on Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, and 7-E.

Although IEEE 802.at only provides for power up to 30W per port, the WS-X4748-UPOE+E module can provide up to 60W using the spare pair of an RJ45 cable (wires 4,5,7,8) with the signal pair (wires 1,2,3,6). Power on the spare pair is enabled when the switch port and end-device mutually identify

themselves as Universal PoE (UPOE) capable using CDP or LLDP and the end-device requests for power on the spare pair to be enabled. When the spare pair is powered, the end-device can negotiate up to 60W power from the switch using CDP or LLDP.

If the end-device is PoE capable on both signal and spare pairs but does not support the CDP or LLDP extensions required for UPOE, then the following configuration automatically enables power on both signal and spare pairs from the switch port:

	Command	Purpose
Step 1	Switch # interface terminal	Changes mode to global configuration.
Step 2	Switch(config)# interface { fastethernet gigabitethernet } <i>slot/port</i>	Selects the interface to configure.
Step 3	Switch(config-if)# power inline four-pair forced	To automatically enables power on both signal and spare pairs from a switch port.
Step 4	Switch(config-if)# shutdown	Shuts down the port.
Step 5	Switch(config-if)# no	Boots the port.
Step 6	Switch(config-if)# end	Exits configuration mode.
Step 7	Switch# show platform software interface { fastethernet gigabitethernet } <i>slot/port</i> status	Displays EEE status.

The following example shows how to automatically enable power on both signal and spare pairs from switch port gigabit ethernet 2/1:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface gigabitethernet 2/1
Switch(config-if)# power inline four-pair forced
Switch(config-if)# shutdown
Switch(config-if)# no shutdown
Switch(config-if)# end
Switch#
```

Do not enter this command if the end-device is incapable of sourcing inline power on the spare pair or if the end-device supports the CDP or LLDP extensions for UPOE.

Additional References for Power over Ethernet

Related Documents

Related Topic	Document Title
Catalyst 4500 switch commands	<i>Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch</i>

MIBs

MIB	MIBs Link
CISCO-POWER-ETHERNET-EXT-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Power over Ethernet

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

**Note**

Table 15-2 lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 15-2 *Feature Information for Power over Ethernet*

Feature Name	Releases	Feature Information
Power over Ethernet	Cisco IOS Release 12.2(25)EW	This feature was introduced.
	Cisco IOS Release 15.2(6)E2	MIB support for CISCO-POWER-ETHERNET-EXT-MIB was introduced.



Configuring Catalyst 4500 Series Switches with Cisco Network Assistant

This chapter describes how to install Network Assistant on the workstation and configure Catalyst 4500 series switch to communicate with Network Assistant. It also describes how to create communities and clusters, which are two technologies used by Network Assistant to manage a group of network devices, including the Catalyst 4500 series switch.

This chapter contains these topics:

- [About Network Assistant, page 16-2](#)
- [Network Assistant-Related Parameters and Their Defaults, page 16-2](#)
- [Network Assistant CLI Commands, page 16-3](#)
- [Configuring Your Switch for Network Assistant, page 16-4](#)
- [Managing a Network Using Community, page 16-6](#)
- [Converting a Cluster into a Community, page 16-10](#)
- [Managing a Network Using Cluster, page 16-11](#)
- [Configuring Network Assistant in Community or Cluster Mode, page 16-13](#)



Note

The Network Assistant is not bundled with an online software image on Cisco.com. You can download the Network Assistant at this location:

<http://www.cisco.com/en/US/products/ps5931/index.html>

For information on software and hardware requirements, installing Network Assistant, launching Network Assistant, and connecting Network Assistant to a device refer to *Getting Started with Cisco Network Assistant*, available at the URL:

http://www.cisco.com/en/US/products/ps5931/prod_installation_guides_list.html.

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About Network Assistant

Network Assistant is a free network management tool that allows you to configure and manage Catalyst 4500 series switches using a graphical user interface (GUI). Network Assistant works in both secure and unsecure environments. Network Assistant manages standalone devices or groups of devices or switches (in communities or clusters) from anywhere in your intranet. Using Network Assistant, you can perform multiple configuration tasks without having to remember commands.

Community Overview

A *community* is a device group that can contain up to 20 connected network devices. Network Assistant uses the Cisco Discovery Protocol (CDP) automatic discovery capability to find eligible network devices and to add them to a community. When a network device is added to a community, it becomes a member device. Network Assistant manages, configures, and monitors each member on an individual basis; therefore, each member must have an IP address assigned to it.

When you use communities, you need to have an HTTP server, and you need to configure an IP address on each switch.

Clustering Overview

A *switch cluster* is a set of up to 16 connected, cluster-capable Catalyst switches that are managed as a single entity. The switches in the cluster use the switch clustering technology so that you can configure and troubleshoot a group of different Catalyst 4500 series switch platforms through a single IP address.

Using switch clusters simplifies the management of multiple switches, regardless of their physical location and platform families.



Note

By default, Network Assistant in clustering mode discovers up to seven hops away.

In a switch cluster, one switch must be the *cluster commander switch*, and up to 15 other switches can be *cluster member switches*. The total number of switches in a cluster cannot exceed 16 switches. The cluster command switch is the single point of access used to configure, manage, and monitor the cluster member switches. Cluster members can belong to only one cluster at a time.



Note

Always choose Catalyst 4500 series switches as the cluster command switch.

Network Assistant-Related Parameters and Their Defaults

Table 16-1 lists the Network Assistant-related configuration parameters on a Catalyst 4500 series switch.

Table 16-1 Network Assistant-Related Configuration on Catalyst 4500 Series Switches

Parameters	Default Value	Recommended Value
Authentication	Disabled	Optional
IP address	Depends on community or discovery option ¹	User selectable
IP HTTP port number	80	Optional ²
IP HTTPS port number	443	Optional ³
IP HTTP server	Disabled	Enabled ⁴
Cluster run	Disabled	Enabled ⁵

1. You need to set an IP address in each switch for community device discovery and for the cluster commander.
2. Port number on the Network Assistant and the Catalyst 4500 series switch must match.
3. You can only change this value for a cluster of devices. Port number on the Network Assistant and on the Catalyst 4500 series switch must match. Value can be changed to any non-default number above 1024.
4. Required for Network Assistant to access the device.
5. Enabled only if you want to manage a cluster of devices.

Network Assistant CLI Commands

Table 16-2 describes the Network Assistant-related CLI commands.

Table 16-2 CLI Commands

Command	Functions
<code>[no] cluster enable</code>	Names the cluster.
<code>[no] cluster run</code>	Enables clustering. Note This command is used strictly for clustering.
<code>[no] ip http server</code>	Configures the HTTP on a switch.
<code>[no] ip http port <i>port_number</i></code>	Configures the HTTP port.
<code>[no] ip domain-name <i>domain_name</i></code>	Configures the domain on the switch.
<code>[no] ip http secure-server</code>	Configures and enable HTTPS on a switch.
<code>[no] ip http secure-port <i>port_number</i></code>	Configures the HTTPS port.
<code>[no] ip http max-connections <i>connection_number</i></code>	Configures the maximum concurrent connections to the HTTP server.
<code>[no] ip http timeout-policy <i>idle idle_time life life_time requests requests</i></code>	Configures the HTTPS port. A idle value of 180 seconds is recommended. A life value of 180 seconds is recommended. The recommended maximum number of requests allowed is 25.

Table 16-2 CLI Commands (continued)

Command	Functions
<code>line vty</code>	Configures additional VTYs for use by Cisco Network Assistant.
<code>show version</code>	Displays the Cisco IOS release.
<code>show running-config</code>	Displays the switch configuration.
<code>vtp domain</code>	Creates a VTP domain to manage VLANs.
<code>vtp mode</code>	Sets the behavior for VTP management of the VLANs.

Configuring Your Switch for Network Assistant

This section includes the following topics:

- [\(Minimum\) Required Configuration, page 16-4](#)
- [\(Additional\) Configuration Required to Use Community, page 16-5](#)
- [\(Additional\) Configuration Required to Use Clustering, page 16-5](#)

(Minimum) Required Configuration

If you use the default configuration, access the Catalyst 4500 series switch and enter the **ip http server** (for HTTP) or **ip http secure-server** (for HTTPS) global configuration command.

To configure the Catalyst 4500 series switch, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# ip http server or Switch(config)# ip domain-name <i>domain_name</i>	(HTTP only) Enables the HTTP server on the switch. By default, the HTTP server is disabled. Enables the domain name on the switch to configure HTTPS.
Step 3	Switch(config)# ip http secure-server	Enables the HTTPS server on the switch. By default, the HTTPS server is disabled.
Step 4	Switch(config)# ip http max-connections <i>connection_number</i>	Configures the maximum concurrent connections to the HTTP server. <i>A connection_number of 16 is recommended.</i>

	Command	Purpose
Step 5	Switch(config)# ip http timeout-policy idle <i>idle_time life life_time requests requests</i>	Configures the HTTPS port. The idle keyword specifies the maximum amount of time a connection can stay idle. A idle value of 180 seconds is recommended. The life keyword specifies the maximum amount of time a connection can stay open since it was established. A life value of 180 seconds is recommended. The requests keyword specifies the maximum amount of requests on a connection. The recommended maximum number of requests allowed is 25.
Step 6	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 7	Switch# show running-config	Verifies the configuration.

**Note**

If you have enabled clustering, disable clustering before configuring a community (see [Table 16-2](#)).

(Additional) Configuration Required to Use Community

If you plan to use community, define an IP address on each switch.



To configure a switch to use community, perform this task:

	Command	Purpose
Step 1	Switch# configuration terminal	Enters global configuration mode.
Step 2	Switch(config)# interface { vlan <i>vlan_ID</i> { fastethernet gigabitethernet } <i>slot/interface</i> Port-channel <i>number</i> }	Selects an interface.
Step 3	Switch(config-if)# ip address ip_address <i>address_mask</i>	(Optional) Assigns an IP address to the Catalyst 4500 series. Note This step is mandatory if the switch is part of community or is a cluster command switch. This step is optional if the switch is a cluster member candidate.
Step 4	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 5	Switch# show running-config	Verifies the configuration.

(Additional) Configuration Required to Use Clustering

If you plan to use clustering, enter the **cluster run** global configuration command on each device and enter the **ip address** interface configuration command on the cluster commander.

To configure a switch to use clustering, perform this task:

	Command	Purpose
Step 1	Switch# configuration terminal	Enters global configuration mode.
Step 2	Switch(config)# cluster run	Enables clustering.
		 Note Enable clustering on all switches that are part of the potential cluster.
Step 3	Switch(config)# cluster enable	Names the cluster.
Step 4	Switch(config)# interface { vlan <i>vlan_ID</i> { fastethernet gigabitethernet } <i>slot/interface</i> Port-channel <i>number</i> }	Selects an interface.
Step 5	Switch(config-if)# ip address <i>ip_address address_mask</i>	(Optional) Assigns an IP address to the Catalyst 4500 series switch cluster master.
		 Note This step is mandatory if the switch is part of a community or is a cluster command switch. This step is optional if the switch is a cluster member candidate.
Step 6	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 7	Switch# show running-config	Verifies the configuration.

Managing a Network Using Community

This section describes how to use communities to manage devices (including Catalyst 4500 series switches, routers, access points, and PIX firewalls) using the Network Assistant application.



Note

Access points have been eliminated from the device limits. There is no current limit for the number of access points that can be managed by CNA.



Note

The **Add to Community** dialog box displays any number of devices, but can only select 20 devices. If you try to add a twenty-first device, the dialog box displays the twenty-first device and prompts you to select the unwanted device.

Some devices like routers and access points do not support clustering. CNA 2.0 introduces a logical grouping for such devices, which are called *communities*. This grouping supports the auto discovery protocol using CDP. Any device that cannot support CDP can be added as a managed device manually by specifying the IP address.



Note

For complete procedures for using Network Assistant to configure switch communities, refer to *Getting Started with Cisco Network Assistant*, available at:

http://www.cisco.com/en/US/products/ps5931/prod_installation_guides_list.html.

This section describes the guidelines and requirements you should understand before you create a community. This section contains the following topics:

- [Candidate and Member Requirements, page 16-7](#)
- [Automatic Discovery of Candidates and Members, page 16-7](#)
- [Community Names, page 16-8](#)
- [Hostnames, page 16-8](#)
- [Passwords, page 16-8](#)
- [Access Modes in Network Assistant, page 16-9](#)
- [Community Information, page 16-9](#)
- [Adding Devices, page 16-9](#)

Candidate and Member Requirements

Candidates are network devices that have IP addresses but are not part of a community. *Members* are network devices that are currently part of a community.

To join a community, a candidate must meet these requirements:

- An IP address has been obtained.
- Cisco Discovery Protocol (CDP) version 2 is enabled (the default) (if you want the device to be auto-discovered).
- HTTP (or HTTPS) is enabled.



Note A cluster member can be added to a community, but the reverse is not possible.



Note If a cluster commander is added to a community, the other member devices of the cluster are not added automatically. The cluster members must be added to the community on an individual basis in order to be managed.

Automatic Discovery of Candidates and Members

Network Assistant forms a community using CDP to locate or discover all the available devices in the network. Beginning with the IP address for a starting device and the port numbers for HTTP (or HTTPS) protocols, Network Assistant uses CDP to compile a list of community candidates that neighbor the starting device. Network Assistant can discover candidate and member devices across multiple networks and VLANs as long as they have valid IP addresses.



Note By default, Network Assistant in community mode discovers up to four hops away.

See the [“Candidate and Member Requirements” section on page 16-7](#) for a list of requirements that network devices must meet in order to be discovered.

**Note**

Do not disable CDP on candidates, members, or on any network devices that you might want Network Assistant to discover.

**Note**

PIX firewalls do not support the CDP, so they are not automatically shown as neighbors in the Topology view. They are shown only after you add them to a community with the Create Community or Modify Community window. To see a PIX firewall link to another community member, you must add the link manually by selecting ADD Link in the Topology popup menu.

You can edit the list of discovered devices to fit your needs and add them to the community. As each device is added to the community, its neighbors are discovered and added to the list of candidate devices. If Network Assistant fails to discover a device, you can add it manually through the IP management IP address.

Community Names

When you apply the community configuration information to the list of member devices, Network Assistant requests that you enter a name (or IP address) for the community. You need to assign a name to the community before you can manage it. Network Assistant saves the name to your PC.

The community name can consist of the characters 0 through 9, a through z and A through Z, with spaces allowed between the characters.

**Note**

You can connect to a cluster only through an IP address. When you select a name the name is always for the community.

Hostnames

You do not need to assign a hostname to a starting device or a community member. However, we recommend that you do assign a hostname because Network Assistant does not assign one by default. If a discovered device does have a hostname, Network Assistant saves it to your PC as identifying information for that device along with its IP address, communication protocol, and designated protocol port.

Passwords

Although you do not need to assign a password to a device if it will become a community member, we recommend that you do so.

Community members can have different passwords.

Communication Protocols

Network Assistant uses the HTTP or HTTPS protocols to communicate with network devices. It attempts communication with HTTP or HTTPS when using CDP to discover candidate devices.

Access Modes in Network Assistant

When Network Assistant is connected to a community or cluster, two access modes are available: read-write and read-only, depending on the password.

Community Information

Network Assistant saves all community configuration information and individual device information such as IP address, hostname, and communication protocol to your local PC. When Network Assistant connects to a community, it uses the locally saved data to rediscover the member devices.

If you attempt to use a different PC to manage an existing community, the member device information is not available. You need to create the community again and add the same member devices.

Adding Devices

You can add members to a community using these methods:

- Use the Devices Found window on Network Assistant to add devices that you discovered to a new community.
 - In the Devices Found window, select the candidate devices that you want to add.
To add more than one candidate, press **Ctrl** and make your choices, or press **Shift** and choose the first and last device in a range.
 - Click **Add**.
- Use the Modify Community window to add devices to an existing community.
 - Choose **Application > Communities** to open the Communities window.
 - In the Communities window, select the name of the community to which you want to add a device, and click **Modify**.
 - To add a single device manually, enter the IP address for the desired device in the Modify Community window, and click **Add**.
 - To discover candidate devices, enter the IP address for the starting device, and click **Discover**.
 - Select a candidate device from the list, click **Add**, and click **OK**.
 - To add more than one candidate, press **Ctrl** and make your choices, or press **Shift** and choose the first and last device in a range.
- Add a device using the Topology view.
 - If the Topology view is not displayed, choose **View window> Topology** from the feature bar.
 - Right-click a candidate icon, and select **Add to Community**.

Candidates are cyan; members are green. To add more than one candidate, press **Ctrl** and left-click the candidates that you want to add.

When a community has 20 members, the **Add to Community** option is not available for that community. You must remove a member before adding a new one.

**Note**

If you are logged into a community and you delete that community from some other CNA instance, then unless you close that community session, you can perform all the configurations through that session. After you close that session (which deletes the community), you cannot connect to that community.

Converting a Cluster into a Community

The Cluster Conversion wizard helps you convert a cluster into a community. When you complete the conversion, you can immediately manage the device group as a community. The benefits of managing a community is that the communication with the devices in a community is more secure (through multiple passwords and HTTPS) than in a cluster. Moreover, device availability is greater, and the range of devices that can be members is broader.

**Note**

The Cluster Conversion wizard does not alter your cluster definition. This means that you can still manage the devices as a cluster.

To launch the Cluster Conversion Wizard, follow these steps:

-
- Step 1** Start Network Assistant and connect to an existing cluster through its commander IP address.
- Step 2** In the feature bar, choose **Configure > Cluster > Cluster Conversion Wizard**.
- You see the query “or you want to convert this cluster to a community?”
- Step 3** Select **Yes** to proceed or **No** if you want to manually bring up the Cluster Conversion Wizard.
- If you select **Yes**, the Welcome screen appears, providing information about clusters, communities, and their benefits.
- A table appears listing the devices in the cluster starting with those that have no IP address and subnet mask. Be aware that all the devices in the cluster must have an IP address and subnet mask to be members of a community.
-
- Note** If a device has more than one interface with an IP address and subnet mask, you see more than one interface listed when you click in the cell. You can choose a different interface from the one originally shown.
-
- Step 4** In the IP Address column, enter an IP address for each device that does not have one.
- Step 5** In the Subnet Mask column, click in the cell for each device that does not have a subnet mask and select one.
- Step 6** Enter a name for the community.
- Step 7** Click **Finish** to begin the conversion.
- When the conversion completes, Network Assistant restarts and automatically connects to the newly created community.
-

**Note**

If you have enabled clustering, you should disable clustering before configuring a community (see [Table 16-2](#)).

Managing a Network Using Cluster

This section describes how to use clustering to create and manage Catalyst 4500 series switches using the standalone Network Assistant application or the command-line interface (CLI).

Use clustering to group the switches in your network. You must enter the **cluster run** command on each switch to be managed. The major advantage is that you can manage 16 devices with one IP address.

**Note**

Clustering is the auto-discovering mechanism used in CNA 1.0.

**Note**

For complete procedures for using Network Assistant to configure switch clusters, refer to *Getting Started with Cisco Network Assistant*, available at:

http://www.cisco.com/en/US/products/ps5931/prod_installation_guides_list.html.

This section contains the following topics:

- [Understanding Switch Clusters, page 16-11](#)
- [Using the CLI to Manage Switch Clusters, page 16-13](#)

Understanding Switch Clusters

These sections describes these topics:

- [Cluster Command Switch Requirements, page 16-11](#)
- [Network Assistant and VTY, page 16-12](#)
- [Candidate Switch and Cluster Member Switch Requirements, page 16-12](#)

Cluster Command Switch Requirements

A cluster command switch must meet these requirements:

- Uses Cisco IOS Release 12.2(20)EWA or later.
- Has an IP address.
- Cisco Discovery Protocol (CDP) version 2 is enabled (the default).
- Uses cluster-capable software and has clustering enabled.
- IP HTTP (or HTTPS) server is enabled.

**Note**

On a Catalyst 4500 series switch, neither HTTP or HTTPS is enabled by default.

- Has 16 VTY lines.



Note On a Catalyst 4500 series switch, the default is 4 lines. You configure the switch to set the value to 16.

- Is not a command or cluster member switch of another cluster.



Note If your switch cluster contains a Catalyst 4500 series switch, the cluster command switch must also be a Catalyst 4500 series switch.

Network Assistant and VTY

Network Assistant uses virtual terminal (VTY) lines to communicate with the cluster command device. Catalyst 4500 series switches have 5 VTY lines configured by default. Network Assistant can use an additional 8 lines. You should configure the maximum number of lines (or at least, $8 + 5 = 13$) so that Network Assistant can communicate with the switch and not use VTY lines that might be needed for Telnet.

You can configure the Catalyst 4500 series switch to support an appropriate number of VTY lines with the **line vty** configuration command. For example, the **line vty 6 15** command configures the switch to include 9 VTY lines.



Note If your existing VTY lines have nondefault configurations, you might want to apply those configurations to the new VTY lines.

Candidate Switch and Cluster Member Switch Requirements

Candidate switches are cluster-capable switches that are not part of a cluster. Cluster member switches are switches that are currently part of a switch cluster. Although not required, a candidate or cluster member switch can have its own IP address and password.



Note The hostname of a candidate should not be in the form [a-zA-Z0-9]-*n*, where *n* is 0 to 16. These names are reserved.

To join a cluster, a candidate switch must meet these requirements:

- Running cluster-capable software and has clustering enabled.
- Has CDP version 2 enabled.
- Has HTTP server enabled.



Note Even when HTTP is enabled on the commander switch, communication between the commander switch and member switch is still carried over HTTP.

- Has 16 VTY lines.
- Is not a command or cluster member switch of another cluster.
- Is connected to the cluster command switch through at least one common VLAN.

We recommend that you configure the Catalyst 4500 candidate and cluster member switches with an SVI on the VLAN connection to the cluster command switch.

Using the CLI to Manage Switch Clusters

You can configure cluster member switches from the CLI by first logging in to the cluster command switch. Enter the **rcommand** user EXEC command and the cluster member switch number to start a Telnet session (through a console or Telnet connection) and to access the cluster member switch CLI. The command mode changes and the Cisco IOS commands operate as usual. Enter the **exit** privileged EXEC command on the cluster member switch to return to the command-switch CLI.

This example shows how to log in to member-switch 3 from the command-switch CLI:

```
switch# rcommand 3
```

If you do not know the member-switch number, enter the **show cluster members** privileged EXEC command on the cluster command switch. For more information about the **rcommand** command and all other cluster commands, refer to the command reference guide.

The Telnet session accesses the member-switch CLI at the same privilege level as on the cluster command switch. The Cisco IOS commands will operate as usual. For instructions on configuring the switch for a Telnet session, see the [“Accessing the CLI Through Telnet”](#) section on page 2-2.



Note

CISCO-CLUSTER_MIB is not supported.

Configuring Network Assistant in Community or Cluster Mode

This section provides a detailed explanation of the CLI used to configure Network Assistant to work in a community or cluster. Network Assistant communicates with a Catalyst 4500 series switch by sending Cisco IOS commands over an HTTP (or HTTPS) connection.

This section includes the following topics:

- [Configuring Network Assistant on a Networked Switch in Community Mode, page 16-13](#)
- [Configuring Network Assistant in a Networked Switch in Cluster Mode, page 16-17](#)

Configuring Network Assistant on a Networked Switch in Community Mode

To configure Network Assistant on a networked switch in community mode, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# enable password <i>name</i>	Enables password protection of configuration mode.
Step 3	Switch(config)# vtp domain <i>name</i>	Creates a VTP domain to manage VLAN.
Step 4	Switch(config)# vlan <i>vlan_id</i>	Creates a VLAN.
Step 5	Switch(config-vlan)# interface { vlan <i>vlan_ID</i> { fastethernet gigabitethernet } <i>slot/interface</i> port-channel <i>number</i> }	Selects the interface that connects to your CNA-enabled PC.

	Command	Purpose
Step 6	Switch(config-if)# switchport access vlan <i>vlan_id</i>	Enables the selected interface to be in the specified VLAN.
Step 7	Switch(config-if)# interface { vlan <i>vlan_ID</i> slot/interface Port-channel <i>number</i> }	Select the VLAN instance for configuration.
Step 8	Switch(config-if)# ip address <i>ip_address</i>	Assigns an IP address to the SVI.
Step 9	Switch(config-if)# no shutdown	Enables the interface.
Step 10	Switch(config-if)# ip http server	Starts the HTTP server so that Network Assistant can talk to the switch.
Step 11	Switch(config)# ip domain-name <i>domain_name</i>	Enables the domain name on the switch to configure HTTPS.
Step 12	Switch(config)# ip http secure-server	Enables the HTTPS server on the switch. By default, the HTTPS server is disabled.
Step 13	Switch(config)# ip http max-connections <i>connection_number</i>	Configures the maximum concurrent connections to the HTTP server. <i>A connection_number</i> of 16 is recommended.
Step 14	Switch(config)# ip http timeout-policy idle <i>idle_time</i> life <i>life_time</i> requests <i>requests</i>	Configures the HTTPS port. The idle keyword specifies the maximum amount of time a connection can stay idle. A idle value of 180 seconds is recommended. The life keyword specifies the maximum amount of time a connection can stay open since it was established. A life value of 180 seconds is recommended. The requests keyword specifies the maximum number of requests on a connection. A requests value of 25 recommended.
Step 15	Switch(config-if)# ip http secure-server	(Optionally) Enables the switch to accept HTTPS connections from Network Assistant.
Step 16	Switch(config)# ip route <i>a.b.c</i>	Establishes the route to the default router, usually supplied by the local Internet provider. Note This line represents the only difference between the configuration for a standalone and a networked switch.
Step 17	Switch(config)# line con 0	Selects the console port to perform the configuration.
Step 18	Switch(config-line)# exec-timeout <i>x y</i>	Configures an automatic session logout if no keyboard input or output is displayed on the terminal.
Step 19	Switch(config-line)# password <i>password</i>	Specifies a password for the console port.
Step 20	Switch(config-line)# login	Allows login to the console port.
Step 21	Switch(config-line)# line vty <i>x y</i>	Creates additional VTY lines for CNA to access the switch.
Step 22	Switch(config-line)# password <i>password</i>	Specifies a password for the switch.
Step 23	Switch(config-line)# login	Allows login to the switch.
Step 24	Switch(config-line)# line vty <i>x y</i>	Creates additional VTY lines for CNA to access the switch.
Step 25	Switch(config-line)# password <i>password</i>	Specifies a password for the switch.
Step 26	Switch(config-line)# login	Allows login to the switch.

	Command	Purpose
Step 27	Switch(config-line)# end	Returns to privileged EXEC mode.
Step 28	Switch# show running-config	Verifies the configuration.

This example shows how to configure Network Assistant on a networked switch in community mode:

```
Switch# configure terminal
Switch(config)# vtp domain cnadoc
Changing VTP domain name from cisco to cnadoc
Switch(config)# vlan 2
Switch(config-vlan)# exit
Switch(config)# interface GigabitEthernet 2/1
Switch(config-if)# switchport access vlan 2
Switch(config-if)# exit
Switch(config)# interface vlan 2
Switch(config-if)# ip address 123.123.123.1 255.255.255.0
Switch(config-if)# no shutdown
Switch(config-if)# exit
Switch(config)# ip http server
Switch(config)# ip domain-name cisco.com
Switch(config)# ip http secure-server
Switch(config)# ip http max-connections 16
Switch(config)# ip http timeout-policy idle 180 life 180 requests 25
Switch(config)# ip route 0.0.0.0 0.0.0.0 123.123.123.2
Switch(config)# line con 0
Switch(config-line)# exec-timeout 0 0
Switch(config-line)# password keepout
Switch(config-line)# login
Switch(config-line)# line vty 5 15
Switch(config-line)# password keepout
Switch(config-line)# login
Switch(config-line)# line vty 5 15
Switch(config-line)# end
Switch# show running-config
Building configuration...

Current configuration : 1426 bytes
!
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service compress-config
!
hostname Switch
!
boot-start-marker
boot-end-marker
!
enable password cna
!
no aaa new-model
ip subnet-zero
ip domain-name cisco.com
!
vtp domain cnadoc
vtp mode transparent
!
crypto pki trustpoint TP-self-signed-913087
    enrollment selfsigned
```

```

subject-name cn=IOS-Self-Signed-Certificate-913087
revocation-check none
rsakeypair TP-self-signed-913087
!!
crypto pki certificate chain TP-self-signed-913087
certificate self-signed 01
  3082028E 308201F7 A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  52312B30 29060355 04031322 494F532D 53656C66 2D536967 6E65642D 43657274
  69666963 6174652D 39313330 38373123 30210609 2A864886 F70D0109 02161456
  61646572 2D343531 302E6369 73636F2E 636F6D30 1E170D30 36303432 30323332
  3435305A 170D3230 30313031 30303030 30305A30 52312B30 29060355 04031322
  494F532D 53656C66 2D536967 6E65642D 43657274 69666963 6174652D 39313330
  38373123 30210609 2A864886 F70D0109 02161456 61646572 2D343531 302E6369
  73636F2E 636F6D30 819F300D 06092A86 4886F70D 01010105 0003818D 00308189
  02818100 F2C86FEA 49C37856 D1FA7CB2 9AFF748C DD443295 F6EC900A E83CDA8E
  FF8F9367 0A1E7A20 C0D3919F 0BAC2113 5EE37525 94CF24CF 7B313C01 BF177A73
  494B1096 B4D24729 E087B39C E44ED9F3 FCCD04BB 4AD3C6BF 66E0902D E234D08F
  E6F6C001 BAC80854 D4668160 9299FC73 C14A33F3 51A17BF5 8C0BEA07 3AC03D84
  889F2661 02030100 01A37430 72300F06 03551D13 0101FF04 05300301 01FF301F
  0603551D 11041830 16821456 61646572 2D343531 302E6369 73636F2E 636F6D30
  1F060355 1D230418 30168014 BB013B0D 00391D79 B628F2B3 74FC62B4 077AD908
  301D0603 551D0E04 160414BB 013B0D00 391D79B6 28F2B374 FC62B407 7AD90830
  0D06092A 864886F7 0D010104 05000381 81002963 26762EFA C52BA4B3 6E641A9D
  742CE404 E45FECB1 B5BD2E74 6F682476 A7C3DAA5 94393AE3 AA103B6E 5974F81B
  09DF16AE 7F9AE67C 5CB3D5B1 B945A5F3 36A8CC8C 8F142364 F849344D 5AE36410
  51182EB9 24A9330B 3583E1A3 79151470 D304C157 3417E240 52BE2A91 FC7BBEDE
  562BEDAD E6C46D9A F7FF3148 4CE9CEE1 5B17
quit
!
!
!
power redundancy-mode redundant
no file verify auto
spanning-tree mode pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
vlan 2
!
interface GigabitEthernet1/1
  switchport access vlan 2
!
interface GigabitEthernet1/2
!
interface GigabitEthernet1/3
!
interface GigabitEthernet1/4
!
interface GigabitEthernet1/5
!
interface GigabitEthernet1/6
!
interface GigabitEthernet1/7
!
interface GigabitEthernet1/8
!
interface GigabitEthernet1/9
!
interface GigabitEthernet1/10
!
interface GigabitEthernet1/11
!
interface GigabitEthernet1/12

```

```

!
interface GigabitEthernet1/13
!
interface GigabitEthernet1/14
!
interface GigabitEthernet1/15
!
interface GigabitEthernet1/16
!
interface GigabitEthernet1/17
!
interface GigabitEthernet1/18
!
interface GigabitEthernet1/19
!
interface GigabitEthernet1/20
!
interface Vlan1
 no ip address
!
interface Vlan2
 ip address 123.123.123.1 255.255.255.0
!
ip route 0.0.0.0 0.0.0.0 123.123.123.2
ip http server
ip http secure-server
ip http max-connections 16
ip http timeout-policy idle 180 life 180 requests 25
!
line con 0
 password cna
 login
 stopbits 1
line vty 0 4
 password cna
 login
line vty 5 15
 password cna
 login
!
!
end

Switch#

```

Configuring Network Assistant in a Networked Switch in Cluster Mode

To configure Network Assistant on a networked switch in cluster mode, perform this task on the switch:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# enable password <i>name</i>	Enables password protection of configuration mode.
Step 3	Switch(config)# vtp domain <i>name</i>	Creates a VTP domain to manage VLANs and names.
Step 4	Switch(config)# cluster run	Launches the cluster on the cluster commander.
Step 5	Switch(config)# cluster enable <i>cluster_name</i>	Makes the switch the cluster commander.
Step 6	Switch(config)# vlan <i>vlan_id</i>	Creates a VLAN.

	Command	Purpose
Step 7	Switch(config-vlan)# interface { vlan <i>vlan_ID</i> { fastethernet gigabitethernet } <i>slot/interface</i> Port-channel <i>number</i> }	Selects the interface that connects to your CNA-enabled PC.
Step 8	Switch(config-if)# switchport access vlan <i>vlan_id</i>	Enables the physical port to be in the specified VLAN.
Step 9	Switch(config-if)# interface { vlan <i>vlan_ID</i> <i>slot/interface</i> Port-channel <i>number</i> }	Select the VLAN instance for configuration.
Step 10	Switch(config-if)# ip address <i>ip_address</i>	Assigns an IP address to the SVI.
Step 11	Switch(config-if)# no shut	Enables the interface.
Step 12	Switch(config-if)# ip http server	Starts the HTTP server so that Network Assistant can talk to the switch.
Step 13	Switch(config)# ip http secure-server	(Optionally) Enables the switch to accept HTTPS connections from Network Assistant.
Step 14	Switch(config)# ip route <i>a.b.c</i>	Establishes the route to the default router, usually supplied by the local Internet provider. Note This line represents the only difference between the configuration for a standalone and a networked switch.
Step 15	Switch(config)# line con 0	Selects the console port to perform the configuration.
Step 16	Switch(config-line)# exec-timeout <i>x y</i>	Configures an automatic session logout if no keyboard input or output is displayed on the terminal.
Step 17	Switch(config-line)# password <i>password</i>	Specifies a password for the console port.
Step 18	Switch(config-line)# login	Allows login to the console port.
Step 19	Switch(config-line)# line vty <i>x y</i>	Creates additional VTY lines for CNA to access the switch.
Step 20	Switch(config-line)# password <i>password</i>	Specifies a password for the switch.
Step 21	Switch(config-line)# login	Allows login to the switch.
Step 22	Switch(config-line)# line vty <i>x y</i>	Creates additional VTY lines for CNA to access the switch.
Step 23	Switch(config-line)# password <i>password</i>	Specifies a password for the switch.
Step 24	Switch(config-line)# login	Allows login to the switch.
Step 25	Switch(config-line)# end	Returns to privileged EXEC mode.
Step 26	Switch# show running-config include http	Verifies that the HTTP server is enabled.

This example shows how to configure Network Assistant on a networked switch in cluster mode:

```
Switch# configure terminal
Switch(config)# vtp domain cnadoc
Switch(config)# cluster run
Switch(config)# cluster enable cnadoc
Switch(config)# vlan 10
Switch(config-vlan)# interface GigabitEthernet 2/1
Switch(config-if)# switchport access vlan 10
Switch(config-if)# interface vlan10
Switch(config-if)# ip address aa.bb.cc.dd
Switch(config-if)# no shut
Switch(config-if)# ip http server
Switch(config-if)# ip http secure-server
Switch(config)# ip route 0.0.0.0 0.0.0.0 123.123.123.2
```

```
Switch(config)# line con 0
Switch(config-line)# exec-timeout 0 0
Switch(config-line)# password keepout
Switch(config-line)# login
Switch(config-line)# line vty 5 15
Switch(config-line)# password keepout
Switch(config-line)# login
Switch(config-line)# line vty 5 15
Switch(config-line)# end
Switch# show running-config
Building configuration...

Current configuration : 1469 bytes
!
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service compress-config
!
hostname Switch
!
boot-start-marker
boot-end-marker
!
enable password cna
!
no aaa new-model
ip subnet-zero
!
vtp domain cnadoc
vtp mode transparent
cluster run
cluster enable cnadoccluster 0
!
!
!
!
power redundancy-mode redundant
no file verify auto
spanning-tree mode pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
vlan 2
!
interface GigabitEthernet1/1
    switchport access vlan 2
!
interface GigabitEthernet1/2
!
interface GigabitEthernet1/3
!
interface GigabitEthernet1/4
!
interface GigabitEthernet1/5
!
interface GigabitEthernet1/6
!
interface GigabitEthernet1/7
!
```

```
interface GigabitEthernet1/8
!
interface GigabitEthernet1/9
!
interface GigabitEthernet1/10
!
interface GigabitEthernet1/11
!
interface GigabitEthernet1/12
!
interface GigabitEthernet1/13
!
interface GigabitEthernet1/14
!
interface GigabitEthernet1/15
!
interface GigabitEthernet1/16
!
interface GigabitEthernet1/17
!
interface GigabitEthernet1/18
!
interface GigabitEthernet1/19
!
interface GigabitEthernet1/20
!
interface Vlan1
  no ip address
!
interface Vlan2
  ip address 123.123.123.1 255.255.255.0
!
ip route 0.0.0.0 0.0.0.0 123.123.123.2
ip http server
no ip http secure-server
!
!
!
line con 0

Switch#
```



Configuring VLANs, VTP, and VMPS

This chapter describes VLANs on Catalyst 4500 series switches. It also describes how to enable the VLAN Trunking Protocol (VTP) and to configure the Catalyst 4500 series switch as a VMPS client.

This chapter includes the following major sections:

- [VLANs, page 17-1](#)
- [VLAN Trunking Protocol, page 17-7](#)
- [VLAN Membership Policy Server, page 17-20](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

VLANs

This section includes the following major subsections:

- [About VLANs, page 17-1](#)
- [VLAN Configuration Guidelines and Restrictions, page 17-3](#)
- [VLAN Default Configuration, page 17-4](#)
- [Configuring VLANs, page 17-5](#)

About VLANs

A VLAN is a group of devices on one or more LANs that are configured to communicate as if they were attached to the same wire, when in fact they are located on a number of different LAN segments. Because VLANs are based on logical instead of physical connections, they are extremely flexible.



Note

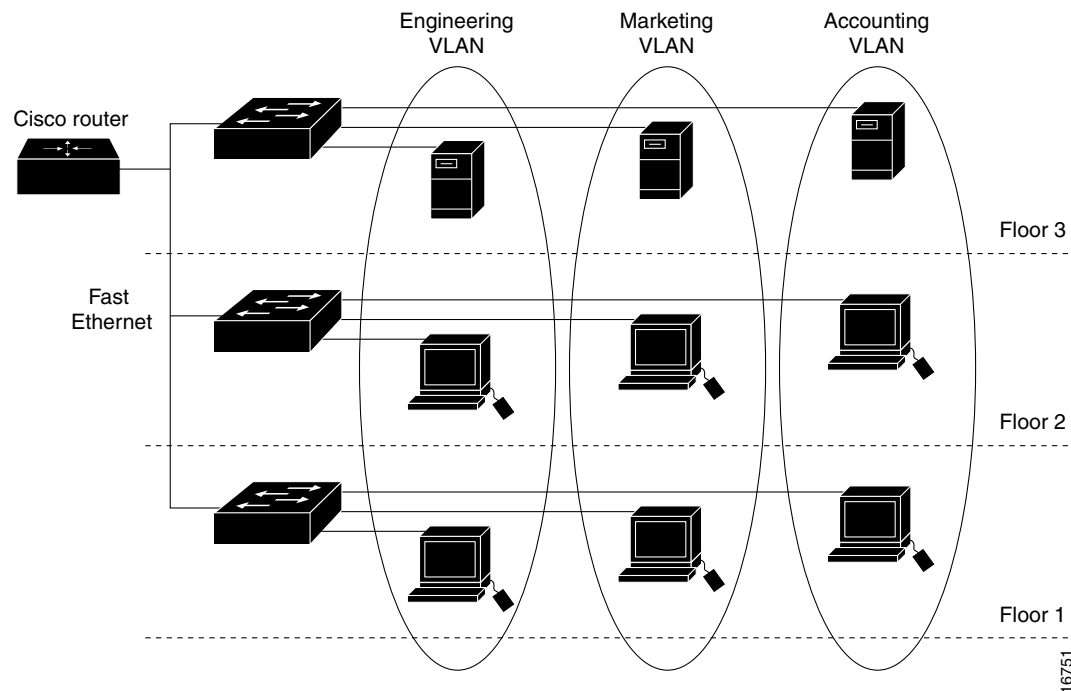
VTP version 3 updates do not pass through promiscuous trunk ports.

VLANs define broadcast domains in a Layer 2 network. A broadcast domain is the set of all devices that receives broadcast frames originating from any device within the set. Broadcast domains are typically bounded by switches because switches do not forward broadcast frames. Layer 2 switches create broadcast domains based on the configuration of the switch. Switches are multiport bridges that allow you to create multiple broadcast domains. Each broadcast domain is like a distinct virtual bridge within a switch.

You can define one or many virtual bridges within a switch. Each virtual bridge you create in the switch defines a new broadcast domain (VLAN). Traffic cannot pass directly to another VLAN (between broadcast domains) within the switch or between two switches. To interconnect two different VLANs, you must use switches or Layer 3 switches. See the [“About Layer 3 Interfaces” section on page 36-1](#) for information on inter-VLAN routing on Catalyst 4500 series switches.

Figure 17-1 shows an example of three VLANs that create logically defined networks.

Figure 17-1 Sample VLANs



VLANs are often associated with IP subnetworks. For example, all of the end stations in a particular IP subnet belong to the same VLAN. Traffic between VLANs must be routed. You must assign LAN interface VLAN membership on an interface-by-interface basis (termed interface-based or static VLAN membership).

You can set the following parameters when you create a VLAN in the management domain:

- VLAN number
- VLAN name
- VLAN type
- VLAN state (active or suspended)
- Maximum transmission unit (MTU) for the VLAN
- Security Association Identifier (SAID)

- VLAN number to use when translating from one VLAN type to another

**Note**

When the software translates from one VLAN type to another, it requires a different VLAN number for each media type.

VLAN Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when creating and modifying VLANs in your network:

- Before creating a VLAN, put the Catalyst 4500 series switch in VTP server mode or VTP transparent mode. If the Catalyst 4500 series switch is a VTP server, you must define a VTP domain. For information on configuring VTP, see the [“VLAN Trunking Protocol” section on page 17-7](#).
- You cannot use the **end** command in VLAN database mode.
- You cannot use **Ctrl-Z** to exit VLAN database mode.
- If a switch running MSTP and configured with all possible VLANs (4094) is in the path of two HSRP peers with the timeout set below 500 ms, HSRP flaps.

Workarounds:

- Use fewer VLANs.
- Set the timers greater than 600 ms.
- Enter the **no igmp snooping** (globally) and **access-list hardware capture mode VLAN** commands

VLAN Ranges

**Note**

You must enable the extended system ID to use 4094 VLANs. See the [“Understanding the Bridge ID” section on page 23-2](#).

With Cisco IOS Release 12.2(31)SGA and later, Catalyst 4500 series switches support 4096 VLANs in compliance with the IEEE 802.1Q standard. These VLANs are organized into three ranges: reserved, normal, and extended.

Some of these VLANs are propagated to other switches in the network when you use the VLAN Trunking Protocol (VTP). The extended-range VLANs are not propagated, so you must configure extended-range VLANs manually on each network device.

[Table 17-1](#) describes the uses for VLAN ranges.

Table 17-1 **VLAN Ranges**

VLANs	Range	Usage	Propagated by VTP
0, 4095	Reserved	For system use only. You cannot see or use these VLANs.	—
1	Normal	Cisco default. You cannot delete this VLAN.	Yes
2–1001	Normal	Used for Ethernet VLANs; you can create, use, and delete these VLANs.	Yes

Table 17-1 VLAN Ranges

VLANs	Range	Usage	Propagated by VTP
1002–1005	Normal	Cisco defaults for FDDI and Token Ring. You cannot delete VLANs 1002–1005.	Yes
1006–4094	Extended	<p>For Ethernet VLANs only. When configuring extended-range VLANs, note the following:</p> <ul style="list-style-type: none"> Layer 3 ports and some software features require internal VLANs. Internal VLANs are allocated from 1006 and up. You cannot use a VLAN that has been allocated for such use. To display the VLANs used internally, enter the show vlan internal usage command. Switches running the Catalyst operating system do not support configuration of VLANs 1006-1024. If you configure VLANs 1006-1024, ensure that the VLANs do not extend to any switches running Catalyst operating system software. You must enable the extended system ID to use extended range VLANs. 	No

Configurable Normal-Range VLAN Parameters


Note

Ethernet VLANs 1 and 1006 through 4094 use only default values.

You can configure the following parameters for VLANs 2 through 1001:

- VLAN name
- VLAN type
- VLAN state (active or suspended)
- SAID
- STP type for VLANs

VLAN Default Configuration

Table 17-2 shows the default VLAN configuration values.

Table 17-2 Ethernet VLAN Defaults and Ranges

Parameter	Default	Valid Values
VLAN ID	1	1–4094
VLAN name	VLANx, where x is a number assigned by the software.	No range
802.1Q SAID	100,001	1–4,294,967,294
MTU size	1500	1500–18,190

Table 17-2 Ethernet VLAN Defaults and Ranges (continued)

Parameter	Default	Valid Values
Translational bridge 1	1002	0–1005
Translational bridge 2	1003	0–1005
VLAN state	active	active; suspend; shutdown

**Note**

Catalyst 4500 series switches do not support Token Ring or FDDI media. The switch does not forward FDDI, FDDI-NET, TrCRF, or TrBRF traffic, but it does propagate the VLAN configuration by using VTP. The software reserves parameters for these media types, but they are not supported.

Configuring VLANs

**Note**

Before you configure VLANs, you must use VLAN Trunking Protocol (VTP) to maintain global VLAN configuration information for your network. For complete information on VTP, see the [“VLAN Trunking Protocol” section on page 7](#).

**Note**

VLANs support a number of parameters that are not discussed in detail in this section. For complete information, refer to the command reference guide.

**Note**

The VLAN configuration is stored in the **vlan.dat** file, which is stored in nonvolatile memory. You can cause inconsistency in the VLAN database if you manually delete the **vlan.dat** file. If you want to modify the VLAN configuration or VTP, use the commands described in the following sections and in the command reference guide.

The following sections describe how to configure VLANs:

- [Configuring VLANs in Global Configuration Mode, page 17-5](#)
- [Assigning a Layer 2 LAN Interface to a VLAN, page 17-7](#)

Configuring VLANs in Global Configuration Mode

If the switch is in VTP server or transparent mode (see the [“VLAN Trunking Protocol” section on page 17-7](#)), you can configure VLANs in global and VLAN configuration modes. When you configure VLANs in global and config-vlan configuration modes, the VLAN configuration is saved in the **vlan.dat** files, not the **running-config** or **startup-config** files. To display the VLAN configuration, enter the **show vlan** command.

If the switch is in VLAN transparent mode, use the **copy running-config startup-config** command to save the VLAN configuration to the **startup-config** file. After you save the running configuration as the startup configuration, the **show running-config** and **show startup-config** commands display the VLAN configuration.

When the switch boots, if the VTP domain name and VTP mode in the **startup-config** and **vlan.dat** files do not match, the switch uses the configuration in the **vlan.dat** file.

You use the interface configuration command mode to define the port membership mode and add and remove ports from a VLAN. The results of these commands are written to the **running-config** file, and you can display the contents of the file by entering the **show running-config** command. Beginning with Cisco IOS Release 15.2(2)E and Cisco IOS XE Release 3.6E, if the VTP mode is off or transparent, VLAN configuration is saved to the **startup-config** file, even when the configuration is not applied to the interface.

User-configured VLANs have unique IDs from 1 to 4094. To create a VLAN, enter the **vlan** command with an unused ID. To verify whether a particular ID is in use, enter the **show vlan id ID** command. To modify a VLAN, enter the **vlan** command for an existing VLAN.

See the “[VLAN Default Configuration](#)” section on page 17-4 for the list of default parameters that are assigned when you create a VLAN. If you do not use the **media** keyword when specifying the VLAN type, the VLAN is an Ethernet VLAN.

To create a VLAN, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# vlan <i>vlan_ID</i> Switch(config-vlan)#	<p>Adds an Ethernet VLAN.</p> <p>Note You cannot delete the default VLANs for these media types: Ethernet VLAN 1 and FDDI or Token Ring VLANs 1002 to 1005.</p> <p>When you delete a VLAN, any LAN interfaces configured as access ports assigned to that VLAN become inactive. They remain associated with the VLAN (and thus inactive) until you assign them to a new VLAN.</p> <p>Use the no keyword to delete a VLAN.</p> <p>When the prompt shows Switch(config-vlan)#; you are in vlan-configuration mode. If you want to change any of the parameters for the newly created VLAN, use this mode.</p>
Step 3	Switch(config-vlan)# end	Returns to enable mode from vlan-configuration mode.
Step 4	Switch# show vlan [<i>id</i> <i>name</i>] <i>vlan_name</i>	Verifies the VLAN configuration.

When you create or modify an Ethernet VLAN, note the following:

- Because Layer 3 ports and some software features require internal VLANs allocated from 1006 and up, configure extended-range VLANs starting with 4094 and work downward.
- You can configure extended-range VLANs only in global configuration mode. You cannot configure extended-range VLANs in VLAN database mode.
- Layer 3 ports and some software features use extended-range VLANs. If the VLAN you are trying to create or modify is being used by a Layer 3 port or a software feature, the switch displays a message and does not modify the VLAN configuration.
- When you create VLANs with the VLAN configuration command, they are automatically added to the existing VTP domain; no action is required of the user.

This example shows how to create an Ethernet VLAN in global configuration mode and verify the configuration:

```
Switch# configure terminal
Switch(config)# vlan 3
Switch(config-vlan)# end
Switch# show vlan id 3
```

VLAN Name	Status	Ports
3 VLAN0003	active	

VLAN	Type	SAID	MTU	Parent	RingNo	BridgeNo	Stp	BrdgMode	Trans1	Trans2
3	enet	100003	1500	-	-	-	-	-	0	0

Primary	Secondary	Type	Interfaces

```
Switch#
```

Assigning a Layer 2 LAN Interface to a VLAN

A VLAN created in a management domain remains unused until you assign one or more LAN interfaces to the VLAN.



Note

Make sure you assign LAN interfaces to a VLAN of the proper type. Assign Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet interfaces to Ethernet-type VLANs.

To assign one or more LAN interfaces to a VLAN, complete the procedures in the [“Configuring Ethernet Interfaces for Layer 2 Switching”](#) section on page 19-5.

VLAN Trunking Protocol

This section describes the VLAN Trunking Protocol (VTP) on the Catalyst 4500 series switches, and includes the following major subsections:

- [About VTP, page 17-7](#)
- [VTP Configuration Guidelines and Restrictions, page 17-12](#)
- [VTP Default Configuration, page 17-13](#)
- [Configuring VTP, page 17-13](#)

About VTP

VTP is a Layer 2 messaging protocol that maintains VLAN configuration consistency by managing the addition, deletion, and renaming of VLANs within a VTP domain. A VTP domain (also called a VLAN management domain) is made up of one or more network devices that share the same VTP domain name and that are interconnected with trunks. VTP minimizes misconfigurations and configuration inconsistencies that can result in a number of problems, such as duplicate VLAN names, incorrect VLAN-type specifications, and security violations.

Before you create VLANs, you must decide whether you want to use VTP in your network. With VTP, you can make configuration changes centrally on one or more network devices and have those changes automatically communicated to all the other network devices in the network. For details on configuring VLANs, see the [“VLANs”](#) section on page 17-1

These sections describe how VTP works:

- [Understanding the VTP Domain, page 17-8](#)
- [Understanding VTP Modes, page 17-8](#)
- [Understanding VTP Advertisements, page 17-9](#)
- [Understanding VTP Versions, page 17-9](#)
- [Understanding VTP Pruning, page 17-10](#)

Understanding the VTP Domain

A VTP domain is made up of one or more interconnected network devices that share the same VTP domain name. A network device can be configured to be in only one VTP domain. You make global VLAN configuration changes for the domain using either the command-line interface (CLI) or Simple Network Management Protocol (SNMP).

By default, the Catalyst 4500 series switch is in VTP server mode and the domain is set to NULL until the switch receives an advertisement for a domain over a trunk link or you configure a management domain. You cannot create or modify VLANs on a VTP server until the management domain name is specified or learned.

If the switch receives a VTP advertisement over a trunk link, it inherits the management domain name and the VTP configuration revision number. The switch ignores advertisements with a different management domain name or an earlier configuration revision number.

If you configure the switch as VTP transparent, you can create and modify VLANs, but the changes affect only the individual switch.

When you make a change to the VLAN configuration on a VTP server, the change is propagated to all network devices in the VTP domain. VTP advertisements are transmitted out all Inter-Switch Link (ISL) and IEEE 802.1Q trunk connections.

VTP maps VLANs dynamically across multiple LAN types with unique names and internal index associations. Mapping eliminates unnecessary device administration for network administrators.

Understanding VTP Modes

You can configure a Catalyst 4500 series switch to operate in any one of these VTP modes:

- **Server**—In VTP server mode, you can create, modify, and delete VLANs and specify other configuration parameters (such as VTP version and VTP pruning) for the entire VTP domain. VTP servers advertise their VLAN configuration to other network devices in the same VTP domain and synchronize their VLAN configuration with other network devices based on advertisements received over trunk links. VTP server is the default mode.



Note In VTP version 3, manipulation of VLANs can be done only to primary servers.

- **Client**—VTP clients behave the same way as VTP servers, but you cannot create, change, or delete VLANs on a VTP client.
- **Transparent**—VTP transparent network devices do not participate in VTP. A VTP transparent network device does not advertise its VLAN configuration and does not synchronize its VLAN configuration based on received advertisements. However, in VTP version 2, transparent network devices do forward VTP advertisements that they receive on their trunking LAN interfaces.

- Off—In VTP off mode, a network device functions in the same manner as a VTP transparent device except that it does not forward VTP advertisements.

**Note**

Catalyst 4500 series switches automatically change from VTP server mode to VTP client mode if the switch detects a failure while writing configuration to NVRAM. If this happens, the switch cannot be returned to VTP server mode until the NVRAM is functioning.

Understanding VTP Advertisements

Each network device in the VTP domain sends periodic advertisements out each trunking LAN interface to a reserved multicast address. VTP advertisements are received by neighboring network devices, which update their VTP and VLAN configurations as necessary.

The following global configuration information is distributed in VTP advertisements:

- VLAN IDs (ISL and 802.1Q)
- Emulated LAN names (for ATM LANE)
- 802.10 SAID values (FDDI)
- VTP domain name
- VTP configuration revision number
- VLAN configuration, including maximum transmission unit (MTU) size for each VLAN
- Frame format

Understanding VTP Versions

VTP Version 2

If you use VTP in your network, you must decide whether to use VTP version 2 or version 3.

**Note**

Catalyst 4500 series switches do not support Token Ring or FDDI media. The switch does not forward FDDI, FDDI-Net, Token Ring Concentrator Relay Function (TrCRF), or Token Ring Bridge Relay Function (TrBRF) traffic, but it does propagate the VLAN configuration by using VTP.

VTP version 2 supports the following features, which are not supported in version 1:

- Token Ring support—Supports Token Ring LAN switching and VLANs (TrBRF and TrCRF).
- Unrecognized Type-Length-Value (TLV) support—A VTP server or client propagates configuration changes to its other trunks, even for TLVs it is not able to parse. The unrecognized TLV is saved in NVRAM.
- Version-dependent transparent mode—In VTP version 1 and version 2, a VTP transparent network device forwards VTP messages in transparent mode without checking the version.
- Consistency checks—In VTP version 2, VLAN consistency checks (such as VLAN names and values) are performed only when you enter new information through the CLI or SNMP. Consistency checks are not performed when new information is obtained from a VTP message or when information is read from NVRAM. If the digest on a received VTP message is correct, its information is accepted without consistency checks.

VTP Version 3

VTP version 3 supports the following features not supported in version 1 or version 2:

- Hidden password support—Supports the option of configuring the password as **hidden** or **secret**.
When the **hidden** keyword is specified, that password must be reentered if a takeover command is issued in the domain. The secret key generated from the password string is saved in the `const_nvram:vlan.dat` file. When configured with this option, the password does not appear in plain text in the configuration. Instead, the secret key associated with the password is saved in hexadecimal format in the running configuration. If the **hidden** keyword is not specified, the password is saved in clear text in the `const_nvram:vlan.dat` file as in VTP version 1 and VTP version 2.
When the **secret** keyword is specified, the password secret key can be directly configured.
- Extended VLAN database propagation support—In VTP version 2, VLAN configuration information is propagated only for VLANs numbered 1 to 1000. In VTP version 3, information also is propagated for extended-range VLANs (VLANs numbered 1006 to 4094).
- On Catalyst 4500 series switches running VTP version 1, VTP version 2, or VTP version 3, default VLANs 1 and 1002 to 1005 cannot be modified.



Note VTP pruning continues to apply only to VLANs numbered 1 to 1000.

- Propagation of any database in a domain—In addition to propagating VLAN database information, VTP can propagate Multiple Spanning Tree (MST) protocol database information.
- Disabling VTP—When VTP is disabled on a trunking port, it applies to all VTP instances on that port. When VTP is disabled globally, the setting applies to all the trunking ports in the system.
- In VTP version 1 and VTP version 2, the role of a VTP server is to back up the database to NVRAM and to allow the administrator to change database information. VTP version 3 introduces the roles of VTP primary server and VTP secondary server. A VTP primary server is used to update the database information. The updates sent out are honored by all the devices in the system. A VTP secondary server can only back up to its NVRAM the VTP configuration received by using updates from the VTP primary server.

The status of primary and secondary servers is a runtime status and is not a configurable option. By default, all devices are initiated as secondary servers. Primary server status is needed only when database updates are needed, and is obtained when the administrator issues a takeover message in the domain. See the [“Starting a Takeover” section on page 17-18](#).

Primary server status is lost upon reload of the device, or when switchover or domain parameters change. Secondary servers back up the configuration and continue to propagate it. Because of that, you may have a working VTP domain without any primary servers.

Understanding VTP Pruning

VTP pruning enhances network bandwidth use by reducing unnecessary flooded traffic, such as broadcast, multicast, and unicast packets. VTP pruning increases available bandwidth by restricting flooded traffic to those trunk links that the traffic must use to access the appropriate network devices. By default, VTP pruning is disabled.

For VTP pruning to be effective, all devices in the management domain must either support VTP pruning or, on devices that do not support VTP pruning, you must manually configure the VLANs allowed on trunks.

Figure 17-2 shows a switched network without VTP pruning enabled. Interface 1 on Switch 1 and Interface 2 on Switch 4 are assigned to the Red VLAN. A broadcast is sent from the host connected to Switch 1. Switch 1 floods the broadcast and every network device in the network receives it, even though Switches 3, 5, and 6 have no interfaces in the Red VLAN.

You can enable pruning globally on the Catalyst 4500 series switch (see the “Enabling VTP Pruning” section on page 17-15).

Figure 17-2 Flooding Traffic without VTP Pruning

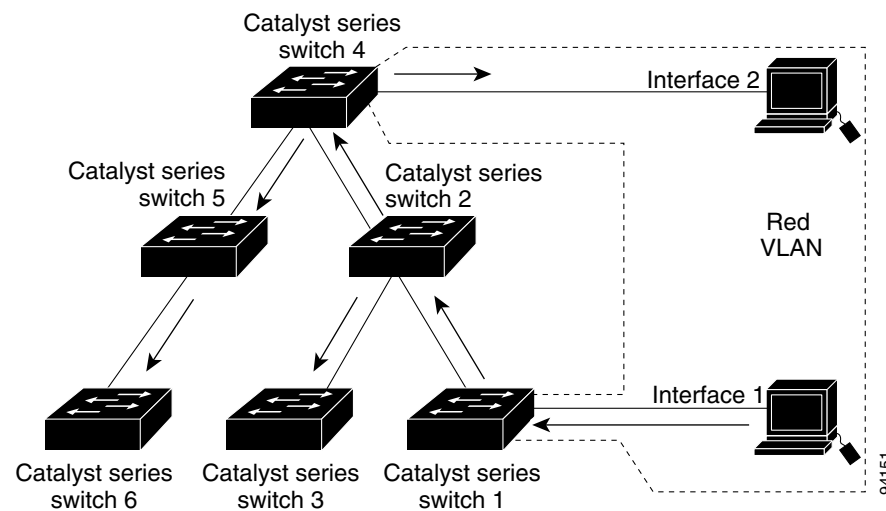
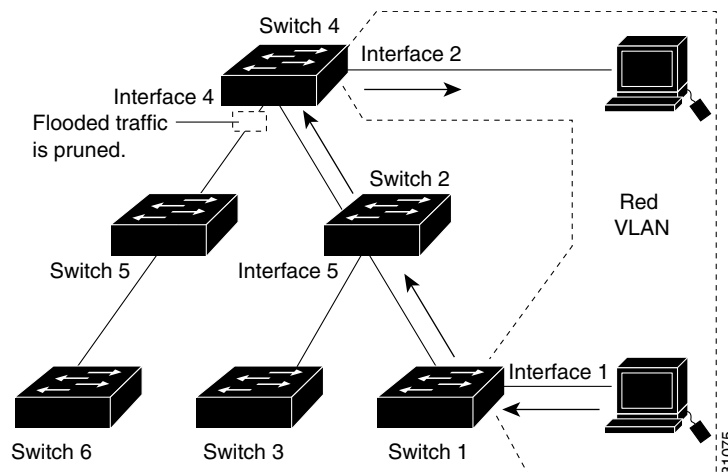


Figure 17-3 shows the same switched network with VTP pruning enabled. The broadcast traffic from Switch 1 is not forwarded to Switches 3, 5, and 6 because traffic for the Red VLAN has been pruned on the links indicated (Interface 5 on Switch 2 and Interface 4 on Switch 4).

Figure 17-3 Flooding Traffic with VTP Pruning



Enabling VTP pruning on a VTP server enables pruning for the entire management domain. VTP pruning takes effect several seconds after you enable it. By default, VLANs 2 through 1000 are eligible for pruning. VTP pruning does not prune traffic from pruning-eligible VLANs. VLAN 1 is always ineligible for pruning; traffic from VLAN 1 cannot be pruned.

To configure VTP pruning on a trunking LAN interface, use the **switchport trunk pruning vlan** command. VTP pruning operates when a LAN interface is trunking. You can set VLAN pruning eligibility regardless of whether VTP pruning is enabled or disabled for the VTP domain, whether any given VLAN exists, and regardless of whether the LAN interface is currently trunking.

VTP Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when implementing VTP in your network:

- Supervisor engine redundancy does not support nondefault VLAN data file names or locations. Do not enter the **vtp file *file_name*** command on a switch that has a redundant supervisor engine.
- Before installing a redundant supervisor engine, enter the **no vtp file** command to return to the default configuration.
- When a VTP version 3 device on a trunk port receives messages from a VTP version 2 device, it sends a scaled-down version of the VLAN database on that particular trunk in a VTP version 2 format. A VTP version 3 device does not send out VTP version 2 formatted packets on a trunk port unless it first receives VTP version 2 packets on that trunk.
- Even when a VTP version 3 device detects a VTP version 2 device on a trunk port, it continues to send VTP version 3 packets in addition to VTP version 2 packets, to allow co-existence of two kinds of neighbors off the trunk.
- A VTP version 3 device does not accept configuration information from a VPT version 2 or version 1 device.
- Unlike in VPT version 2, when VTP is configured to be version 3, this does not configure all the version-3-capable devices in the domain to start behaving as VPT version 3 systems.
- When a VTP version 1 device, capable of version 2 or version 3, receives a VTP version 3 packet, the device is configured as a VTP version 2 device provided a VTP version 2 conflict does not exist.
- Devices that are only VTP version 1 capable cannot interoperate with VTP version 3 devices.
- In a Token Ring environment, you must enable VTP version 2 or version 3 for Token Ring VLAN switching to function properly.
- Two VPT version 3 regions can only communicate in transparent mode over a VTP version 1 or VTP version 2 region.
- All network devices in a VTP domain must run the same VTP version.
- You must configure a password on each network device in the management domain when VTP is in secure mode.



Caution

If you configure VTP in secure mode and you do not assign a management domain password to each network device in the domain, the management domain does not function properly.

- A VTP version 2-capable network device can operate in the same VTP domain as a network device running VTP version 1 if VTP version 2 is disabled on the VTP version 2-capable network device (VTP version 2 is disabled by default).
- Do not enable VTP version 2 on a network device unless all of the network devices in the same VTP domain are version 2-capable. When you enable VTP version 2 on a server, all of the version 2-capable network devices in the domain enable VTP version 2.
- Enabling or disabling VTP pruning on a VTP server enables or disables VTP pruning for the entire management domain.

- Configuring VLANs as eligible for pruning on a Catalyst 4500 series switch affects pruning eligibility for those VLANs on that switch only, not on all network devices in the VTP domain.
- The VLAN database is saved in the NVRAM file in a format compliant with the VTP version running on the system. Since older images supporting only VTP version 2 do not recognize the VTP version 3 file format, the NVRAM VLAN database information is lost if the system is downgraded from a new image supporting VTP to one that does not.

VTP Default Configuration

Table 17-3 shows the default VTP configuration.

Table 17-3 VTP Default Configuration

Feature	Default Value
VTP domain name	Null
VTP mode	Server
VTP version 2 enable state	Version 2 is disabled
VTP password	None
VTP pruning	Disabled

The default VTP mode for newly manufactured Catalyst 4500 Series Switches supervisor engines is transparent. Deleting `vlan.dat` or entering the `erase cat4000_flash:` command, and resetting the switch changes the VTP mode to server.

Configuring VTP

These sections describe how to configure VTP:

- [Configuring VTP Global Parameters, page 17-13](#)
- [Configuring the VTP Mode, page 17-16](#)
- [Starting a Takeover, page 17-18](#)
- [Displaying VTP Statistics, page 17-19](#)
- [Displaying VTP Devices in a Domain, page 17-19](#)

Configuring VTP Global Parameters

These sections describe configuring the VTP global parameters:

- [Configuring a VTP Password, page 17-14](#)
- [Enabling VTP Pruning, page 17-15](#)
- [Enabling the VTP Version Number, page 17-15](#)



Note

You can enter the VTP global parameters in either global configuration mode or in EXEC mode.

Configuring a VTP Password

To configure the VTP global parameters, use these commands:

Command	Purpose
Switch(config)# vtp password <i>password_string</i> [hidden secret]	Sets a password, which can be from 8 to 64 characters long, for the VTP domain. In VTP version 3 the keywords hidden and secret are available. <ul style="list-style-type: none"> If the hidden keyword is used, the secret key generated from the password string is saved in the const_nvram:vlan.dat file. If a takeover command is issued, that password must be reentered. If the secret keyword is used, the password secret key can be directly configured. The secret password must contain 32 hexadecimal characters.
Switch(config)# no vtp password	Clears the password.

This example shows one way to configure a VTP password in global configuration mode:

```
Switch# configure terminal
Switch(config)# vtp password myPassword
Setting device VLAN database password to myPassword.
Switch#
```

This example shows how to configure a VTP password in EXEC mode:

```
Switch# vtp password myPassword
Setting device VLAN database password to myPassword.
Switch#
```



Note

The password is not stored in the running-config file.

This example shows how to configure a **hidden** password:

```
Switch# configure terminal
Switch(config)# vtp password myPassword hidden
Generating the secret associated to the password.
Switch(config)#
```

This example shows how the password myPassword is displayed when it is configured with the **hidden** keyword.

```
Switch# show vtp password
VTP Password: 89914640C8D90868B6A0D8103847A733
Switch#
```

Enabling VTP Pruning

To enable VTP pruning in the management domain, perform this task:

	Command	Purpose
Step 1	Switch(config)# vtp pruning	Enables VTP pruning in the management domain.
Step 2	Switch# show vtp status include pruning	(Optional) Verifies the configuration.

This example shows one way to enable VTP pruning in the management domain:

```
Switch# configure terminal
Switch(config)# vtp pruning
Pruning switched ON
```

This example shows how to enable VTP pruning in the management domain with any release:

```
Switch# vtp pruning
Pruning switched ON
```

This example shows how to verify the configuration:

```
Switch# show vtp status | include Pruning
VTP Pruning Mode: Enabled
Switch#
```

For information about configuring prune eligibility, see the [“Understanding VTP Pruning” section on page 17-10](#).

Enabling the VTP Version Number

VTP version 2 is disabled by default on VTP version-2-capable network devices. When you enable VTP version 2 on a network device, every VTP version-2-capable network device in the VTP domain enables version 2.



Caution

VTP version 1 and VTP version 2 are not interoperable on network devices in the same VTP domain. Every network device in the VTP domain must use the same VTP version. Do not enable VTP version 2 unless every network device in the VTP domain supports version 2.



Note

In a Token Ring environment, you must enable VTP version 2 or VTP version 3 for Token Ring VLAN switching to function properly on devices that support Token Ring interfaces.

To enable the VTP version, perform this task:

	Command	Purpose
Step 1	Switch(config)# vtp version {1 2 3}	Enables the VTP version.
Step 2	Switch# show vtp status include {v1 v2 v3}	(Optional) Verifies the configuration.

This example shows one way to enable VTP version 2:

```
Switch# configure terminal
Switch(config)# vtp version 2
V2 mode enabled.
Switch(config)#
```

This example shows how to enable VTP version 2 with any release:

```
Switch# vtp version 2
V2 mode enabled.
Switch#
```

This example shows how to verify the configuration:

```
Switch# show vtp status | include V2
VTP V2 Mode: Enabled
Switch#
```

Configuring the VTP Mode

To configure the VTP mode, perform this task:

	Command	Purpose
Step 1	Switch(config)# vtp mode {client server transparent off}	Configures the VTP mode.
Step 2	Switch(config)# vtp domain <i>domain_name</i>	(Optional; for server mode only) Defines the VTP domain name, which can be up to 32 characters long. VTP server mode requires a domain name. If the switch has a trunk connection to a VTP domain, the switch learns the domain name from the VTP server in the domain. Note You cannot clear the domain name.
Step 3	Switch(config)# end	Exits VLAN configuration mode.
Step 4	Switch# show vtp status	(Optional) Verifies the configuration.



Note

When VTP is disabled, you can enter VLAN configuration commands in configuration mode instead of the VLAN database mode and the VLAN configuration is stored in the startup configuration file.

This example shows how to configure the switch as a VTP server:

```
Switch# configure terminal
Switch(config)# vtp mode server
Setting device to VTP SERVER mode.
Switch(config)# vtp domain Lab_Network
Setting VTP domain name to Lab_Network
Switch(config)# end
Switch#
```

This example shows how to configure the switch as a VTP client:

```
Switch# configure terminal
Switch(config)# vtp mode client
Setting device to VTP CLIENT mode.
Switch(config)# end
Switch#
```

This example shows how to disable VTP on the switch:

```
Switch# configure terminal
Switch(config)# vtp mode transparent
Setting device to VTP TRANSPARENT mode.
Switch(config)# end
Switch#
```

This example shows how to disable VTP on the switch and to disable VTP advertisement forwarding:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# vtp mode off
Setting device to VTP OFF mode.
Switch(config)# end
Switch#
```

This example shows an example of the VTP configuration parameters when the device is running VTP version 1:

```
Switch# show vtp status
VTP Version capable           : 1 to 3
VTP version running          : 1
VTP Domain Name               : Lab_Network
VTP Pruning Mode              : Enabled
VTP Traps Generation          : Disabled
Device ID                     : 0016.9c6d.5300
Configuration last modified by 127.0.0.12 at 10-18-07 10:12:42
Local updater ID is 127.00.12 at 10-18-07 10:2:42

Feature VLAN:
-----
VTP Operating Mode            : Server
Maximum number of existing VLANs : 5
Configuration Revision        : 1
MD5 digest                    : 0x92 0xF1 0xE8 0x52 0x2E 0x5C 0x36 0x10 0x70 0x61 0xB8
                                0x24 0xB6 0x93 0x21 0x09

Switch#
```

This example shows an example of the VTP configuration parameters when the device is running VTP version 2:

```
Switch# show vtp status
VTP Version capable           : 1 to 3
VTP version running          : 2
VTP Domain Name               : Lab_Network
VTP Pruning Mode              : Disabled
VTP Traps Generation          : Disabled
Device ID                     : 0012.44dc.b800
Configuration last modified by 127.0.0.12 at 10-18-07 10:38:45
Local updater ID is 127.0.0.12 on interface EO 0/0 (first interface found)

Feature VLAN:
-----
VTP Operating Mode            : Server
Maximum VLANs supported locally: 1005
Number of existing VLANs      : 1005
Configuration Revision        : 1
MD5 digest                    : 0x2E 0x6B 0x99 0x58 0xA2 0x4F 0xD5 0x15 0x70 0x61 0xB8
                                0x24 0xB6 0x93 0x21 0x09

Switch#
```

This example shows an example of the VTP configuration parameters when the device is running VTP version 3:

```
Switch# show vtp status
VTP Version capable      : 1 to 3
VTP version running      : 3
VTP Domain Name          : Lab_Network
VTP Pruning Mode         : Disabled
VTP Traps Generation     : Disabled
Device ID                : 0012.44dc.b800

Feature VLAN:
-----
VTP Operating Mode       : Server
Number of existing VLANs : 1005
Number of existing extended VLANs: 3074
Configuration Revision   : 18
Primary ID               : 0012.4371.9ec0
Primary Description      :
Switch#
```

Starting a Takeover

This process applies to VTP version 3 only. To start a takeover, perform this task:

Command	Purpose
Switch# vtp primary-server [vlan mst] [force]	<p>Changes the operational state of a switch from a secondary to a primary server and advertises the configuration to the whole domain. (If the password for this device is configured with the hidden keyword, the user is prompted to re-enter it.)</p> <p>Note Using the force keyword overwrites the configuration of any conflicting servers. If not using the force keyword, you are prompted for confirmation before proceeding with the takeover.</p> <p>Specify where to direct the takeover by selecting the appropriate feature (vlan or mst). If no feature is selected, the takeover is directed to the VLAN database.</p>

This example shows how to start a takeover and direct it to the **vlan** database:

```
Switch# vtp primary-server vlan
Enter VTP password:password
This system is becoming primary for feature vlan

VTP Feature Conf Revision Primary Server Device ID      Description
-----
MST          Yes    4          0012.4371.9ec0=0012.4371.9ec0 R1
Do you want to continue? (confirm)
Switch#
```


Displaying VTP Statistics

To display VTP statistics, including VTP advertisements sent and received and VTP errors, perform this task:

Command	Purpose
Switch# show vtp counters	Displays VTP statistics.

This example shows how to display VTP statistics:

```
Switch# show vtp counters
VTP statistics:
Summary advertisements received      : 7
Subset advertisements received      : 5
Request advertisements received     : 0
Summary advertisements transmitted : 997
Subset advertisements transmitted   : 13
Request advertisements transmitted   : 3
Number of config revision errors    : 0
Number of config digest errors      : 0
Number of V1 summary errors         : 0

VTP pruning statistics:

Trunk          Join Transmitted Join Received      Summary advts received from
-----          -----          -----          non-pruning-capable device
Fa5/8          43071          42766          5
```

Displaying VTP Devices in a Domain

To display information for all the VTP devices in a domain, perform this task:

Command	Purpose
Switch# show vtp devices [conflicts]	<p>Gathers and displays information for all the VTP devices in the domain.</p> <p>Note No information is gathered or displayed from switches set to vtp modes off or to transparent for a particular feature.</p> <p>The conflicts keyword (optional) displays the information of devices that have conflicting primary servers.</p>

This example shows how to display information for VTP devices in a domain:

```
Switch# show vtp devices
Retrieving information from the VTP domain, please wait for 5 seconds.
VTP Feature Conf Revision Primary Server Device ID      Device Description
-----
VLAN          No    18      0016.9c6d.5300 0012.011a.0d00    R2
VLAN          No    18      0016.9c6d.5300 0012.4371.9ec0    R1
MST           Yes    4       0012.4371.9ec0=0012.4371.9ec0    R1

Switch#
```

VLAN Membership Policy Server

This section describes how to configure dynamic port VLAN membership through the VLAN Membership Policy Server (VMPS), and includes the following subsections:

- [About VMPS, page 17-20](#)
- [Overview of VMPS Clients, page 17-22](#)
- [Dynamic Port VLAN Membership Configuration Example, page 17-28](#)
- [VMPS Database Configuration File Example, page 17-31](#)

About VMPS

These subsections describe what a VMPS server does and how it operates:

- [Understanding the VMPS Server, page 17-20](#)
- [Security Modes for VMPS Server, page 17-21](#)
- [Fallback VLAN, page 17-22](#)
- [Illegal VMPS Client Requests, page 17-22](#)

Understanding the VMPS Server

A VLAN Membership Policy Server (VMPS) provides a centralized server for selecting the VLAN for a port dynamically based on the MAC address of the device connected to the port. When the host moves from a port on one switch in the network to a port on another switch in the network, that switch dynamically assigns the new port to the proper VLAN for that host.

A Catalyst 4500 series switch running Cisco IOS software does not support the functionality of a VMPS. It can only function as a VLAN Query Protocol (VQP) client, which communicates with a VMPS through the VQP. For VMPS functionality, you need to use a Catalyst 4500 series switch (or Catalyst 6500 series switch) running Catalyst operating system (OS) software.

VMPS uses a UDP port to listen to VQP requests from clients, so, it is not necessary for VMPS clients to know if the VMPS resides on a local or remote device on the network. Upon receiving a valid request from a VMPS client, a VMPS server searches its database for an entry of a MAC-address to VLAN mapping.

In response to a request, the VMPS takes one of the following actions:

- If the assigned VLAN is restricted to a group of ports, the VMPS verifies the requesting port against this group and responds as follows:

- If the VLAN is allowed on the port, the VMPS sends the VLAN name to the client in response.
- If the VLAN is not allowed on the port and the VMPS is not in secure mode, the VMPS sends an “access-denied” response.
- If the VLAN is not allowed on the port and the VMPS is in secure mode, the VMPS sends a “port-shutdown” response.
- If the VLAN in the database does not match the current VLAN on the port and active hosts exist on the port, the VMPS sends an “access-denied” (open), a “fallback VLAN name” (open with fallback VLAN configured), a “port-shutdown” (secure), or a “new VLAN name” (multiple) response, depending on the secure mode setting of the VMPS.

If the switch receives an “access-denied” response from the VMPS, the switch continues to block traffic from the MAC address to or from the port. The switch continues to monitor the packets directed to the port and sends a query to the VMPS when it identifies a new address. If the switch receives a “port-shutdown” response from the VMPS, the switch disables the port. The port must be manually reenabled by using the CLI, Cisco Visual Switch Manager (CVSM), or SNMP.

You can also use an explicit entry in the configuration table to deny access to specific MAC addresses for security reasons. If you enter the **none** keyword for the VLAN name, the VMPS sends an “access-denied” or “port-shutdown” response.

Security Modes for VMPS Server

VMPS operates in three different modes. The way a VMPS server responds to illegal requests depends on the mode in which the VMPS is configured:

- [Open Mode, page 17-21](#)
- [Secure Mode, page 17-21](#)
- [Multiple Mode, page 17-22](#)

Open Mode

If no VLAN is assigned to this port, VMPS verifies the requesting MAC address against this port:

- If the VLAN associated with this MAC address is allowed on the port, the VLAN name is returned to the client.
- If the VLAN associated with this MAC address is not allowed on the port, the host receives an “access denied” response.

If a VLAN is already assigned to this port, VMPS verifies the requesting MAC address against this port:

- If the VLAN associated with this MAC address in the database does not match the current VLAN assigned on the port, and a fallback VLAN name is configured, VMPS sends the fallback VLAN name to the client.
- If a VLAN associated with this MAC address in the database does not match the current VLAN assigned on the port, and a fallback VLAN name is not configured, the host receives an “access denied” response.

Secure Mode

If no VLAN is assigned to this port, VMPS verifies the requesting MAC address against this port:

- If the VLAN associated with this MAC address is allowed on the port, the VLAN name is returned to the client.

- If the VLAN associated with this MAC address is not allowed on the port, the port is shut down.

If a VLAN is already assigned to this port, VMPS verifies the requesting MAC address against this port:

- If a VLAN associated with this MAC address in the database does not match the current VLAN assigned on the port, the port is shutdown, even if a fallback VLAN name is configured.

Multiple Mode

Multiple hosts (MAC addresses) can be active on a dynamic port if they are all in the same VLAN. If the link fails on a dynamic port, the port returns to the unassigned state. Any hosts that come online through the port are checked again with VMPS before the port is assigned to a VLAN.

If multiple hosts connected to a dynamic port belong to different VLANs, the VLAN matching the MAC address in the last request is returned to the client provided that multiple mode is configured on the VMPS server.



Note

Although Catalyst 4500 series and Catalyst 6500 series switches running Catalyst operating system software support VMPS in all three operation modes, the User Registration Tool (URT) supports open mode only.

Fallback VLAN

You can configure a fallback VLAN name on a VMPS server.

If no VLAN has been assigned to this port, VMPS compares the requesting MAC address to this port:

- If you connect a device with a MAC address that is not in the database, the VMPS sends the fallback VLAN name to the client.
- If you do not configure a fallback VLAN name and the MAC address does not exist in the database, the VMPS sends an “access-denied” response.

If a VLAN is already assigned to this port, VMPS compares the requesting MAC address to this port:

- If the VMPS is in secure mode, it sends a “port-shutdown” response, whether a fallback VLAN has been configured on the server.

Illegal VMPS Client Requests

Two examples of illegal VMPS client requests are as follows:

- When a MAC-address mapping is not present in the VMPS database and “no fall back” VLAN is configured on the VMPS.
- When a port is already assigned a VLAN (and the VMPS mode is not “multiple”) but a second VMPS client request is received on the VMPS for a different MAC-address.

Overview of VMPS Clients

The following subsections describe how to configure a switch as a VMPS client and configure its ports for dynamic VLAN membership.

The following topics are included:

- [Understanding Dynamic VLAN Membership, page 17-23](#)

- [Default VMPS Client Configuration, page 17-23](#)
- [Configuring a Switch as a VMPS Client, page 17-24](#)
- [Administering and Monitoring the VMPS, page 17-27](#)
- [Troubleshooting Dynamic Port VLAN Membership, page 17-28](#)

Understanding Dynamic VLAN Membership

When a port is configured as “dynamic,” it receives VLAN information based on the MAC-address that is on the port. The VLAN is not statically assigned to the port; it is dynamically acquired from the VMPS based on the MAC-address on the port.

A dynamic port can belong to one VLAN only. When the link becomes active, the switch does not forward traffic to or from this port until the port is assigned to a VLAN. The source MAC address from the first packet of a new host on the dynamic port is sent to the VMPS as part of the VQP request, which attempts to match the MAC address to a VLAN in the VMPS database. If there is a match, the VMPS sends the VLAN number for that port. If there is no match, the VMPS either denies the request or shuts down the port (depending on the VMPS security mode setting). See the [“About VMPS” section on page 17-20](#) for a complete description of possible VMPS responses.

Multiple hosts (MAC addresses) can be active on a dynamic port if all are in the same VLAN. If the link goes down on a dynamic port, the port returns to the unassigned state and does not belong to a VLAN. Any hosts that come online through the port are checked again with the VMPS before the port is assigned to a VLAN.

For this operation to work, the client device must be able to reach the VMPS. A VMPS client sends VQP requests as UDP packets, trying a certain number of times before giving up. For details on how to set the retry interval, refer to section “Configuring the Retry Interval” on page 26.

The VMPS client also periodically reconfirms the VLAN membership. For details on how to set the reconfirm frequency, refer to section “Administering and Monitoring the VMPS” on page 27.

A maximum of 50 hosts are supported on a given port at any given time. Once this maximum is exceeded, the port is shut down, irrespective of the operating mode of the VMPS server.

**Note**

The VMPS shuts down a dynamic port if more than 50 hosts are active on that port.

Default VMPS Client Configuration

[Table 17-4](#) shows the default VMPS and dynamic port configuration on client switches.

Table 17-4 **Default VMPS Client and Dynamic Port Configuration**

Feature	Default Configuration
VMPS domain server	None
VMPS reconfirm interval	60 minutes
VMPS server retry count	3
Dynamic ports	None configured

Configuring a Switch as a VMPS Client

This section contains the following topics:

- [Configuring the IP Address of the VMPS Server, page 17-24](#)
- [Configuring Dynamic Access Ports on a VMPS Client, page 17-24](#)
- [Reconfirming VLAN Memberships, page 17-25](#)
- [Configuring Reconfirmation Interval, page 17-26](#)
- [Reconfirming VLAN Memberships, page 17-25](#)

Configuring the IP Address of the VMPS Server

To configure a Catalyst 4500 series switch as a VMPS client, you must enter the IP address or hostname of the switch acting as the VMPS.

To define the primary and secondary VMPS on a Catalyst 4500 series switch, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# vmps server {ipaddress hostname} primary	Specifies the IP address or hostname of the switch acting as the primary VMPS server.
Step 3	Switch(config)# vmps server {ipaddress hostname}	Specifies the IP address or hostname of the switch acting as a secondary VMPS server.
Step 4	Switch(config)# end	Returns to privileged EXEC mode.
Step 5	Switch# show vmps	Verifies the VMPS server entry.

This example shows how to define the primary and secondary VMPS devices:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# vmps server 172.20.128.179 primary
Switch(config)# vmps server 172.20.128.178
Switch(config)# end
```



Note

You can configure up to four VMPS servers using this CLI on the VMPS client.

```
Switch# show vmps
VQP Client Status:
-----
VMPS VQP Version: 1
Reconfirm Interval: 60 min
Server Retry Count: 3
VMPS domain server: 172.20.128.179 (primary, current)
                   172.20.128.178

Reconfirmation status
-----
VMPS Action:          No Dynamic Port
```

Configuring Dynamic Access Ports on a VMPS Client

To configure a dynamic access port on a VMPS client switch, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface interface	Enters interface configuration mode and specifies the port to be configured.
Step 3	Switch(config-if)# switchport mode access	Sets the port to access mode.
Step 4	Switch(config-if)# switchport access vlan dynamic	Configures the port as eligible for dynamic VLAN access.
Step 5	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 6	Switch# show interface interface switchport	Verifies the entry.

This example shows how to configure a dynamic access port and to verify the entry:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface fa1/1
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan dynamic
Switch(config-if)# end

Switch# show interface fa1/1 switchport
Name: Fa0/1
Switchport: Enabled
Administrative mode: dynamic auto
Operational Mode: dynamic access
Administrative Trunking Encapsulation: isl
Operational Trunking Encapsulation: isl
Negotiation of Trunking: Disabled
Access Mode VLAN: 0 ((Inactive))
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: NONE
Pruning VLANs Enabled: NONE
```

Voice Ports

If a VVID (voice VLAN ID) is configured on a dynamic access port, the port can belong to both an access VLAN and a voice VLAN. Consequently, an access port configured for connecting an IP phone can have separate VLANs for the following:

- Data traffic to and from the PC that is connected to the switch through the access port of the IP phone (access VLAN)
- Voice traffic to and from the IP phone (voice VLAN)

Reconfirming VLAN Memberships

To confirm the dynamic port VLAN membership assignments that the switch has received from the VMPS, perform this task:

	Command	Purpose
Step 1	Switch# vmps reconfirm	Reconfirms dynamic port VLAN membership.
Step 2	Switch# show vmps	Verifies the dynamic VLAN reconfirmation status.

Configuring Reconfirmation Interval

VMPS clients periodically reconfirm the VLAN membership information received from the VMPS. You can set the number of minutes the VMPS client waits before reconfirming the VLAN-to-MAC-address assignments.

To configure the reconfirmation interval, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# vmips reconfirm <i>minutes</i>	Specifies the number of minutes between reconfirmations of the dynamic VLAN membership.
Step 3	Switch(config)# end	Returns to privileged EXEC mode.
Step 4	Switch# show vmips	Verifies the dynamic VLAN reconfirmation status.

This example shows how to change the reconfirmation interval to 60 minutes and verify the change:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# vmips reconfirm 60
Switch(config)# end
Switch# show vmips
VQP Client Status:
-----
VMPS VQP Version: 1
Reconfirm Interval: 60 min
Server Retry Count: 10
VMPS domain server: 172.20.130.50 (primary, current)

Reconfirmation status
-----
VMPS Action: No Host
```

Configuring the Retry Interval

You can set the number of times that the VMPS client attempts to contact the VMPS before querying the next server.

To configure the retry interval, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# vmips retry <i>count</i>	Specifies the retry count for the VPQ queries. Default is 3. Range is from 1 to 10.
Step 3	Switch(config)# end	Returns to privileged EXEC mode.
Step 4	Switch# show vmips	Verifies the retry count.

This example shows how to change the retry count to 5 and to verify the change:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# vmips retry 5
Switch(config)# end
```



```

Switch# show vmps
VQP Client Status:
-----
VMPS VQP Version: 1
Reconfirm Interval: 60 min
Server Retry Count: 5
VMPS domain server: 172.20.130.50 (primary, current)

Reconfirmation status
-----
VMPS Action:          No Host

```

Administering and Monitoring the VMPS

You can display the following information about the VMPS with the **show vmps** command:

VQP Version	The version of VQP used to communicate with the VMPS. The switch queries the VMPS using VQP Version 1.
Reconfirm Interval	The number of minutes the switch waits before reconfirming the VLAN-to-MAC-address assignments.
Server Retry Count	The number of times VQP resends a query to the VMPS. If no response is received after this many tries, the switch starts to query the secondary VMPS.
VMPS Domain Server	The IP address of the configured VLAN membership policy servers. The switch currently sends queries to the one marked “current.” The one marked “primary” is the primary server.
VMPS Action	The result of the most-recent reconfirmation attempt. This action can occur automatically when the reconfirmation interval expired, or you can force it by entering the vmps reconfirm command or its CVSM or SNMP equivalent.

The following example shows how to display VMPS information:

```

Switch# show vmps
VQP Client Status:
-----
VMPS VQP Version: 1
Reconfirm Interval: 60 min
Server Retry Count: 3
VMPS domain server:

Reconfirmation status
-----
VMPS Action:          other

```

The following example shows how to display VMPS statistics:

```

Switch# show vmps statistics
VMPS Client Statistics
-----
VQP  Queries:          0
VQP  Responses:        0
VMPS  Changes:          0
VQP  Shutdowns:        0
VQP  Denied:           0

```

```
VQP Wrong Domain:          0
VQP Wrong Version:         0
VQP Insufficient Resource: 0
```

**Note**

Refer to command reference guide for details on VMPS statistics.

Troubleshooting Dynamic Port VLAN Membership

VMPS errdisables a dynamic port under the following conditions:

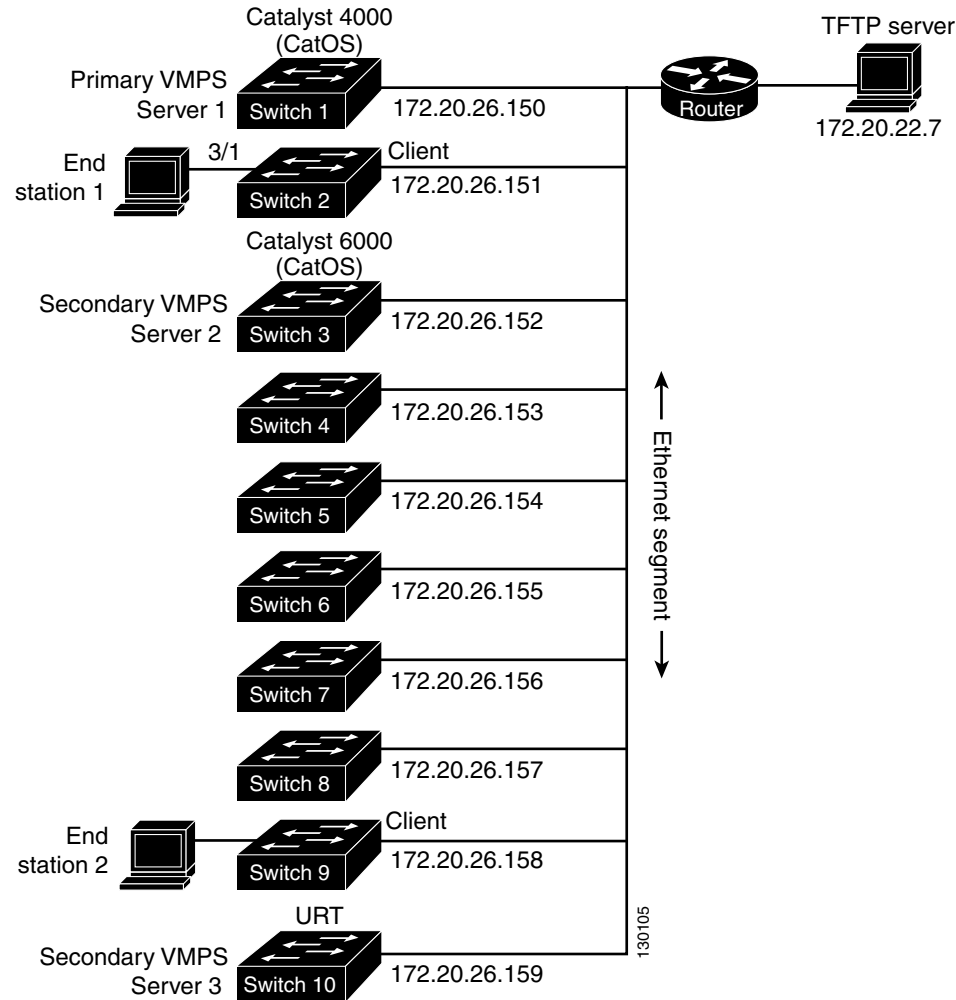
- The VMPS is in secure mode, and it does not allow the host to connect to the port. The VMPS errdisables the port to prevent the host from connecting to the network.
- More than 50 active hosts reside on a dynamic port.

For information on how to display the status of interfaces in error-disabled state, refer to [Chapter 10, “Checking Port Status and Connectivity.”](#) To recover an errdisabled port, use the **errdisable recovery cause vmps** global configuration command.

Dynamic Port VLAN Membership Configuration Example

[Figure 17-4 on page 17-29](#) shows a network with a VMPS servers and VMPS client switches with dynamic ports. In this example, these assumptions apply:

- The VMPS server and the VMPS client are separate switches.
- The Catalyst 4000 family Switch 1 (running Catalyst Operating System) is the primary VMPS server.
- The Catalyst 6000 family Switch 3 (running Catalyst Operating System) and the URT are secondary VMPS servers.
- End stations are connected to these clients:
 - Catalyst 4500 series XL Switch 2 (running Catalyst Cisco IOS)
 - Catalyst 4500 series XL Switch 9 (running Catalyst Cisco IOS)
- The database configuration file is called Bldg-G.db and is stored on the TFTP server with the IP address 172.20.22.7.

Figure 17-4 *Dynamic Port VLAN Membership Configuration*

Two topologies are possible. Figure 17-5 illustrates a topology with one end station attached directly to a Catalyst 4500 series switch operating as a VMPS client. Figure 17-6 illustrates a topology with an end station attached to a Cisco IP Phone, which is attached to a Catalyst 4500 series switch.

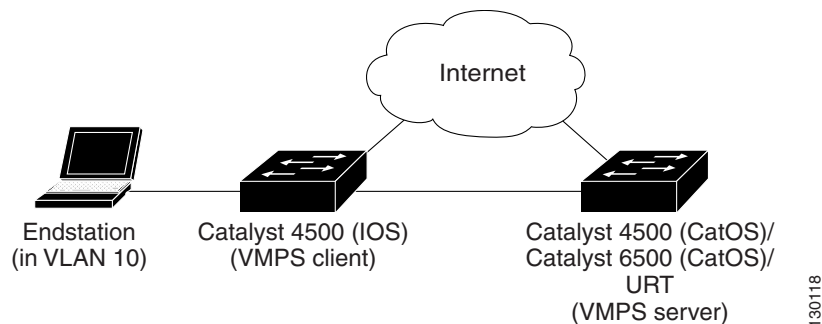
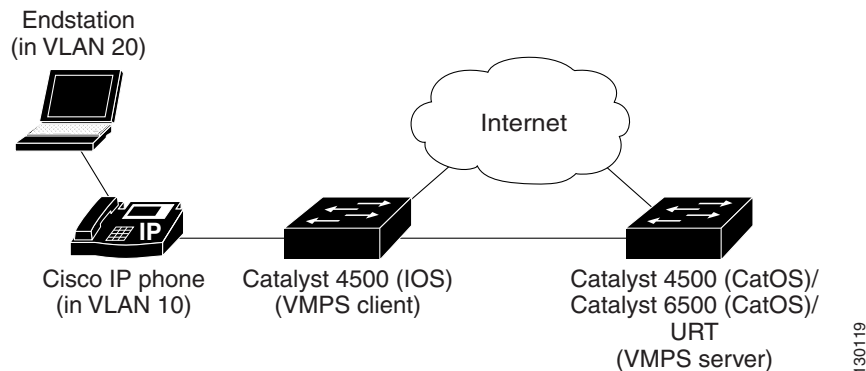
Figure 17-5 *Topology with an End Station Attached Directly to a Catalyst 4500 Series Switch Operating as a VMPS Client*

Figure 17-6 *Topology with an End Station Attached to a Cisco IP Phone that is Attached to a Catalyst 4500 Series Switch*



In the following procedure, the Catalyst 4500 and Catalyst 6500 series switches (running Catalyst Operating System) are the VMPS servers. Use this procedure to configure the Catalyst 4500 series switch clients in the network:

Step 1 Configure the VMPS server addresses on Switch 2, the client switch.

- a. Starting from privileged EXEC mode, enter global configuration mode:

```
switch# configuration terminal
```

- b. Enter the primary VMPS server IP address:

```
switch(config)# vmps server 172.20.26.150 primary
```

- c. Enter the secondary VMPS server IP addresses:

```
switch(config)# vmps server 172.20.26.152
```

- d. To verify your entry of the VMPS IP addresses, return to privileged EXEC mode:

```
switch(config)# exit
```

- e. Display VMPS information configured for the switch:

```
switch# show vmps
VQP Client Status:
-----
VMPS VQP Version: 1
Reconfirm Interval: 60 min
Server Retry Count: 3
VMPS domain server: 172.20.26.152
                    172.20.26.150 (primary, current)
```

Step 2 Configure port Fa0/1 on Switch 2 as a dynamic port.

- a. Return to global configuration mode:

```
switch# configure terminal
```

- b. Enter interface configuration mode:

```
switch(config)# interface fa2/1
```

- c. Configure the VLAN membership mode for static-access ports:

```
switch(config-if)# switchport mode access
```

- d. Assign the port dynamic VLAN membership:

```
switch(config-if)# switchport access vlan dynamic
```

- e. Return to privileged EXEC mode:

```
switch(config-if)# exit
switch#
```

Step 3 Connect End Station 2 on port Fa2/1. When End Station 2 sends a packet, Switch 2 sends a query to the primary VMPS server, Switch 1. Switch 1 responds with the VLAN ID for port Fa2/1. If spanning-tree PortFast mode is enabled on Fa2/1, port Fa2/1 connects immediately and begins forwarding.

Step 4 Set the VMPS reconfirmation period to 60 minutes. The reconfirmation period is the number of minutes the switch waits before reconfirming the VLAN to MAC address assignments.

```
switch# config terminal
switch(config)# vmps reconfirm 60
```

Step 5 Confirm the entry from privileged EXEC mode:

```
switch# show vmps
VQP Client Status:
-----
VMPS VQP Version: 1
Reconfirm Interval: 60 min
Server Retry Count: 3
VMPS domain server:

Reconfirmation status
-----
VMPS Action: No Dynamic Port
```

Step 6 Repeat Steps 1 and 2 to configure the VMPS server addresses, and assign dynamic ports on each VMPS client switch.

VMPS Database Configuration File Example

This example shows a sample VMPS database configuration file as it appears on a VMPS server. A VMPS database configuration file is an ASCII text file that is stored on a TFTP server accessible to the switch that functions as the VMPS server.

```
!vmips domain <domain-name>
! The VMPS domain must be defined.
!vmips mode {open | secure}
! The default mode is open.
!vmips fallback <vlan-name>
!vmips no-domain-req {allow | deny}
!
! The default value is allow.
vmips domain WBU
vmips mode open
vmips fallback default
vmips no-domain-req deny
!
!
```

```

!MAC Addresses
!
vmmps-mac-addr
!
! address <addr> vlan-name <vlan_name>
!
address 0012.2233.4455 vlan-name hardware
address 0000.6509.a080 vlan-name hardware
address aabb.ccdd.eeff vlan-name Green
address 1223.5678.9abc vlan-name ExecStaff
address fedc.ba98.7654 vlan-name --NONE--
address fedc.ba23.1245 vlan-name Purple
!
!Port Groups
!
!vmmps-port-group <group-name>
! device <device-id> {port <port-name> | all-ports}
!
vmmps-port-group WiringCloset1
  device 198.92.30.32 port Fa1/3
  device 172.20.26.141 port Fa1/4
vmmps-port-group "Executive Row"
  device 198.4.254.222 port es5%Fa0/1
  device 198.4.254.222 port es5%Fa0/2
  device 198.4.254.223 all-ports
!
!VLAN groups
!
!vmmps-vlan-group <group-name>
! vlan-name <vlan-name>
!
vmmps-vlan-group Engineering
vlan-name hardware
vlan-name software
!
!VLAN port Policies
!
!vmmps-port-policies {vlan-name <vlan_name> | vlan-group <group-name>}
! {port-group <group-name> | device <device-id> port <port-name>}
!
vmmps-port-policies vlan-group Engineering
  port-group WiringCloset1
vmmps-port-policies vlan-name Green
  device 198.92.30.32 port Fa0/9
vmmps-port-policies vlan-name Purple
  device 198.4.254.22 port Fa0/10
  port-group "Executive Row"

```



Configuring IP Unnumbered Interface

This chapter discusses the IP Unnumbered Interface feature, which allows you to enable IP processing on an interface without assigning an explicit IP address.

This chapter contains these sections:

- [About IP Unnumbered Interface Support, page 18-1](#)
- [IP Unnumbered Configuration Guidelines and Restrictions, page 18-3](#)
- [Configuring IP Unnumbered Interface Support with DHCP Server, page 18-4](#)
- [Configuring IP Unnumbered Interface Support with Connected Host Polling, page 18-6](#)
- [Displaying IP Unnumbered Interface Settings, page 18-7](#)
- [Troubleshooting IP Unnumbered Interface, page 18-8](#)
- [Related Documents, page 18-8](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About IP Unnumbered Interface Support

Before you configure VLANs and LAN interfaces with IP unnumbered interfaces, you should understand the following concepts:

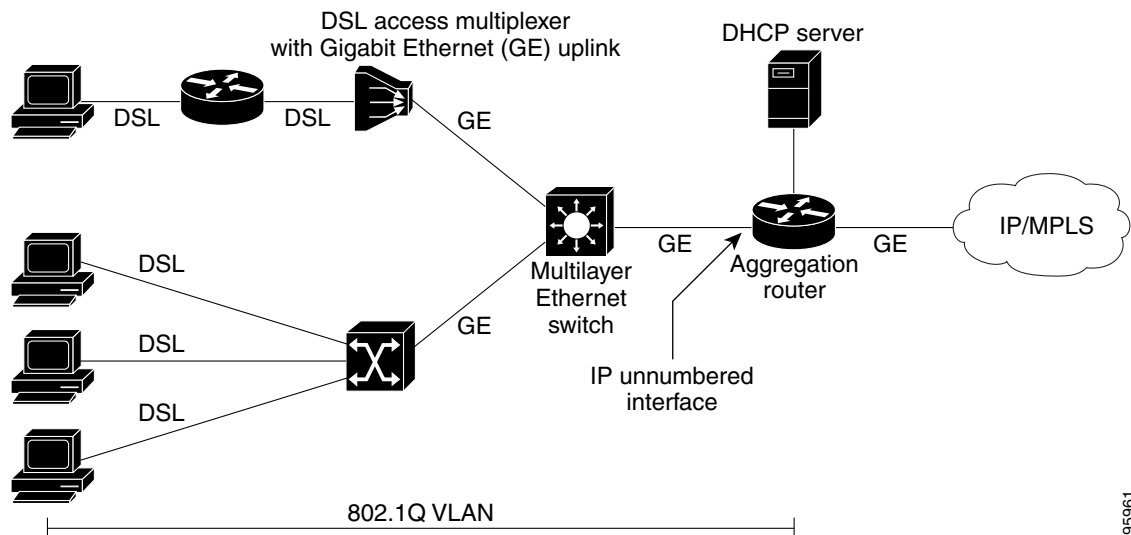
- [IP Unnumbered Interface Support with DHCP Server and Relay Agent, page 18-2](#)
- [DHCP Option 82, page 18-2](#)
- [IP Unnumbered Interface with Connected Host Polling, page 18-3](#)

IP Unnumbered Interface Support with DHCP Server and Relay Agent

The IP unnumbered interface configuration allows you to enable IP processing on an interface without assigning it an explicit IP address. The IP unnumbered interface can “borrow” the IP address from another interface that is already configured on the Catalyst 4500 series switch, which conserves network and address space. When used with the DHCP server/relay agent, this feature allows a host address assigned by the DHCP server to be learned dynamically at the DHCP relay agent.

Figure 18-1 shows a sample network topology implementing the IP Unnumbered Interface feature. In this topology, IP routes are dynamically established by the aggregation switch when the DHCP server assigns IP addresses to the hosts.

Figure 18-1 Sample Network Topology Using the VLANs over IP Unnumbered Interfaces Feature



95961

DHCP Option 82

DHCP provides a framework for passing configuration information to hosts on a TCP/IP network. Configuration parameters and other control information are carried in tagged data items that are stored in the options field of the DHCP message. The data items are also called *options*. Option 82 is organized as a single DHCP option that contains information known by the relay agent.

The IP Unnumbered Interface feature communicates information to the DHCP server using a suboption of the DHCP relay agent information option called *agent remote ID*. The information sent in the agent remote ID includes an IP address identifying the relay agent and information about the interface and the connection over which the DHCP request entered. The DHCP server can use this information to make IP address assignments and security policy decisions.

Figure 18-2 shows the agent remote ID suboption format that is used with the IP Unnumbered Interfaces feature.

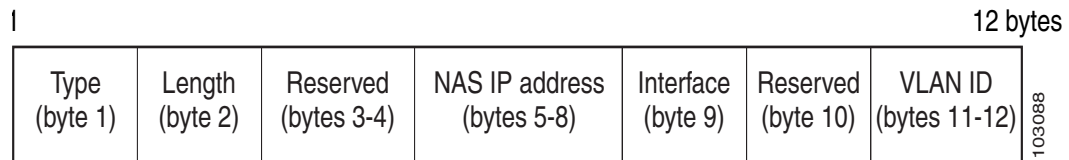
Figure 18-2 **Format of the Agent Remote ID Suboption**

Table 18-1 describes the agent remote ID suboption fields displayed in Figure 18-2.

Table 18-1 **Agent Remote ID Suboption Field Descriptions**

Field	Description
Type	Format type. The value 2 specifies the format for use with this feature. (1 byte)
Length	Length of the Agent Remote ID suboption, not including the type and length fields. (1 byte)
Reserved	Reserved. (2 bytes)
NAS IP Address	IP address of the interface specified by the ip unnumbered command. (4 bytes)
Interface	Physical interface. This field has the following format: slot (4 bits) module (1 bit) port (3 bits). For example, if the interface name is interface ethernet 2/1/1, the slot is 2, the module is 1, and the port is 1. (1 byte)
Reserved	Reserved. (1 byte)
VLAN ID	VLAN identifier for the Ethernet interface. (2 bytes)

IP Unnumbered Interface with Connected Host Polling



Note

This feature option is applicable to LAN and VLAN interfaces only.

In some cases, the host IP address is assigned statically. The IP Unnumbered Interfaces feature can learn the static host IP address dynamically.

IP Unnumbered Configuration Guidelines and Restrictions

When using (or configuring) IP Unnumbered Interface, consider these guidelines and restrictions:

- For IP Unnumbered Interfaces, the following features are not supported:
 - Dynamic routing protocols
 - HSRP/VRRP
 - Static ARP
 - Unnumbered Interface and Numbered Interface in different VRFs

- The option to add *dhcp host routes* as connected routes is available in Cisco IOS. When using connected mode, however, the **clear ip route *** command deletes the dhcp host connected routes permanently.

Workarounds:

- For a layer 3 interface (SVI), enter **shut** then **no shut**.
- To enable IP unnumbered to use static routes, enter the **ip dhcp route static** command.
- IP Redirect is not sent by an interface configured with IP Unnumbered Interface.
- IP Unnumbered Interface is unable to forward multicast source packets.

Configuring IP Unnumbered Interface Support with DHCP Server



Note

DHCP must be configured and operational before you perform this task.

This section contains the following procedures:

- [Configuring IP Unnumbered Interface Support on LAN and VLAN Interfaces, page 18-4](#)
- [Configuring IP Unnumbered Interface Support on a Range of Ethernet VLANs, page 18-5](#)

Configuring IP Unnumbered Interface Support on LAN and VLAN Interfaces

To configure IP unnumbered interface support on a single LAN or VLAN interface, perform this task.

	Command	Purpose
Step 1	Switch# enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	Switch# configure terminal	Enters global configuration mode.
Step 3	Switch(config)# interface [<i>fastethernet</i> <i>gigabitethernet</i> <i>tengigabitethernet</i> <i>vlan</i> <i>vlan</i> <i>port-channel</i> <i>loopback</i>]	Enters interface configuration mode and the interface to be configured as a tunnel port.
Step 4	Switch(config-if)# ip unnumbered <i>type number</i>	Enables IP processing on an interface without assigning an explicit IP address to the interface. The <i>type</i> and <i>number</i> arguments specify another interface on which the switch has an assigned IP address. The interface specified cannot be another unnumbered interface.
Step 5	Switch(config-if)# exit	Returns to global configuration mode.
Step 6	Switch(config)# end	Returns to privileged EXEC mode.
Step 7	Switch# show running-config	Verifies that IP unnumbered support has been configured correctly.

In the following example, Ethernet VLAN 10 is configured as an IP unnumbered interfaces:

```
Switch> enable
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface vlan 10
Switch(config-if)# ip unnumbered Loopback 0
```

Configuring IP Unnumbered Interface Support on a Range of Ethernet VLANs

To configure IP unnumbered interface support on a range of Ethernet VLAN interfaces, perform this task:

	Command or Action	Purpose
Step 1	Switch# enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	Switch# configure terminal	Enters global configuration mode.
Step 3	Switch(config)# interface range {{ fastethernet gigabitethernet vlan <i>vlan</i> } <i>slot/interface</i> { fastethernet gigabitethernet vlan <i>vlan</i> } <i>slot/interface</i> macro <i>macro-name</i> }	Executes commands on multiple interfaces at the same time. A hyphen must be entered with a space on either side to separate the range information.
Step 4	Switch(config-if)# ip unnumbered <i>type number</i>	Enables IP processing on an interface without assigning an explicit IP address to the interface. The <i>type</i> and <i>number</i> arguments specify another interface on which the switch has an assigned IP address. The specified interface cannot be another unnumbered interface.
Step 5	Switch(config-if)# exit	Returns to global configuration mode.
Step 6	Switch(config)# end	Returns to privileged EXEC mode.
Step 7	Switch# show running-config	Verifies that IP unnumbered support has been configured correctly.

In the following example, VLANs in the range from 1 to 10 are configured as IP unnumbered interfaces, sharing IP address of Fast Ethernet 3/1:

```
Switch> enable
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface range vlan 1 - 10
Switch(config-if)# ip unnumbered fastethernet 3/1
Switch(config-if)# exit
Switch(config)# end
```

Configuring IP Unnumbered Interface Support with Connected Host Polling

To configure IP unnumbered interface support with connected host polling, perform this task:

	Command	Purpose
Step 1	Switch# enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	Switch# configure terminal	Enters global configuration mode.
Step 3	Switch(config)# interface vlan <i>vlan-id</i>	Enters interface configuration mode and the interface to be configured as a tunnel port.
Step 4	Switch(config-if)# ip unnumbered <i>type number</i> poll	Enables IP processing and connected host polling on an interface without assigning an explicit IP address to the interface <i>type</i> and <i>number</i> specify another interface on which the switch has an assigned IP address. The interface specified cannot be another unnumbered interface. The <i>type</i> argument can have the values: <i>loopback</i> , <i>fastethernet</i> , <i>gigabitethernet</i> , <i>svi</i> , and <i>portchannel</i> .
Step 5	Switch(config-if)# exit	Returns to global configuration mode.
Step 6	Switch(config)# ip arp poll queue <10-10000>	Configures the global backlog queue of host addresses to be discovered. Default for the queue size is 1000.
Step 7	Switch(config)# ip arp poll rate <10-10000>	Configures the maximum number of ARP requests sent over unnumbered interfaces. Default number of ARP requests is 1000 packet per second.
Step 8	Switch(config)# end	Returns to privileged EXEC mode.
Step 9	Switch# show running-config	Verifies that IP unnumbered support has been configured correctly.

The following example shows how to enable IP processing and connected host polling on Fast Ethernet interface 6/2. It also shows how to set the global backlog queue to 2000 and the maximum number of ARP requests to 500:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface fastEthernet 6/2
Switch(config-if)# no switchport
Switch(config-if)# ip unnumbered loopback 0 poll
Warning: dynamic routing protocols will not work on non-point-to-point interfaces with IP
unnumbered configured.
Switch(config-if)# exit
Switch(config)# ip arp poll queue 2000
Switch(config)# ip arp poll rate 500
Switch(config)# end
```

Displaying IP Unnumbered Interface Settings

Use the **show ip interface unnumbered** command to display status of an unnumbered interface with connected host polling for the switch.

To display **status of an unnumbered interface**, enter this command:

Command	Purpose
Switch# show ip interface [type number] unnumbered [detail]	Displays the status of unnumbered interface with connected host polling for the Catalyst 4500 series switch.

The following example shows how to display the status of unnumbered interfaces with connected host polling:

```
Switch# show ip interface loopback 0 unnumbered detail
Number of unnumbered interfaces with polling: 1
Number of IP addresses processed for polling: 2
10.1.1.7
10.1.1.8
Number of IP addresses in queue for polling: 2 (high water mark: 3)
10.1.1.17
10.1.1.18
```

To display key statistic for the backlog of unnumbered interfaces with connected host polling for the switch, perform this task:

Command	Purpose
Switch# show ip arp poll [detail]	Displays key statistic for the backlog of unnumbered interfaces with connected host polling for the switch.

The following example shows how to display key statistic for the backlog of unnumbered interfaces with connected host polling:

```
Switch# show ip arp poll
Number of IP addresses processed for polling: 439
Number of IP addresses in queue for polling: 3 (high water mark: 0, max: 1000)
Number of requests dropped:
  Queue was full: 0
  Request was throttled by incomplete ARP: 0
  Duplicate request was found in queue: 0
```

To clear the key statistic for the backlog of unnumbered interfaces, use the **clear ip arp poll statistic** command, as follows:

```
Switch# clear ip arp poll statistic
Switch# show ip arp poll
Number of IP addresses processed for polling: 0
Number of IP addresses in queue for polling: 0 (high water mark: 0, max: 1000)
Number of requests dropped:
  Queue was full: 0
  Request was throttled by incomplete ARP: 0
  Duplicate request was found in queue: 0
```

Troubleshooting IP Unnumbered Interface

To understand how to debug connect host polling, see the Cisco IOS documentation of the **debug arp** command on cisco.com.

When an IP unnumbered interface shares the IP address of a loopback interface whose prefix is advertised in an OSPF network, you must modify the loopback interface as a point-to-point interface. Otherwise, only the loopback interface host route is advertised to an OSPF neighbor.

```
Switch(config)# int loopback 0
Switch(config-if)# ip address
Switch(config-if)# ip address 10.1.0.1 255.255.0.0
Switch(config-if)# ip ospf network point-to-point
Switch(config-if)# end
```

Related Documents

Related Topic	Document Title
DHCP and other IP addressing configuration tasks	“IP Addressing and Services” section of the <i>Cisco IOS IP Addressing Services Configuration Guide, Release 12.4</i>
DHCP and other IP addressing commands	<i>Cisco IOS IP Addressing Services Command Reference, Release 12.4 T</i>
VLAN configuration tasks	“Virtual LANs” chapter of the <i>Cisco IOS LAN Switching Configuration Guide, Release 12.4</i>
VLAN configuration commands	<i>Cisco IOS LAN Switching Command Reference, Release 12.4 T</i>



Configuring Layer 2 Ethernet Interfaces

This chapter describes how to use the command-line interface (CLI) to configure Fast Ethernet and Gigabit Ethernet interfaces for Layer 2 switching on Catalyst 4500 series switches. It also provides guidelines, procedures, and configuration examples. The configuration tasks in this chapter apply to Fast Ethernet and Gigabit Ethernet interfaces on any module, including the uplink ports on the supervisor engine.

This chapter includes the following major sections:

- [About Layer 2 Ethernet Switching, page 19-1](#)
- [Default Layer 2 Ethernet Interface Configuration, page 19-4](#)
- [Layer 2 Interface Configuration Guidelines and Restrictions, page 19-4](#)
- [Configuring Ethernet Interfaces for Layer 2 Switching, page 19-5](#)



Note

To configure Layer 3 interfaces, see [Chapter 36, “Configuring Layer 3 Interfaces.”](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About Layer 2 Ethernet Switching

The following sections describe how Layer 2 Ethernet switching works on Catalyst 4500 series switches:

- [Layer 2 Ethernet Switching, page 19-2](#)
- [VLAN Trunks, page 19-3](#)
- [Layer 2 Interface Modes, page 19-3](#)

Layer 2 Ethernet Switching

Catalyst 4500 series switches support simultaneous, parallel connections between Layer 2 Ethernet segments. Switched connections between Ethernet segments last only for the duration of the packet. New connections can be made between different segments for successive packets.

**Note**

With Cisco IOS Release 12.1(13)EW, the Catalyst 4500 series switches can handle packets of 1600 bytes, rather than treat them as “oversized” and discard them. This size is larger than the usual IEEE Ethernet Maximum Transmission Unit (MTU) (1518 bytes) and 802.1q MTU (1522 bytes). The ability to handle larger packets is required to support two nested 802.1q headers and Multiprotocol Label Switching (MPLS) on a network.

The Catalyst 4500 series switch solves congestion problems caused by high-bandwidth devices and a large number of users by assigning each device (for example, a server) to its own 10-, 100-, or 1000-Mbps segment. Because each Ethernet interface on the switch represents a separate Ethernet segment, servers in a properly configured switched environment achieve full access to the bandwidth.

Because collisions are a major bottleneck in Ethernet networks, an effective solution is full-duplex communication. Normally, Ethernet operates in half-duplex mode, which means that stations can either receive or transmit. In full-duplex mode, two devices can transmit and receive at the same time. When packets can flow in both directions simultaneously, effective Ethernet bandwidth doubles to 20 Mbps for 10-Mbps interfaces and to 200 Mbps for Fast Ethernet interfaces. Gigabit Ethernet interfaces on the Catalyst 4500 series switch are full-duplex mode only, providing 2-Gbps effective bandwidth.

Switching Frames Between Segments

Each Ethernet interface on a Catalyst 4500 series switch can connect to a single workstation or server, or to a hub through which workstations or servers connect to the network.

On a typical Ethernet hub, all ports connect to a common backplane within the hub, and the bandwidth of the network is shared by all devices attached to the hub. If two devices establish a session that uses a significant level of bandwidth, the network performance of all other stations attached to the hub is degraded.

To reduce degradation, the switch treats each interface as an individual segment. When stations on different interfaces need to communicate, the switch forwards frames from one interface to the other at wire speed to ensure that each session receives full bandwidth.

To switch frames between interfaces efficiently, the switch maintains an address table. When a frame enters the switch, it associates the MAC address of the sending station with the interface on which it was received.

Building the MAC Address Table

The Catalyst 4500 series switch builds the MAC address table by using the source address of the frames received. When the switch receives a frame for a destination address not listed in its MAC address table, it floods the frame to all interfaces of the same VLAN except the interface that received the frame. When the destination device replies, the switch adds its relevant source address and interface ID to the address table. The switch then forwards subsequent frames to a single interface without flooding to all interfaces.

The address table can store at least 32,000 address entries without flooding any entries. The switch uses an aging mechanism, defined by a configurable aging timer, so if an address remains inactive for a specified number of seconds, it is removed from the address table.

VLAN Trunks

A trunk is a point-to-point link between one or more Ethernet switch interfaces and another networking device such as a router or a switch. Trunks carry the traffic of multiple VLANs over a single link and allow you to extend VLANs across an entire network.

Only 802.1Q trunking encapsulations is available on all Catalyst 4500 Ethernet interfaces.



Note Ports 3 to 18 are blocking Gigabit ports on the WS-X4418-GB module. Ports 1 to 12 are blocking Gigabit ports on the WS-X4412-2GB-T module.

You can configure a trunk on a single Ethernet interface or on an EtherChannel bundle. For more information about EtherChannel, see [Chapter 27, “Configuring EtherChannel and Link State Tracking.”](#)

Layer 2 Interface Modes

[Table 19-1](#) lists the Layer 2 interface modes and describes how they function on Ethernet interfaces.

Table 19-1 *Layer 2 Interface Modes*

Mode	Purpose
switchport mode access	Puts the interface into permanent nontrunking mode and negotiates to convert the link into a nontrunking link. The interface becomes a nontrunk interface even if the neighboring interface does not change.
switchport mode dynamic desirable	Makes the interface actively attempt to convert the link to a trunking link. The interface becomes a trunk interface if the neighboring interface is set to trunk , desirable , or auto mode.
switchport mode dynamic auto	Makes the interface convert the link to a trunking link if the neighboring interface is set to trunk or desirable mode. It is the default mode for all Ethernet interfaces.
switchport mode trunk	Puts the interface into permanent trunking mode and negotiates to convert the link into a trunking link. The interface becomes a trunk interface even if the neighboring interface does not change.
switchport nonegotiate	Puts the interface into permanent trunking mode but prevents the interface from generating DTP frames. You must configure the neighboring interface manually as a trunk interface to establish a trunking link.



Note

DTP is a point-to-point protocol. However, some internetworking devices might forward DTP frames improperly. To avoid this problem, ensure that interfaces connected to devices that do not support DTP are configured with the **access** keyword if you do not intend to trunk across those links. To enable trunking to a device that does not support DTP, use the **nonegotiate** keyword to cause the interface to become a trunk without generating DTP frames.

Default Layer 2 Ethernet Interface Configuration

Table 19-2 shows the Layer 2 Ethernet interface default configuration.

Table 19-2 Layer 2 Ethernet Interface Default Configuration

Feature	Default Value
Interface mode	switchport mode dynamic auto
Trunk encapsulation	switchport trunk encapsulation negotiate
Allowed VLAN range	VLANs 1–1005
VLAN range eligible for pruning	VLANs 2–1001
Default VLAN (for access ports)	VLAN 1
Native VLAN (for 802.1Q only trunks)	VLAN 1
STP ¹	Enabled for all VLANs
STP port priority	128
STP port cost	<ul style="list-style-type: none"> • 100 for 10-Mbps Ethernet LAN ports • 19 for 10/100-Mbps Fast Ethernet ports • 19 for 100-Mbps Fast Ethernet ports • 4 for 1000-Mbps Gigabit Ethernet ports • 2 for 10,000-Mbps 10-Gigabit Ethernet LAN ports

1. STP = Spanning Tree Protocol

Layer 2 Interface Configuration Guidelines and Restrictions

When using (or configuring) Layer 2 interfaces, consider these guidelines and restrictions:

- In a network of Cisco switches connected through 802.1Q trunks, the switches maintain one instance of spanning tree for each VLAN allowed on the trunks. Non-Cisco 802.1Q switches maintain only one instance of spanning tree for all VLANs allowed on the trunks.

When you connect a Cisco switch to a non-Cisco device through an 802.1Q trunk, the Cisco switch combines the spanning tree instance of the native VLAN of the trunk with the spanning tree instance of the non-Cisco 802.1Q switch. However, spanning tree information for each VLAN is maintained by Cisco switches separated by a cloud of non-Cisco 802.1Q switches. The non-Cisco 802.1Q cloud separating the Cisco switches is treated as a single trunk link between the switches.

- Make sure the native VLAN for an 802.1Q trunk is the same on both ends of the trunk link. If the VLAN on one end of the trunk is different from the VLAN on the other end, spanning tree loops might result.
- Disabling spanning tree on any VLAN of an 802.1Q trunk can cause spanning tree loops.

Configuring Ethernet Interfaces for Layer 2 Switching

The following sections describe how to configure Layer 2 switching on a Catalyst 4500 series switch:

- [Configuring an Ethernet Interface as a Layer 2 Trunk, page 19-5](#)
- [Configuring an Interface as a Layer 2 Access Port, page 19-7](#)
- [Clearing Layer 2 Configuration, page 19-8](#)

Configuring an Ethernet Interface as a Layer 2 Trunk



Note

The default for Layer 2 interfaces is **switchport mode dynamic auto**. If the neighboring interface supports trunking and is configured to trunk mode or dynamic desirable mode, the link becomes a Layer 2 trunk.

To configure an interface as a Layer 2 trunk, perform this task:

	Command	Purpose
Step 1	Switch(config)# interface { fastethernet gigabitethernet tengigabitethernet } <i>slot/port</i>	Specifies the interface to configure.
Step 2	Switch(config-if)# shutdown	(Optional) Shuts down the interface to prevent traffic flow until configuration is complete.
Step 3	Switch(config-if)# switchport trunk encapsulation { dot1q negotiate }	(Optional) Specifies the encapsulation. Note You must enter this command with the dot1q keyword to support the switchport mode trunk command, which is not supported by the default mode (negotiate).
Step 4	Switch(config-if)# switchport mode { dynamic { auto desirable } trunk }	Configures the interface as a Layer 2 trunk. (Required only if the interface is a Layer 2 access port or to specify the trunking mode.)
Step 5	Switch(config-if)# switchport access vlan <i>vlan_num</i>	(Optional) Specifies the access VLAN, which is used if the interface stops trunking. The access VLAN is not used as the native VLAN. Note The <i>vlan_num</i> parameter is either a single VLAN number from 1 to 1005 or a range of VLANs described by two VLAN numbers, the lesser one first, separated by a dash. Do not enter any spaces between comma-separated <i>vlan</i> parameters or in dash-specified ranges.
Step 6	Switch(config-if)# switchport trunk native vlan <i>vlan_num</i>	For 802.1Q trunks, specifies the native VLAN. Note If you do not set the native VLAN, the default is used (VLAN 1).
Step 7	Switch(config-if)# switchport trunk allowed vlan { add except all remove } <i>vlan_num[,vlan_num[,vlan_num[,...]]]</i>	(Optional) Configures the list of VLANs allowed on the trunk. All VLANs are allowed by default. You cannot remove any of the default VLANs from a trunk.

	Command	Purpose
Step 8	Switch(config-if)# switchport trunk pruning vlan {add except none remove} vlan_num[,vlan_num[,vlan_num[,...]]	(Optional) Configures the list of VLANs allowed to be pruned from the trunk (see the “VLAN Trunking Protocol” section on page 17-7). The default list of VLANs allowed to be pruned contains all VLANs, except for VLAN 1.
Step 9	Switch(config-if)# no shutdown	Activates the interface. (Required only if you shut down the interface.)
Step 10	Switch(config-if)# end	Exits interface configuration mode.
Step 11	Switch# show running-config interface {fastethernet gigabitethernet tengigabitethernet} slot/port	Displays the running configuration of the interface.
Step 12	Switch# show interfaces [fastethernet gigabitethernet tengigabitethernet] slot/port switchport	Displays the switch port configuration of the interface.
Step 13	Switch# show interfaces [{fastethernet gigabitethernet tengigabitethernet} slot/port] trunk	Displays the trunk configuration of the interface.

This example shows how to configure the Fast Ethernet interface 5/8 as an 802.1Q trunk. This example assumes that the neighbor interface is configured to support 802.1Q trunking and that the native VLAN defaults to VLAN 1:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface fastethernet 5/8
Switch(config-if)# shutdown
Switch(config-if)# switchport mode dynamic desirable
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# no shutdown
Switch(config-if)# end
Switch# exit
```

This example shows how to verify the running configuration:

```
Switch# show running-config interface fastethernet 5/8
Building configuration...
Current configuration:
!
interface FastEthernet5/8
 switchport mode dynamic desirable
 switchport trunk encapsulation dot1q
end
```

This example shows how to verify the switch port configuration:

```
Switch# show interfaces fastethernet 5/8 switchport
Name: Fa5/8
Switchport: Enabled
Administrative Mode: dynamic desirable
Operational Mode: trunk
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: Enabled
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
```

This example shows how to verify the trunk configuration:

```
Switch# show interfaces fastethernet 5/8 trunk

Port      Mode           Encapsulation  Status        Native vlan
Fa5/8     desirable      n-802.1q       trunking      1

Port      Vlans allowed on trunk
Fa5/8     1-1005

Port      Vlans allowed and active in management domain
Fa5/8     1-6,10,20,50,100,152,200,300,303-305,349-351,400,500,521,524,570,801-802,850,917,999,1002-1005

Port      Vlans in spanning tree forwarding state and not pruned
Fa5/8     1-6,10,20,50,100,152,200,300,303-305,349-351,400,500,521,524,570,801-802,850,917,999,1002-1005

Switch#
```

Configuring an Interface as a Layer 2 Access Port



Note

If you assign an interface to a VLAN that does not exist, the interface is not operational until you create the VLAN in the VLAN database (see the [“Configuring VLANs in Global Configuration Mode”](#) section on page 17-5).

To configure an interface as a Layer 2 access port, perform this task:

	Command	Purpose
Step 1	Switch(config)# interface { fastethernet gigabitethernet tengigabitethernet } <i>slot/port</i>	Specifies the interface to configure.
Step 2	Switch(config-if)# shutdown	(Optional) Shuts down the interface to prevent traffic flow until configuration is complete.
Step 3	Switch(config-if)# switchport	Configures the interface for Layer 2 switching: <ul style="list-style-type: none"> You must enter the switchport command once without any keywords to configure the interface as a Layer 2 port before you can enter additional switchport commands with keywords. Required only if you previously entered the no switchport command for the interface.
Step 4	Switch(config-if)# switchport mode access	Configures the interface as a Layer 2 access port.
Step 5	Switch(config-if)# switchport access vlan <i>vlan_num</i>	Places the interface in a VLAN.
Step 6	Switch(config-if)# no shutdown	Activates the interface. (Required only if you had shut down the interface.)
Step 7	Switch(config-if)# end	Exits interface configuration mode.

	Command	Purpose
Step 8	Switch# show running-config interface { fastethernet gigabitethernet } <i>slot/port</i>	Displays the running configuration of the interface.
Step 9	Switch# show interfaces [{ fastethernet gigabitethernet tengigabitethernet } <i>slot/port</i>] switchport	Displays the switch port configuration of the interface.

This example shows how to configure the Fast Ethernet interface 5/6 as an access port in VLAN 200:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface fastethernet 5/6
Switch(config-if)# shutdown
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 200
Switch(config-if)# no shutdown
Switch(config-if)# end
Switch# exit
```

This example shows how to verify the running configuration:

```
Switch# show running-config interface fastethernet 5/6
Building configuration...
!
Current configuration :33 bytes
interface FastEthernet 5/6
  switchport access vlan 200
  switchport mode access
end
```

This example shows how to verify the switch port configuration:

```
Switch# show interface fastethernet 5/6 switchport
Name:Fa5/6
Switchport:Enabled
Administrative Mode:dynamic auto
Operational Mode:static access
Administrative Trunking Encapsulation:negotiate
Operational Trunking Encapsulation:native
Negotiation of Trunking:On
Access Mode VLAN:1 (default)
Trunking Native Mode VLAN:1 (default)
Administrative private-vlan host-association:none
Administrative private-vlan mapping:none
Operational private-vlan:none
Trunking VLANs Enabled:ALL
Pruning VLANs Enabled:2-1001
Switch#
```

Clearing Layer 2 Configuration

To clear the Layer 2 configuration on an interface, perform this task:

	Command	Purpose
Step 1	Switch(config)# default interface { fastethernet gigabitethernet tengigabitethernet } <i>slot/port</i>	Specifies the interface to clear.
Step 2	Switch(config-if)# end	Exits interface configuration mode.

	Command	Purpose
Step 3	Switch# show running-config interface { fastethernet gigabitethernet tengigabitethernet } <i>slot/port</i>	Displays the running configuration of the interface.
Step 4	Switch# show interfaces [{ fastethernet gigabitethernet tengigabitethernet } <i>slot/port</i>] switchport	Displays the switch port configuration of the interface.

This example shows how to clear the Layer 2 configuration on the Fast Ethernet interface 5/6:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# default interface fastethernet 5/6
Switch(config)# end
Switch# exit
```

This example shows how to verify that the Layer 2 configuration was cleared:

```
Switch# show running-config interface fastethernet 5/6
Building configuration...
Current configuration:
!
interface FastEthernet5/6
end
```

This example shows how to verify the switch port configuration:

```
Switch# show interfaces fastethernet 5/6 switchport
Name: Fa5/6
Switchport: Enabled
Switch#
```




Configuring EVC-Lite

This document describes how to configure EVC-Lite, which is a lite version of the Ethernet Virtual Connections (EVC) feature on Supervisor Engine 7L-E. The associated command pages are also provided.

The document details:

- [About EVC-Lite, page 20-1](#)
- [How to Configure EVC-Lite, page 20-2](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About EVC-Lite

Ethernet virtual circuits (EVCs) define a Layer 2 bridging architecture that supports Ethernet services. An EVC is defined by the Metro-Ethernet Forum (MEF) as an association between two or more user network interfaces that identifies a point-to-point or multipoint-to-multipoint path within the service provider network. An EVC is a conceptual service pipe within the service provider network. A bridge domain is a local broadcast domain that exists separately from VLANs.

A Catalyst 4500 series switch comprises of two bridge domains (BDs), BD 0 and BD 1. By default, all ports belong to BD 0 and you can move them to BD 1 manually.

EVC-Lite supports 8K VLANs using the existing support of 8K Internal VLANs, segregated under the two BDs, where each BD has its own representation of 4K VLANs.

A system can have two types of VLAN's: EVC-Lite and non EVC-Lite (the default). Ordinary VLANs are VLANs without awareness of any BDs (the situation that exists when the feature is not enabled). EVC-Lite VLANs are treated differently in different bridge domains (e.g the same VLAN 2 is treated differently in BD 0 and BD 1).

A port can support both EVC-Lite and non EVC-Lite VLANs. EVC-Lite VLANs that are part of BD 1 are mapped to an internal VLAN ID as VLAN ID + 4096. Remaining VLANs are not mapped internally.

How to Configure EVC-Lite

Step 1 Create a VLAN:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# vlan vlan-ID
Switch(config-vlan)# exit
```

Step 2 Configure the VLAN as an EVC-Lite VLAN:

```
Switch(config)# vlan configuration vlan_ID
Switch(config-vlan-config)# evc-lite
Switch(config-vlan-config)# exit
```

Step 3 Associate the EVC-Lite VLAN to an interface:

```
Switch(config)# interface gigabitEthernet slot/port
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan vlan_ID
```

Step 4 Enable the interface with bridge-domain {0|1}:

```
Switch(config-if)#?
Interface configuration commands:
-----
      evc-lite          set the interface in evc-lite mode
--More--
Switch(config-if)# evc-lite bridge-domain bridge-domain
Switch(config-if)# exit
```

This sequence creates an EVC-Lite VLAN and associates it to an interface. The same VLAN can be associated with multiple interfaces, although each interface can have only one bridge-domain. On a Catalyst 4500 series switch, we can have two BDs (0 and 1). Because each BD supports 4K VLAN's, we can support 8K VLANs. An EVC-Lite VLAN can be associated with the BD 0 and 1 interfaces. However, traffic flowing on this VLAN under BD 1 will never flow under BD 0 and vice-versa.

This example shows how to configure VLAN 10 as EVC-Lite, enable interface GigabitEthernet 7/1 in BD 1, and display configured EVC-Lite VLANs and ports in BD 1:

```
Switch(config)# vlan configuration 10
Switch(config-vlan-config)# evc-lite
Switch(config-vlan-config)# exit
Switch# show running-config | begin vlan configuration
vlan configuration 10
    evc-lite

Switch(config)# interface gigabitEthernet 7/1
Switch(config-if)# switchport mode trunk
Switch(config-if)# evc-lite bridge-domain 1
Switch(config-if)# exit
Switch# show run interface gigabitEthernet 7/1
Building configuration...

Current configuration : 119 bytes
!
interface GigabitEthernet7/1
    switchport mode trunk
    evc-lite bridge-domain 1
end
```

```
Switch# show evc-lite
evc-lite vlans: 10
Ports in bridge-domain 1: Gi7/1
```

**Note**

Because a port channel can only accommodate member links belonging to the BD of the port-channel, the **show evc-lite** command displays only the port-channel instead of all the member links.



Configuring SmartPort Macros

This chapter describes how to configure and apply SmartPort and Static SmartPort macros on a Catalyst 4500 series switch. This chapter consists of these sections:

- [About SmartPort Macros and Static SmartPort, page 21-1](#)
- [Configuring SmartPort Macros, page 21-2](#)
- [Displaying SmartPort Macros, page 21-12](#)
- [Configuring Static SmartPort Macros, page 21-13](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About SmartPort Macros and Static SmartPort

SmartPort macros provide a convenient way to save and share common configurations. Use SmartPort macros to enable features and settings based on the location of a switch in the network and for mass configuration deployments across the network.

Each SmartPort macro is a set of CLI commands that you define. SmartPort macro sets do not contain new CLI commands; each SmartPort macro is a group of existing CLI commands.

When you apply a SmartPort macro on an interface, the CLI commands contained within the macro are configured on the interface. When the macro is applied to an interface, the existing interface configurations are not lost. The new commands are added to interface and are saved in the running configuration file.

In addition to SmartPort macros, static SmartPort macros provide port configuration that you manually apply based on the device connected to the port. When you apply a static SmartPort macro the CLI commands within the macro are added to the existing port configuration. When there is a link-down event on the port, the switch does not remove the static macro.

Cisco-default SmartPort macros are embedded in the switch software (see [Table 21-1](#)). You can display these macros and the commands they contain by using the **show parser macro** user EXEC command.

Table 21-1 *Cisco-Default SmartPort Macros*

Macro Name ¹	Description
cisco-global	Use this global configuration macro to enable rapid PVST+, loop guard, and dynamic port error recovery for link state failures.
cisco-desktop	Use this interface configuration macro for increased network security and reliability when connecting a desktop device, such as a PC, to a switch port.
cisco-phone	Use this interface configuration macro when connecting a desktop device such as a PC with a Cisco IP Phone to a switch port. This macro is an extension of the cisco-desktop macro and provides the same security and resiliency features, but with the addition of dedicated voice VLANs to ensure proper treatment of delay-sensitive voice traffic.
cisco-switch	Use this interface configuration macro when connecting an access switch and a distribution switch or between access switches connected using GigaStack modules or GBICs.
cisco-router	Use this interface configuration macro when connecting the switch and a WAN router.

1. Cisco-default SmartPort macros vary depending on the software version running on your switch.

Cisco also provides a collection of pretested, Cisco-recommended baseline configuration templates for Catalyst switches. The online reference guide templates provide the CLI commands that you use to create SmartPort macros based on the use of the port. Use the configuration templates to create SmartPort macros to build and deploy Cisco-recommended network designs and configurations.

Configuring SmartPort Macros

You can create a new SmartPort macro or use an existing macro as a template to create a new macro that is specific to your application. After you create the macro, you can apply it to an interface or a range of interfaces.

This section includes information about these topics:

- [Passing Parameters Through the Macro, page 21-2](#)
- [Default SmartPort Macro Configuration, page 21-3](#)
- [SmartPort Macro Configuration Guidelines, page 21-6](#)
- [Creating SmartPort Macros, page 21-7](#)
- [Applying SmartPort Macros, page 21-8](#)

Passing Parameters Through the Macro

Some commands might not be sufficiently generic for all the interfaces; for example, VLAN ID for Layer 2 interfaces and the IP address for Layer 3 interface. Retaining such commands in macro definitions requires that you change the value of such parameters (such as VLAN ID or IP address) before applying the macro to different interfaces. Alternatively, it requires that you create different macros for each possible value of its parameters.

The macro infrastructure can be enhanced to support accepting parameters while applying a macro. The parameters are passed as *keyword-value* pairs.

The CLI limits the number of keyword-value pairs to a maximum of three, where the first parameter must be the keyword, the second is its corresponding value, and the third parameter is the keyword for the second keyword-value pair. Here is an example of how to pass parameters to a command macro:

```
Switch(config)# macro name parameter-test  
Enter macro commands one per line. End with the character '@'.  
switchport mode access  
switchport access vlan $VLANID  
switchport port-security  
switchport port-security maximum $MAXHOST
```

If the above macro is applied to some interface without parameters, the invalid commands fail. Instead, you should apply the macro with appropriate keyword-value pair parameters, as follows:

```
Switch(config-if)# macro apply parameter-test $VLANID 1 $MAXHOST 5
```

The above command applies the macro after replacing \$VLANID with 1 and \$MAXHOST with 5. Be aware that you can specify any string in the macro as a keyword.

Macro Parameter Help

It is often difficult to remember the macro keywords while applying a macro to an interface or switch. Macros can contain the definitions for mandatory keywords. If you apply a macro without those keyword values, the commands are considered invalid and they fail.

You can enhance the macro infrastructure to provide help on keywords defined in macros. While creating a macro, you can specify a help string (as a comment) to list the mandatory keywords for that macro.

The following example illustrates how to specify the help string for the keywords:

```
Switch(config)# macro name test  
switchport access vlan $VLANID  
switchport port-security maximum $MAX  
#macro keywords $VLANID $MAX
```

Help string can be anywhere in the macro. The following example illustrates an alternate way to specify the help string:

```
Switch(config)# macro name test  
switchport access vlan $VLANID  
#macro keywords $VLANID  
switchport port-security maximum $MAX  
#macro keywords $MAX
```

Default SmartPort Macro Configuration

This section illustrates the default configurations for the four supported macros. These macros can only be viewed and applied; they cannot be modified by the user.

- [cisco-global, page 21-4](#)
- [cisco-desktop, page 21-4](#)
- [cisco-phone, page 21-4](#)
- [cisco-router, page 21-5](#)
- [cisco-switch, page 21-5](#)

cisco-global

This is the example for the cisco-global macro:

```
# Enable dynamic port error recovery for link state failures.
errdisable recovery cause link-flap
errdisable recovery interval 60

# VTP requires Transparent mode for future 802.1x Guest VLAN
# and current Best Practice
vtp domain [smartports]
vtp mode transparent

# Enable aggressive mode UDLD on all fiber uplinks
udld aggressive

# Enable Rapid PVST+ and Loopguard
spanning-tree mode rapid-pvst
spanning-tree loopguard default
spanning-tree extend system-id
```

cisco-desktop

This is the example for the cisco-desktop macro:

```
# Basic interface - Enable data VLAN only
# Recommended value for access vlan (AVID) should not be 1
switchport access vlan $AVID
switchport mode access
# Enable port security limiting port to a single
# MAC address -- that of desktop
switchport port-security
# Ensure port-security age is greater than one minute
# and use inactivity timer
# "Port-security maximum 1" is the default and will not
# Show up in the config
switchport port-security violation restrict
switchport port-security aging time 2
switchport port-security aging type inactivity
# Configure port as an edge network port
spanning-tree portfast
spanning-tree bpduguard enable
```

cisco-phone

This is the example for the cisco-phone macro:

```
# VoIP enabled interface - Enable data VLAN
# and voice VLAN (VVID)
# Recommended value for access vlan (AVID) should not be 1\
switchport access vlan $AVID
switchport mode access
# Update the Voice VLAN (VVID) value which should be
# different from data VLAN
# Recommended value for voice vlan (VVID) should not be 1
switchport voice vlan $VVID
# Enable port security limiting port to a 2 MAC
# addressess -- One for desktop and two for phone
switchport port-security
switchport port-security maximum 2
# Ensure port-security age is greater than one minute
# and use inactivity timer
```



```
switchport port-security violation restrict
switchport port-security aging time 2
switchport port-security aging type inactivity
# Enable auto-qos to extend trust to attached Cisco phone
auto qos voip cisco-phone
# Configure port as an edge network port
spanning-tree portfast
spanning-tree bpduguard enable@
```

cisco-router

This is the example for the cisco-router macro:

```
# Access Uplink to Distribution
switchport trunk encapsulation dot1q
# Define unique Native VLAN on trunk ports
# Recommended value for native vlan (NVID) should not be 1
switchport trunk native vlan $NVID
# Update the allowed VLAN range (VRANGE) such that it
# includes data, voice and native VLANs
# switchport trunk allowed vlan $VRANGE
# Hardcode trunk and disable negotiation to
# speed up convergence
# Hardcode speed and duplex to router
switchport mode trunk
switchport nonegotiate
speed 100
duplex full
# Configure qos to trust this interface
auto qos voip trust
qos trust dscp
# Ensure fast access to the network when enabling the interface.
# Ensure that switch devices cannot become active on the interface.
spanning-tree portfast
spanning-tree bpduguard enable
```

cisco-switch

This is the example for the cisco-switch macro:

```
# Access Uplink to Distribution
switchport trunk encapsulation dot1q
# Define unique Native VLAN on trunk ports
# Recommended value for native vlan (NVID) should not be 1
switchport trunk native vlan $NVID
# Update the allowed VLAN range (VRANGE) such that it
# includes data, voice and native VLANs
# switchport trunk allowed vlan $VRANGE
# Hardcode trunk and disable negotiation to
# speed up convergence
switchport mode trunk
switchport nonegotiate
# Configure qos to trust this interface
auto qos voip trust
# 802.1w defines the link as pt-pt for rapid convergence
spanning-tree link-type point-to-point
```

SmartPort Macro Configuration Guidelines

Follow these guidelines when configuring macros on your switch:

- If a command fails when you apply a macro, either due to a syntax error or to a configuration error, the macro continues to apply the remaining commands to the interface.
- **cisco-global** needs to be applied at the global configuration mode. We recommend that you apply this macro before any other interface level macro.
- Specific keywords are required when you apply the system-defined macros (**cisco-desktop**, **cisco-phone**, **cisco-switch**, and **cisco-router**) on an interface.
- When using the **cisco-phone** macro to apply port security, the port security maximum is 2 (**switchport port-security maximum 2**).
- At most, three keyword-value pairs are allowed per system-defined macro.
- When creating a macro, do not use the **exit** or **end** commands or change the command mode by using **interface interface-id**. This could cause commands that follow **exit**, **end**, or **interface interface-id** to execute in a different command mode.
- When creating a macro, ensure that all CLI commands are in the same configuration mode.
- When creating a macro that requires the assignment of unique values, use the **parameter value** keywords to designate values specific to the interface. Keyword matching is case sensitive. All matching occurrences of the keyword are replaced with the corresponding value. Any full match of a keyword, even if it is part of a larger string, is considered a match and is replaced by the corresponding value.
- Macro names are case sensitive. For example, the commands **macro name Sample-Macro** and **macro name sample-macro** result in two separate macros.
- Some macros might contain keywords that require a parameter value. Use the **macro global apply macro-name ?** global configuration command or the **macro apply macro-name ?** interface configuration command to display a list of any required values in the macro. If you apply a macro without entering the keyword values, the commands are invalid and are not applied.
- When a macro is applied globally to a switch or to a switch interface, all existing configuration on the interface is retained. It is helpful when applying an incremental configuration.
- If you modify a macro definition by adding or deleting commands, the changes are not reflected on the interface where the original macro was applied. You need to reapply the updated macro on the interface to apply the new or changed commands.
- Use the **macro global trace macro-name** global configuration command or the **macro trace macro-name** interface configuration command to apply and debug a macro to find any syntax or configuration errors. If a command fails because of a syntax error or a configuration error, the macro continues to apply the remaining commands.
- Some CLI commands are specific to certain interface types. If a macro is applied to an interface that does not accept the configuration, the macro fails the syntax check or the configuration check, and the switch returns an error message.
- Applying a macro to an interface range is the same as applying a macro to a single interface. When you use an interface range, the macro is applied sequentially to each interface within the range. If a macro command fails on one interface, it is still applied to the remaining interfaces.

- When you apply a macro to a switch or a switch interface, the macro name is automatically added to the macro description of the switch or interface. You can display the applied commands and macro names by using the **show parser macro description** user EXEC command.
- The user-configurable macro has a buffer that can take commands and comments up to 3000 characters. Each new line takes two characters, and empty lines are counted as is.

Cisco-default SmartPort macros are embedded in the switch software (see [Table 21-1](#)). You can display these macros and the commands they contain by using the **show parser macro** user EXEC command.

Follow these guidelines when you apply a Cisco-default SmartPort macro on an interface:

- Display all macros on the switch by using the **show parser macro** user EXEC command. Display the contents of a specific macro by using the **show parser macro macro-name** user EXEC command.
- Keywords that begin with \$ mean that a unique parameter value is required. Append the Cisco-default macro with the required values by using the **parameter value** keywords.

The Cisco-default macros use the \$ character to help identify required keywords. There is no restriction on using the \$ character to define keywords when you create a macro.

Creating SmartPort Macros

To create a SmartPort macro, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# macro name <i>macro-name</i>	<p>Creates a macro definition, and enter a macro name. A macro definition can contain up to 3000 characters.</p> <p>Enter the macro commands with one command per line. Use the @ character to end the macro. Use the # character at the beginning of a line to enter comment text within the macro.</p> <p>Macro names are case sensitive. For example, the commands macro name Sample-Macro and macro name sample-macro result in two separate macros.</p> <p>We recommend that you do not use the exit or end commands or change the command mode by using interface interface-id in a macro. This could cause any commands following exit, end, or interface interface-id to execute in a different command mode. For best results, all commands in a macro should be in the same configuration mode.</p> <p>Note The no form of the macro name global configuration command only deletes the macro definition. It does not affect the configuration of those interfaces on which the macro is already applied.</p>
Step 3	Switch(config)# end	Returns to privileged EXEC mode.
Step 4	Switch# show parser macro name <i>macro-name</i>	Verifies that the macro was created.

Applying SmartPort Macros

To apply a SmartPort macro, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# macro global { apply trace } <i>macro-name</i> [parameter { <i>value</i> }] [parameter { <i>value</i> }] [parameter { <i>value</i> }]	<p>Applies each individual command defined in the macro to the switch by entering macro global apply macro-name. Specify macro global trace macro-name to apply and debug a macro to find any syntax or configuration errors.</p> <p>(Optional) Specify unique parameter values that are specific to the switch. You can enter up to three keyword-value pairs. Parameter keyword matching is case sensitive. All matching occurrences of the keyword are replaced with the corresponding value.</p> <p>Some macros might contain keywords that require a parameter value. Use the macro global apply macro-name ? command to display a list of any required values in the macro. If you apply a macro without entering the keyword values, the commands are invalid and are not applied.</p>
Step 3	Switch(config)# macro global description <i>text</i>	(Optional) Enters a description about the macro that is applied to the switch.
Step 4	Switch(config-if)# interface <i>interface-id</i>	(Optional) Enters interface configuration mode, and specify the interface on which to apply the macro.
Step 5	Switch(config-if)# default interface <i>interface-id</i>	(Optional) Clears all configuration from the specified interface.
Step 6	Switch(config-if)# macro { apply trace } <i>macro-name</i> [parameter { <i>value</i> }] [parameter { <i>value</i> }] [parameter { <i>value</i> }]	<p>Applies each individual command defined in the macro to the interface by entering macro apply macro-name. Specify macro trace macro-name to apply and debug a macro to find any syntax or configuration errors.</p> <p>(Optional) Specify unique parameter values that are specific to the interface. You can enter up to three keyword-value pairs. Parameter keyword matching is case sensitive. All matching occurrences of the keyword are replaced with the corresponding value.</p> <p>Some macros might contain keywords that require a parameter value. Use the macro apply macro-name ? command to display a list of any required values in the macro. If you apply a macro without entering the keyword values, the commands are invalid and are not applied.</p> <p>For example, here is how you apply this command:</p> <pre>Switch(config-if)# macro apply cisco-phone ? WORD Keyword to replace with a value e.g. \$AVID, \$VVID <cr></pre>
Step 7	Switch(config-if)# macro description <i>text</i>	(Optional) Enters a description about the macro that is applied to the interface.
Step 8	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 9	Switch# show parser macro description [interface <i>interface-id</i>]	Verifies that the macro is applied to the interface.
Step 10	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

You can delete a global macro-applied configuration on a switch only by entering the **no** version of each command that is in the macro. You can delete a macro-applied configuration on an interface by entering the **default interface** *interface-id* interface configuration command.

The **no** form of the **macro name** global configuration command deletes only the macro definition. It does not affect the configuration of those interfaces on which the macro is already applied. You can delete a macro-applied configuration on an interface by entering the **default interface** *interface-id* interface configuration command. Alternatively, you can create an *anti-macro* for an existing macro that contains the **no** form of all the corresponding commands in the original macro and apply the anti-macro to the interface.

The following sections describe how to apply and display the attachments on each of the supported macros:

- [cisco-global, page 21-9](#)
- [cisco-desktop, page 21-9](#)
- [cisco-phone, page 21-10](#)
- [cisco-switch, page 21-11](#)
- [cisco-router, page 21-11](#)

cisco-global

This example shows how to use the system-defined macro **cisco-global**:

```
Switch(config)# macro global apply cisco-global
Changing VTP domain name from gsg-switch to [smartports]
Setting device to VTP TRANSPARENT mode.
Switch(config)# end
Switch# show parser macro name cisco-global
Macro name : cisco-global
Macro type : default global
# Enable dynamic port error recovery for link state failures.
errdisable recovery cause link-flap
errdisable recovery interval 60

# VTP requires Transparent mode for future 802.1x Guest VLAN
# and current Best Practice vtp domain [smartports] vtp mode transparent

# Enable aggressive mode UDLD on all fiber uplinks udld aggressive

# Enable Rapid PVST+ and Loopguard
spanning-tree mode rapid-pvst
spanning-tree loopguard default
spanning-tree extend system-id
```

cisco-desktop

This example shows how to use the system-defined macro **cisco-desktop** to assign a value of 35 to the access VLAN of the Fast Ethernet interface 2/9.



Note

This macro requires the **\$AVID** keyword, which is the access VLAN of the port.

```
Switch(config)# interface fastethernet2/9
Switch(config-if)# macro apply cisco-desktop $AVID 35
Switch(config-if)# end
```

```

Switch# show parser macro name cisco-desktop
Macro name : cisco-desktop
Macro type : customizable

# Basic interface - Enable data VLAN only
# Recommended value for access vlan (AVID) should not be 1
switchport access vlan $AVID [access_vlan_id]
switchport mode access
# Enable port security limiting port to a single
# MAC address -- that of desktop
switchport port-security
# Ensure port-security age is greater than one minute
# and use inactivity timer
# "Port-security maximum 1" is the default and will not
# Show up in the config
switchport port-security violation restrict
switchport port-security aging time 2
switchport port-security aging type inactivity
# Configure port as an edge network port
spanning-tree portfast
spanning-tree bpduguard enable
Switch# show parser macro description
Interface      Macro Description
-----
Fa2/9          cisco-desktop
-----

```

cisco-phone

This example shows how to use the system-defined macro **cisco-phone** to assign a value of 35 to the access VLAN and 56 to the voice VLAN on the Fast Ethernet interface 2/9.



Note

This macro requires the **\$AVID** and **\$VVID** keywords, which are the access and voice VLANs of the port.

```

Switch(config)# interface fastethernet2/9
Switch(config-if)# macro apply cisco-phone
Switch(config-if)# macro description cisco-phone $AVID 35 $VVID 56
Switch(config-if)# end
Switch# show parser macro name cisco-phone
Macro name : cisco-phone
Macro type : customizable

# VoIP enabled interface - Enable data VLAN
# and voice VLAN (VVID)
# Recommended value for access vlan (AVID) should not be 1\
switchport access vlan $AVID [access_vlan_id]
switchport mode access
# Update the Voice VLAN (VVID) value which should be
# different from data VLAN
# Recommended value for voice vlan (VVID) should not be 1
switchport voice vlan $VVID [voice_vlan_id]
# Enable port security limiting port to a 2 MAC
# addressess -- One for desktop and one for phone
switchport port-security
switchport port-security maximum 2
# Ensure port-security age is greater than one minute
# and use inactivity timer
switchport port-security violation restrict
switchport port-security aging time 2

```

```

switchport port-security aging type inactivity
# Enable auto-qos to extend trust to attached Cisco phone
auto qos voip cisco-phone
# Configure port as an edge network port
spanning-tree portfast
spanning-tree bpduguard enable@

```

```
Switch# show parser macro description
```

```
Interface      Macro Description
```

```
-----
Fa2/9          cisco-phone
-----
```

cisco-switch

This example shows how to use the system-defined macro **cisco-switch** to assign a value of 38 to the native VLAN on the Fast Ethernet interface 2/9.



Note

This macro requires the **\$NVID** keyword, which is the native VLANs of the port.

```

Switch(config)# interface fastethernet2/9
Switch(config-if)# macro apply cisco-switch
Switch(config-if)# macro description cisco-switch $NVID 38
Switch(config-if)# end
Switch# show parser macro name cisco-switch
Macro name : cisco-switch
Macro type : customizable

```

```

# Access Uplink to Distribution
switchport trunk encapsulation dot1q
# Define unique Native VLAN on trunk ports
# Recommended value for native vlan (NVID) should not be 1
switchport trunk native vlan $NVID [native_vlan_id]
# Update the allowed VLAN range (VRANGE) such that it
# includes data, voice and native VLANs
# switchport trunk allowed vlan $VRANGE [vlan_range]
# Hardcode trunk and disable negotiation to
# speed up convergence
switchport mode trunk
switchport nonegotiate
# Configure qos to trust this interface
auto qos voip trust
# 802.1w defines the link as pt-pt for rapid convergence
spanning-tree link-type point-to-point

```

```
Switch# show parser macro description
```

```
Interface      Macro Description
```

```
-----
Fa2/9          cisco-switch
-----
```

cisco-router

This example shows how to use the system-defined macro **cisco-router** to assign a value of 451 to the native VLAN on the Fast Ethernet interface 2/9.



Note

This macro requires the **\$NVID** keyword, which is the native VLANs of the port.

```
Switch(config)# interface fastethernet2/9
Switch(config-if)# macro apply cisco-router
Switch(config-if)# macro description cisco-router $NVID 45I
Switch(config-if)# end
Switch# show parser macro name cisco-router
Macro name : cisco-router
Macro type : customizable

# Access Uplink to Distribution
switchport trunk encapsulation dot1q
# Define unique Native VLAN on trunk ports
# Recommended value for native vlan (NVID) should not be 1
switchport trunk native vlan $NVID [native_vlan_id]
# Update the allowed VLAN range (VRANGE) such that it
# includes data, voice and native VLANs
# switchport trunk allowed vlan $VRANGE [vlan_range]
# Hardcode trunk and disable negotiation to
# speed up convergence
# Hardcode speed and duplex to router
switchport mode trunk
switchport nonegotiate
speed 100
duplex full
# Configure qos to trust this interface
auto qos voip trust
qos trust dscp
# Ensure fast access to the network when enabling the interface.
# Ensure that switch devices cannot become active on the interface.
spanning-tree portfast
spanning-tree bpduguard enable

Switch# show parser macro description
Interface      Macro Description
-----
Fa2/9          cisco-router
-----
```

Displaying SmartPort Macros

To display the SmartPort macros, use one or more of the privileged EXEC commands in [Table 21-2](#).

Table 21-2 *Commands for Displaying SmartPort Macros*

Command	Purpose
<code>show parser macro</code>	Displays all configured macros.
<code>show parser macro name <i>macro-name</i></code>	Displays a specific macro.
<code>show parser macro brief</code>	Displays the configured macro names.
<code>show parser macro description [<i>interface interface-id</i>]</code>	Displays the macro description for all interfaces or for a specified interface.

Configuring Static SmartPort Macros

This section describes how to configure and enable static SmartPort macros:

- [Default SmartPort Macro Configuration, page 21-3](#)
- [SmartPort Macro Configuration Guidelines, page 21-6](#)
- [Applying Static SmartPort Macros, page 21-14](#)

Default Static SmartPort Configuration

No static SmartPort macros are enabled on the switch.

Table 21-3 **Default Static SmartPort Macros**

Macro Name ¹	Description
cisco-global	Use this global configuration macro to enable rapid PVST+, loop guard, and dynamic port error recovery for link state failures.
cisco-desktop	Use this interface configuration macro for increased network security and reliability when connecting a desktop device, such as a PC, to a switch port.
cisco-phone	Use this interface configuration macro when connecting a desktop device such as a PC with a Cisco IP Phone to a switch port. This macro is an extension of the cisco-desktop macro and provides the same security and resiliency features, but with the addition of dedicated voice VLANs to ensure proper treatment of delay-sensitive voice traffic.
cisco-switch	Use this interface configuration macro when connecting an access switch and a distribution switch or between access switches connected by using small form-factor pluggable (SFP) modules.
cisco-router	Use this interface configuration macro when connecting the switch and a WAN router.

1. Cisco-default SmartPort macros vary, depending on the software version running on your switch.

Static SmartPort Configuration Guidelines

- When a macro is applied globally to a switch or to a switch interface, all existing configuration on the interface is retained. It is helpful when applying an incremental configuration.
- If a command fails because of a syntax or a configuration error, the macro continues to apply the remaining commands. Use the **macro global trace macro-name** global configuration command or the **macro trace macro-name** interface configuration command to apply and debug a macro to find any syntax or configuration errors.
- Some CLI commands are specific to certain interface types. If you apply a macro to an interface that does not accept the configuration, the macro fails the syntax or the configuration check, and the switch returns an error message.
- Applying a macro to an interface range is the same as applying a macro to a single interface. When you use an interface range, the macro is applied sequentially to each interface within the range. If a macro command fails on one interface, it is still applied to the remaining interfaces.
- When you apply a macro to a switch or a switch interface, the macro name is automatically added to the switch or interface. You can display the applied commands and macro names by using the **show running-config** user EXEC command.

Applying Static SmartPort Macros

To apply a static SmartPort macro, perform these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	show parser macro	Displays the Cisco-default staticSmartPort macros embedded in the switch software.
Step 2	show parser macro name <i>macro-name</i>	Displays the specific macro that you want to apply.
Step 3	configure terminal	Enters global configuration mode.
Step 4	macro global { apply trace } <i>macro-name</i> [parameter { <i>value</i> }] [parameter { <i>value</i> }] [parameter { <i>value</i> }]	<p>Applies each individual command defined in the macro to the switch by entering macro global apply macro-name. Specify macro global trace macro-name to apply and to debug a macro to find any syntax or configuration errors.</p> <p>Append the macro with the required values by using the parameter value keywords. Keywords that begin with \$ require a unique parameter value.</p> <p>Use the macro global apply macro-name ? command to display a list of any required values for the macro. If you apply a macro without entering the keyword values, the commands are invalid and are not applied.</p> <p>(Optional) Specify unique parameter values that are specific to the switch. You can enter up to three keyword-value pairs. Parameter keyword matching is case sensitive. The corresponding value replaces all matching occurrences of the keyword.</p>
Step 5	interface <i>interface-id</i>	(Optional) Enters interface configuration mode, and specify the interface on which to apply the macro.
Step 6	default interface <i>interface-id</i>	(Optional) Clears all configuration from the specified interface.
Step 7	macro { apply trace } <i>macro-name</i> [parameter { <i>value</i> }] [parameter { <i>value</i> }] [parameter { <i>value</i> }]	<p>Applies each individual command defined in the macro to the port by entering macro global apply macro-name. Specify macro global trace macro-name to apply and to debug a macro to find any syntax or configuration errors.</p> <p>Append the macro with the required values by using the parameter value keywords. Keywords that begin with \$ require a unique parameter value.</p> <p>Use the macro global apply macro-name ? command to display a list of any required values for the macro. If you apply a macro without entering the keyword values, the commands are invalid and are not applied.</p> <p>(Optional) Specify unique parameter values that are specific to the switch. You can enter up to three keyword-value pairs. Parameter keyword matching is case sensitive. The corresponding value replaces all matching occurrences of the keyword.</p>
Step 8	end	Returns to privileged EXEC mode.
Step 9	show running-config interface <i>interface-id</i>	Verifies that the macro is applied to an interface.
Step 10	copy running-config startup-config	(Optional) Saves your entries in the configuration file.

You can only delete a global macro-applied configuration on a switch by entering the **no** version of each command in the macro. You can delete a macro-applied configuration on a port by entering the **default interface interface-id** interface configuration command.

This example shows how to display the **cisco-desktop** macro, to apply the macro and to set the access VLAN ID to 25 on an interface:

```
Switch# show parser macro cisco-desktop
-----
Macro name : cisco-desktop
Macro type : default

# Basic interface - Enable data VLAN only
# Recommended value for access vlan (AVID) should not be 1
switchport access vlan $AVID
switchport mode access

# Enable port security limiting port to a single
# MAC address -- that of desktop
switchport port-security
switchport port-security maximum 1

# Ensure port-security age is greater than one minute
# and use inactivity timer
switchport port-security violation restrict
switchport port-security aging time 2
switchport port-security aging type inactivity

# Configure port as an edge network port
spanning-tree portfast
spanning-tree bpduguard enable
-----
Switch# configure terminal
Switch(config)# interface gigabitethernet1/0/4
Switch(config-if)# macro apply cisco-desktop $AVID 25
```




Configuring Cisco IOS Auto Smartport Macros

This chapter describes how to configure macros on the Catalyst 4500 series switches.

This chapter includes the following major sections:

- [About Auto Smartport Macros, page 22-1](#)
- [Configuring Auto Smartport Macros, page 22-3](#)
- [Displaying Auto Smartport, page 22-14](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About Auto Smartport Macros

Auto Smartport macros dynamically configure ports based on the device type detected on the port. When the switch detects a new device on a port, it applies the appropriate Auto Smartport macro. When a link-down event occurs on the port, the switch removes the macro. For example, when you connect a Cisco IP phone to a port, Auto Smartport automatically applies the Cisco IP phone macro. The Cisco IP phone macro enables quality of service (QoS), security features, and a dedicated voice VLAN to ensure proper treatment of delay-sensitive voice traffic.

Auto Smartport uses event triggers to map devices to macros. The most common event triggers are based on Cisco Discovery Protocol (CDP) messages received from connected devices. The detection of a device (Cisco IP phone, Cisco wireless access point, Cisco switch, or Cisco router) invokes an event trigger for that device.



Note

Although Auto SmartPort detects the Cisco switch it does not invoke the event trigger automatically. The event trigger needs to be manually invoked to map the switch to macros.

Link Layer Discovery Protocol (LLDP) is used to detect devices that do not support CDP. Other mechanisms used as event triggers include the 802.1X authentication result and MAC-address learned.

System built-in event triggers exist for various devices based mostly on CDP and LLDP messages (Table 22-1) and some MAC address. (Through Cisco IOS Release 12.2(54)SG, DMP is detected using the MAC address. Starting with Cisco IOS Release 15.0(2)SG, DMP is also detected using CDP.) These triggers are enabled as long as Auto Smartport is enabled.

You can also define your own trigger. User-defined triggers can be CDP/LLDP-based, a group of MAC addresses, or the value of the attribute-value (AV) pair for the **auto-smart-port** keyword.

The Auto Smartport macros are groups of CLI commands. Detection of devices on a port triggers the application of the macro for the device. (For example, detecting a CISCO_PHONE event on a port triggers the switch to apply the commands in the CISCO_PHONE_AUTO_SMARTPORT macro.) System built-in macros exist for various devices, and, by default, system built-in triggers are mapped to the corresponding built-in macros. You can change the mapping of built-in triggers or macros as needed.

A macro basically applies or removes a set of CLIs on an interface based on the link status. In a macro, the link status is checked. If the link is up, then a set of CLIs is applied; if the link is down, the set is removed (the **no** format of the CLIs are applied). The part of the macro that applies the set of CLIs is termed *macro*. The part that removes the CLIs (the **no** format of the CLIs) are termed *antimacro*.

Besides creating user-defined triggers, you can also create user-defined macros and map one to the other among all triggers (both built-in and user-defined) and all macros (both built-in and user-defined). Use the Cisco IOS scripting capability to create the macros. Cisco IOS scripting is a BASH-like language syntax for command automation and variable replacement.

The four detection mechanisms adhere to the following order of priority:

- If 802.1X authentication is configured on a port, an authentication response-based trigger is applied, and other triggers are ignored.
- If 802.1X authentication fails and the CDP/LLDP fallback mechanism is configured, CDP/LLDP triggers for phone devices only; if no fallback mechanism is configured, or a device is not a phone device, nothing is triggered.
- If 802.1X authentication is configured on a port, a MAC address-based trigger is never triggered.
- If 802.1X authentication is not configured on a port, CDP/LLDP has priority over a MAC address-based trigger with a hold-off timer applied for MAC-address based trigger. Between CDP/LLDP, there is no particular order; whichever one arrives first is triggered.

Device Classifier

Starting with Cisco Release IOS XE 3.3.0SG and IOS 15.1(1)SG, the device classifier (DC) feature is enabled by default on the Catalyst 4500 series switch.

The DC collects information from MAC-OUI and protocols such as CDP, LLDP, and DHCP to identify devices. You must enable CDP and LLDP on the switch. To make DHCP options information available to the DC, you must enable the DHCP snooping feature on the switch. The device attributes collected from these protocols are evaluated against a set of profiles available to the DC to find the best match. The best-matched profile is used for device identification.

Devices that do not send CDP, LLDP or DHCP traffic may not be properly identified by the device classifier.

Device-classifier uses profile definitions—built-in and default profiles. The built-in profiles contain the device profiles that are known to the Auto Smartport module, comprising a limited set of Cisco devices. They are built into Cisco IOS and cannot be changed. The default profiles are stored as a text file in nonvolatile storage and allow the DC to identify a much larger set of devices. The default profiles are updated as part of the Cisco IOS archive download.

When a new device is detected, the corresponding shell trigger executes the Auto Smartport configuration macro. Auto Smartport has built-in mappings for a large set of devices. You can use the commands described in the [“Configuring Mapping Between User-Defined Triggers and Built-in Macros” section on page 22-10](#) to create new mappings. You can create the trigger mappings based on the profile name or device name that is provided by the DC.

Device Visibility Mode

The DC function is enabled on the switch by default. You can disable it by using the **no macro auto monitor** global configuration command. The DC feature provides **show** commands to display the devices that are connected to the switch. It also provides information about the physical port to which the device is connected, along with device MAC address and other vendor information. Only directly connected devices, such as another Layer 2 switch, are classified on nonaccess ports. On access ports that are connected to hubs, device classification is limited to 32 devices.

When you enable Auto Smartport, the DC is automatically enabled.

Configuring Auto Smartport Macros

The following topics are included:

- [Enabling Auto Smartport Macros, page 22-3](#)
- [Auto Smartport Configuration Guidelines, page 22-5](#)
- [Configuring Auto Smartport Built-in Macro Parameters, page 22-6](#)
- [Configuring Mapping Between Event Triggers and Built-in Macros, page 22-8](#)
- [Configuring User-Defined Event Triggers, page 22-9](#)
- [Configuring Mapping Between User-Defined Triggers and Built-in Macros, page 22-10](#)
- [Configuring Auto Smartport User-Defined Macros, page 22-11](#)

Enabling Auto Smartport Macros



Note

By default, Auto Smartport is disabled globally. To disable Auto Smartport macros on a specific port, use the **no macro auto global processing** interface command before enabling Auto Smartport globally.

To enable Auto Smartport globally, use the **macro auto global processing** global configuration command.

To enable Auto Smartport macros, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# [no] macro auto global processing [cdp lldp]	Enables Auto Smartport on the switch globally. Note Starting with Release 15.0(2)SG, the fallback option has been deprecated. Use no macro auto global processing to disable Auto Smartport globally. Note The macro auto processing command turns Auto Smartport on or off on the interface level. The default is on.
Step 3	Switch(config)# end	Returns to privileged EXEC mode.
Step 4	Switch# show running-config	Verifies that Auto Smartport is enabled.
Step 5	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Use the **show shell functions** and the **show shell triggers** privileged EXEC command to display the event triggers, the built-in macros, and the built-in macro default values.

This example shows how enable Auto Smartport on the switch and how to disable the feature on a specific interface:

```
Switch(config)# macro auto global processing
Switch(config)# interface interface_id
Switch(config-if)# no macro auto processing
```

Auto Smartport Default Configuration

By default, Cisco IOS shell is enabled and Auto Smartport is disabled globally.

Table 22-1 shows the Auto Smartport built-in event triggers that are embedded in the switch software by default.

Table 22-1 Auto Smartport Built-in Event Trigger Macros

Event Trigger Name	Description
CISCO_PHONE_EVENT	System detects that a phone device is connected to an interface.
CISCO_SWITCH_EVENT	System detects that a switch is connected to an interface.
CISCO_ROUTER_EVENT	System detects that a router is connected to an interface.
CISCO_WIRELESS_AP_EVENT	System detects that a wireless application is connected to an interface.
CISCO_WIRELESS_LIGHTWEIGHT_AP_EVENT	System detects that a wireless lightweight application is connected to an interface.
CISCO_DMP_EVENT	System detects that a digital media player is connected to an interface.
CISCO_IPVSC_EVENT	System detects that an IP video surveillance camera is connected to an interface.

Table 22-2 shows the Auto Smartport built-in macros that are embedded in the switch software.

Table 22-2 Auto Smartport Built-in Macros

Macro Name	Description
CISCO_PHONE_AUTO_SMARTPORT	Use this macro for Cisco IP phone device. It enables QoS, port security, Address Resolution Protocol (ARP) inspection (dynamic ARP inspection), IP source guard, DHCP snooping, storm control and spanning tree protection on the port.
CISCO_SWITCH_AUTO_SMARTPORT	Use this macro to apply the switch macro for Cisco switches. It enables trunking on the port.
CISCO_ROUTER_AUTO_SMARTPORT	Use this macro to apply the router macro for Cisco routers. It enables QoS, trunking, and spanning-tree protection on the port.
CISCO_AP_AUTO_SMARTPORT	Use this macro to apply the wireless access point (AP) macro for Cisco APs. It enables support for an autonomous wireless access point and QoS on the port.
CISCO_LWAP_AUTO_SMARTPORT	Use this macro to apply the lightweight wireless access point macro for Cisco lightweight wireless APs. It enables QoS, port security, dynamic ARP inspection, IP source guard, DHCP snooping, storm control, and spanning tree protection on the port.
CISCO_IP_CAMERA_AUTO_SMARTPORT	Use this macro for a Cisco IP surveillance camera device. It enables QoS, port security, and access VLAN on the port.
CISCO_DMP_AUTOSMARTPORT	Use this macro for a Cisco digital media player device. It enables QoS, port security, and access VLAN on the port.



Note

By default, the built-in event triggers are mapped to the built-in macros.

Auto Smartport Configuration Guidelines

Auto Smartport guidelines include the following:

- To avoid system conflicts when Auto Smartport macros are applied, remove all port configuration except for 802.1X authentication.
- If the macro conflicts with the original configuration, some macro commands might not be applied, or some antimacro commands might not be applied. (The antimacro is the portion of the applied macro that removes it at link down.)



Note

Failure of one command in the macro halts the application of the entire macro.

For example, if 802.1X authentication is enabled, you cannot remove switchport-mode access configuration. You must remove the 802.1X authentication before removing the configuration.

- A port should not be a member of an EtherChannel when applying Auto Smartport macros.
If Auto Smartport is not yet enabled globally, disable Auto Smartport on all the EtherChannel ports before enabling it globally. If Auto Smartport is already enabled, shut down the port and disable it before adding the port to an EtherChannel.



Note

If an Auto Smartport macro is applied on an interface, EtherChannel configuration usually fails because of conflict with the auto-QoS configuration applied by the macro.

- The built-in macro default data VLAN is VLAN 1. The default voice VLAN is VLAN 2. You should modify the built-in macro default values if your switch uses different VLANs. To view all built-in macro default values, use the **show shell functions** privileged EXEC command.
- To detect non-Cisco devices for 802.1X authentication or MAB, configure the RADIUS server to support the Cisco AV pair **auto-smart-port=event trigger**. You must configure a user-defined trigger with the value returned in the AV pair for **auto-smart-port**.
- For stationary devices that do not support CDP, MAB, or 802.1X authentication, such as network printers, we recommend that you disable Auto Smartport on the port.
- If authentication is enabled on a port, the switch ignores CDP unless the **fallback cdp** keyword is in the **macro auto global processing** global configuration command.
- The order of CLI commands within the macro and the corresponding antimacro can differ.
- Before converting a port into an Layer 3 interface, enter the **no macro auto processing** command. This prevents Auto Smartport from applying macros on the interface. If Layer 3 is already configured, enter the **no macro auto processing** command on the Layer 3 interface enable Auto Smartport globally.
- Auto Smartport macros and Smartport cannot coexist on an interface.
- A switch applies a macro in accordance with the LLDP advertisement from the attached device. If the device does not identify itself properly, the wrong macro is applied. Consult the specific device documentation to ensure the device's firmware is current.
- The LWAP's WLC software version must be 6.0.188 (=> Cisco IOS 12.4(21a)JA2) or later to make it detectable as LWAP by AutoSmartport.
- As of Cisco IOS Release 12.2(54)SG, Auto Smartport does not support macros that apply EtherChannel configurations. Interfaces that belong to EtherChannel groups are treated as standard interfaces. You can apply macros on individual interfaces based on the device type but the CLIs in the macro (for example, auto-QoS) might conflict with an EtherChannel configuration. We recommend that you disable Auto Smartport on interfaces belonging to EtherChannels before you enable Auto Smartport globally. If Auto Smartport is already enabled, disable Auto Smartport on the interfaces before configuring EtherChannel.
- When a Cisco switch is detected on the Auto Smartport, you have to manually map the event trigger to either a built-in macro or user-defined macro. You need to also match the event trigger to the device PID.

Configuring Auto Smartport Built-in Macro Parameters

The switch automatically maps from built-in event triggers to built-in macros. You can replace the built-in macro default values with values that are specific to your switch.

To configure Auto Smartport built-in macros parameters, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.

Command	Purpose
Step 2 Switch(config)# macro auto execute <i>event trigger builtin built-in</i> <i>macro name [parameter=value]</i> <i>[parameter=value]</i>	Defines mapping from an event trigger to a built-in macro. Specify an <i>event trigger</i> value: <ul style="list-style-type: none"> CISCO_PHONE_EVENT CISCO_SWITCH_EVENT CISCO_ROUTER_EVENT CISCO_WIRELESS_AP_EVENT CISCO_WIRELESS_LIGHTWEIGHT_AP_EVENT CISCO_DMP_EVENT CISCO_IPVSC_EVENT WORD—Apply a user-defined event trigger. Specify a <i>built-in macro name</i> value: <ul style="list-style-type: none"> CISCO_PHONE_AUTO_SMARTPORT (Optional) Specify the parameter values: \$ACCESS_VLAN=(1) and \$VOICE_VLAN=(2). CISCO_SWITCH_AUTO_SMARTPORT (Optional) Specify the parameter values: \$NATIVE_VLAN=(1). CISCO_ROUTER_AUTO_SMARTPORT (Optional) Specify the parameter values: \$NATIVE_VLAN=(1). CISCO_AP_AUTO_SMARTPORT (Optional) Specify the parameter values: \$NATIVE_VLAN=(1). CISCO_LWAP_AUTO_SMARTPORT (Optional) Specify the parameter values: \$ACCESS_VLAN=(1). CISCO_DMP_AUTO_SMARTPORT CISCO_IP_CAMERA_AUTO_SMARTPORT (Optional) <i>parameter=value</i> —Replace default values that begin with \$. Enter new values in the form of name value pair separated by a space: [<i>name1=value1 name2=value2...</i>]. Default values are shown in parenthesis.
Step 3 Switch(config)# end	Returns to privileged EXEC mode.
Step 4 Switch# show running-config	Verifies your entries.
Step 5 Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

The **no macro auto execute event trigger** {[**builtin** *built-in macro name* [*parameter=value*]] | [*parameter=value*] {*function contents*}} command deletes the mapping.

This example shows how to use two built-in Auto Smartport macros for connecting Cisco switches and Cisco IP phones to the switch. This example modifies the default voice VLAN, access VLAN, and native VLAN for the trunk interface:

```
Switch# configure terminal
Switch(config)# macro auto execute CISCO_PHONE_EVENT builtin CISCO_PHONE_AUTO_SMARTPORT
ACCESS_VLAN=10 VOICE_VLAN=20
Switch(config)#
Switch(config)#
```

```

Switch(config)#!!! the next command enables auto smart ports globally
Switch(config)# macro auto global processing fallback cdp
Switch(config)#
Switch(config)# exit

Switch#
Switch# show running-config interface gigabitethernet2/7
Building configuration...

Current configuration : 284 bytes
!
switchport access vlan 10
switchport mode access
switchport voice vlan 2
switchport port-security maximum 2
switchport port-security
switchport port-security aging time 2
switchport port-security violation restrict
switchport port-security aging type inactivity
auto qos voip cisco-phone
qos trust device cisco-phone
neighbor device type phone
macro description CISCO_PHONE_EVENT
spanning-tree portfast
spanning-tree bpduguard enable
service-policy input AutoQos-VoIP-Input-Cos-Policy
service-policy output AutoQos-VoIP-Output-Policy
end

```



Note

You can also use the **macro auto device** command to simplify changing the parameters for a built-in function for a device type.

Configuring Mapping Between Event Triggers and Built-in Macros



Note

You need to perform this task when a Cisco switch is connected to the Auto Smartport.

To map event trigger to a built-in macros, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# macro auto execute event trigger builtin <i>built-in macro name</i>	Specifies a user-defined event trigger and a macro name. This action configures mapping from an event trigger to a built-in Auto Smartports macro .
Step 3	Switch(config)# macro auto trigger event trigger	Invokes the user-defined event trigger.
Step 4	Switch(config)# device <i>device_ID</i>	Matches the event trigger to the device identifier.
Step 5	Switch(config)# end	Returns to privileged EXEC mode.
Step 6	Switch# show shell triggers	Displays the event triggers on the switch.
Step 7	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to map a event trigger called CISCO_SWITCH_EVENT to the built-in macro CISCO_SWITCH_AUTO_SMARTPORT.

```
Switch(config)# macro auto execute CISCO_SWITCH_EVENT builtin CISCO_SWITCH_AUTO_SMARTPORT
Switch(config)# macro auto trigger CISCO_SWITCH_EVENT
Switch(config)# device cisco WS-C3560CX-8PT-S
Switch(config)# exit
```

Configuring User-Defined Event Triggers

You can configure two types of event triggers: user-defined and MAC address-based.

The following sections describe these triggers:

- [802.1X-Based Event Trigger, page 22-9](#)
- [MAC Address-Based Event Trigger, page 22-10](#)

802.1X-Based Event Trigger

When using MAB or 802.1X authentication to trigger Auto Smartport macros, you need to create an event trigger that corresponds to the Cisco AV pair (**auto-smart-port=event trigger**) sent by the RADIUS server.

To configure an event trigger, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# shell trigger <i>identifier description</i>	Specifies the event trigger identifier and description. The identifier should have no spaces or hyphens between words.
Step 3	Switch(config)# end	Returns to privileged EXEC mode.
Step 4	Switch# show shell triggers	Displays the event triggers on the switch.
Step 5	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Use the **no shell trigger identifier** global configuration command to delete the event trigger.

The following example shows how to define a user-defined trigger:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# shell trigger RADIUS_MAB_EVENT MAC_AuthBypass Event
Switch(config)#
```

MAC Address-Based Event Trigger

To configure a MAC address group as an event trigger, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# macro auto mac-address <i>group</i>	Specifies a group of MAC address as an event trigger. Changes mode to config-mac-addr-grp. You can then add or remove the MAC address or Organizational Unique Identifier (OUI) from the group. The <i>group</i> value defines the user-defined trigger.
Step 3	Switch(config)# end	Returns to privileged EXEC mode.
Step 4	Switch# show shell triggers	Displays the event triggers on the switch.
Step 5	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Use the **no macro auto mac-address-group** *grp_name* to delete the event trigger.

Configuring Mapping Between User-Defined Triggers and Built-in Macros

You need to map the user-defined trigger to either a built-in macro or user-defined macro.

To map a user-defined trigger to a built-in macros, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# macro auto execute event trigger builtin <i>built-in macro</i> <i>name</i> [<i>parameter=value</i>] [<i>parameter=value</i>]	Specifies a user-defined event trigger and a macro name. This action replaces built-in macro default values, and configures mapping from an event trigger to a built-in Auto Smartport macros. Note When performing a mapping, you must provide parameter values. For example, you must specify \$ACCESS_VLAN=(1) and \$VOICE_VLAN=(2) for the macro CISCO_PHONE_AUTO_SMARTPORT.
Step 3	Switch(config)# end	Returns to privileged EXEC mode.
Step 4	Switch# show shell triggers	Displays the event triggers on the switch.
Step 5	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to map a user-defined event trigger called RADIUS_MAB_EVENT to the built-in macro CISCO_PHONE_AUTO_SMARTPORT with access VLAN set to 10, and how to verify the entries.

This procedure shows how to map a user-defined trigger to a built-in macro:

-
- Step 1** Connect the device to a MAB-enabled switch port.
 - Step 2** On the RADIUS server, set the attribute-value pair to auto-smart-port=RADIUS_MAB_EVENT.

Step 3 On the switch, create the event trigger RADIUS_MAB_EVENT.

The switch recognizes the attribute-value pair=RADIUS_MAB_EVENT response from the RADIUS server and applies the macro CISCO_PHONE_AUTO_SMARTPORT, as in the following example:

```
Switch(config)# macro auto execute RADIUS_MAB_EVENT builtin CISCO_PHONE_AUTO_SMARTPORT
ACCESS_VLAN=10
Switch(config)# exit
Switch# show shell triggers
User defined triggers
-----
Trigger Id: RADIUS_MAB_EVENT
Trigger description: MAC_AuthBypass Event
Trigger environment:
Trigger mapping function: CISCO_PHONE_AUTO_SMARTPORT
<output truncated>
```

Configuring Auto Smartport User-Defined Macros

The Cisco IOS shell provides basic scripting capabilities for configuring the user-defined Auto Smartport macros. These macros can contain multiple lines and can include any CLI command. You can also define variable substitution, conditionals, functions, and triggers within the macro.

Inside a user-defined macro, besides parameters specified through **macro auto execute trigger parameter-name=value ..**, you also can use the following variables published by EEM ([Table 22-3](#)):

Table 22-3 Variables Published by EEM

Parameter Name	Meaning
\$INTERFACE	Name of the interface where the trigger event is detected.
\$LINKUP	Indicates whether the interface is up or down (true/false).
\$TRIGGER	Name of the trigger event that is raised (for example, CISCO_PHONE_EVENT).
\$AUTH_ENABLED	Indicates whether 802.1X authentication is configured on the interface (true/false).

To map an event trigger to a user-defined macro, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# macro auto execute event trigger [parameter=value] {function contents}	Specifies a user-defined macro that maps to an event trigger. Specify an <i>event trigger</i> value: <ul style="list-style-type: none"> • CISCO_PHONE_EVENT • CISCO_SWITCH_EVENT • CISCO_ROUTER_EVENT • CISCO_WIRELESS_AP_EVENT • CISCO_WIRELESS_LIGHTWEIGHT_AP_EVENT • WORD Applies a user-defined event trigger. • CISCO_DMP_EVENT • CISCO_IPVSC_EVENT <i>function contents</i> —Specifies a user-defined macro to associate with the trigger. Enter the macro contents within braces. Begin the Cisco IOS shell commands with the left brace and end the command grouping with the right brace. (Optional) <i>parameter=value</i> —Replaces default values that begin with \$, enter new values in the form of name value pair separated by a space: [name1=value1 name2=value2...].
Step 3	Switch(config)# end	Returns to privileged EXEC mode.
Step 4	Switch# show running-config	Verifies your entries.
Step 5	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to map a user-defined event trigger called Cisco Digital Media Player (DMP) to a user-defined macro.

-
- Step 1** Connect the DMP to an 802.1X- or MAB-enabled switch port.
- Step 2** On the RADIUS server, set the attribute-value pair to **auto-smart-port** =MY_MEDIAPLAYER_EVENT.
- Step 3** On the switch, create the event trigger CISCO_DMP_EVENT, and map it to the user-defined macro commands shown below.

The switch recognizes the attribute-value pair=CISCO_DMP_EVENT response from the RADIUS server and applies the macro associated with this event trigger.

The following example shows the macro portion of the automacro:

```
Switch(config)# shell trigger CISCO_DMP_EVENT Cisco DMP player
Switch(config)# macro auto execute CISCO_DMP_EVENT {
if [[ $LINKUP -eq YES ]]; then
conf t
    interface $INTERFACE
        macro description $TRIGGER
        switchport access vlan 1
        switchport mode access
        switchport port-security
        switchport port-security maximum 1
        switchport port-security violation restrict
        switchport port-security aging time 2
        switchport port-security aging type inactivity
        spanning-tree portfast
        spanning-tree bpduguard enable
    exit
fi
```

The following represents the anti-macro portion of the automacro:

```
if [[ $LINKUP -eq NO ]]; then
conf t
interface $INTERFACE
    no macro description $TRIGGER
    no switchport access vlan 1
    if [[ $AUTH_ENABLED -eq NO ]]; then
        no switchport mode access
    fi
    no switchport port-security
    no switchport port-security maximum 1
    no switchport port-security violation restrict
    no switchport port-security aging time 2
    no switchport port-security aging type inactivity
    no spanning-tree portfast
    no spanning-tree bpduguard enable
    exit
fi
}
Switch(config)# end
```

Table 22-4 lists the supported shell keywords you can apply in your macros and antimacro statements.

Table 22-4 Supported Cisco IOS Shell Keywords

Command	Description
{	Begin the command grouping.
}	End the command grouping.
[[Use as a conditional construct.
]]	Use as a conditional construct.
else	Use as a conditional construct.
-eq	Use as a conditional construct.
fi	Use as a conditional construct.
if	Use as a conditional construct.
then	Use as a conditional construct.
-z	Use as a conditional construct.

Table 22-4 Supported Cisco IOS Shell Keywords (continued)

Command	Description
\$	Variables that begin with the \$ character are replaced with a parameter value.
#	Use the # character to enter comment text.

Table 22-5 lists the shell keywords that are not supported in macros and antimacros.

Table 22-5 Unsupported Cisco IOS Shell Reserved Keywords

Command	Description
	Pipeline.
case	Conditional construct.
esac	Conditional construct.
for	Looping construct.
function	Shell function.
in	Conditional construct.
select	Conditional construct.
time	Pipeline.
until	Looping construct.
while	Looping construct.

Displaying Auto Smartport

To display the Auto Smartport and static Smartport macros, use one or more of the privileged EXEC commands in Table 22-6.

Table 22-6 Commands for Displaying *Auto Smartport* and *Static Smartport* Macros

Command	Purpose
<code>show macro auto monitor clients</code>	Displays the clients using the device classifier facility on the switch.
<code>show macro auto monitor device</code>	Displays the devices connected to a switch and their associated properties.
<code>show macro auto monitor type</code>	Displays all the device types recognized by the device classifier.
<code>show parser macro</code>	Displays all static Smartport macros.
<code>show parser macro name <i>macro-name</i></code>	Displays a specific static Smartport macro.
<code>show parser macro brief</code>	Displays the static Smartport macro names.
<code>show parser macro description [<i>interface interface-id</i>]</code>	Displays the static Smartport macro description for all interfaces or for a specified interface.
<code>show shell</code>	Displays information about Auto Smartport event triggers and macros.

This example shows how to use the **show macro auto monitor device** privileged EXEC command with the optional **mac-address** keyword to view summary information about the connected device with the specified MAC address:

```
Switch# show macro auto monitor device mac-address 001f.9e90.1250
MAC_Address      Port_Id      Profile Name
=====
001f.9e90.1250   Gi1/0/4      Cisco-AP-Aironet-1130
=====
```

This example shows how to use the **show macro auto monitor type** privileged EXEC command with no optional keywords to view the devices recognized by the device classifier:

```
Switch# show macro auto monitor type table
Valid      Type      Profile Name      min Conf      ID
=====
Valid      Default   Apple-Device      10            0
Valid      Default   Aruba-Device      10            1
Valid      Default   Avaya-Device      10            2
Valid      Default   Avaya-IP-Phone    20            3
Valid      Default   BlackBerry         20            4
Valid      Default   Cisco-Device       10            5
Valid      Default   Cisco-IP-Phone     20            6
Valid      Default   Cisco-IP-Phone-7902 70            7
Valid      Default   Cisco-IP-Phone-7905 70            8
Valid      Default   Cisco-IP-Phone-7906 70            9
Valid      Default   Cisco-IP-Phone-7910 70           10
Valid      Default   Cisco-IP-Phone-7911 70           11
Valid      Default   Cisco-IP-Phone-7912 70           12
Valid      Default   Cisco-IP-Phone-7940 70           13
Valid      Default   Cisco-IP-Phone-7941 70           14
Valid      Default   Cisco-IP-Phone-7942 70           15
Valid      Default   Cisco-IP-Phone-7945 70           16
Valid      Default   Cisco-IP-Phone-7945G 70           17
Valid      Default   Cisco-IP-Phone-7960 70           18
Valid      Default   Cisco-IP-Phone-7961 70           19
Valid      Default   Cisco-IP-Phone-7962 70           20
Valid      Default   Cisco-IP-Phone-7965 70           21
Valid      Default   Cisco-IP-Phone-7970 70           22
Valid      Default   Cisco-IP-Phone-7971 70           23
Valid      Default   Cisco-IP-Phone-7975 70           24
Valid      Default   Cisco-IP-Phone-7985 70           25
Valid      Default   Cisco-IP-Phone-9971 70           26
Valid      Default   Cisco-WLC-2100-Series 40           27
Valid      Default   DLink-Device       10           28
Valid      Default   Enterasys-Device   10           29
Valid      Default   HP-Device           10           30
Valid      Default   HP-JetDirect-Printer 30           31
Valid      Default   Lexmark-Device      10           32
Valid      Default   Lexmark-Printer-E260dn 30           33
Valid      Default   Microsoft-Device    10           34
Valid      Default   Netgear-Device      10           35
Valid      Default   NintendoWII         10           36
Valid      Default   Nortel-Device       10           37
Valid      Default   Nortel-IP-Phone-2000-Series 20           38
Valid      Default   SonyPS3             10           39
Valid      Default   XBOX360             20           40
Valid      Default   Xerox-Device        10           41
Valid      Default   Xerox-Printer-Phaser3250 30           42
Valid      Default   Aruba-AP            20           43
Valid      Default   Cisco-Access-Point  10           44
Valid      Default   Cisco-IP-Conference-Station-7935 70           45
Valid      Default   Cisco-IP-Conference-Station-7936 70           46
Valid      Default   Cisco-IP-Conference-Station-7937 70           47
Valid      Default   DLink-DAP-1522      20           48
```

Valid	Default	Cisco-AP-Aironet-1130	30	49
Valid	Default	Cisco-AP-Aironet-1240	30	50
Valid	Default	Cisco-AP-Aironet-1250	30	51
Valid	Default	Cisco-AIR-LAP	25	52
Valid	Default	Cisco-AIR-LAP-1130	30	53
Valid	Default	Cisco-AIR-LAP-1240	50	54
Valid	Default	Cisco-AIR-LAP-1250	50	55
Valid	Default	Cisco-AIR-AP	25	56
Valid	Default	Cisco-AIR-AP-1130	30	57
Valid	Default	Cisco-AIR-AP-1240	50	58
Valid	Default	Cisco-AIR-AP-1250	50	59
Invalid	Default	Sun-Workstation	10	60
Valid	Default	Linksys-Device	20	61
Valid	Default	LinksysWAP54G-Device	30	62
Valid	Default	HTC-Device	10	63
Valid	Default	MotorolaMobile-Device	10	64
Valid	Default	VMWare-Device	10	65
Valid	Default	ISE-Appliance	10	66
Valid	Built-in	Cisco-Device	10	0
Valid	Built-in	Cisco-Router	10	1
Valid	Built-in	Router	10	2
Valid	Built-in	Cisco-IP-Camera	10	3
Valid	Built-in	Cisco-IP-Camera-2xxx	30	4
Valid	Built-in	Cisco-IP-Camera-2421	50	5
Valid	Built-in	Cisco-IP-Camera-2500	50	6
Valid	Built-in	Cisco-IP-Camera-2520	50	7
Valid	Built-in	Cisco-IP-Camera-2530	50	8
Valid	Built-in	Cisco-IP-Camera-4xxx	50	9
Valid	Built-in	Cisco-Transparent-Bridge	8	10
Valid	Built-in	Transparent-Bridge	8	11
Valid	Built-in	Cisco-Source-Bridge	10	12
Valid	Built-in	Cisco-Switch	10	13
Valid	Built-in	Cisco-IP-Phone	20	14
Valid	Built-in	IP-Phone	20	15
Valid	Built-in	Cisco-DMP	10	16
Valid	Built-in	Cisco-DMP-4305G	70	17
Valid	Built-in	Cisco-DMP-4310G	70	18
Valid	Built-in	Cisco-DMP-4400G	70	19
Valid	Built-in	Cisco-WLC-2100-Series	40	20
Valid	Built-in	Cisco-Access-Point	10	21
Valid	Built-in	Cisco-AIR-LAP	30	22
Valid	Built-in	Cisco-AIR-AP	30	23
Valid	Built-in	Linksys-Device	20	24

This example shows how to use the **show shell triggers** privileged EXEC command to view the event triggers in the switch software:

```
Switch# show shell triggers
```

```
User defined triggers
```

```
-----
```

```
Built-in triggers
```

```
-----
```

```
Trigger Id: CISCO_PHONE_EVENT
```

```
Trigger description: Event for ip-phone macro
```

```
Trigger environment: ACCESS_VLAN=1 VOICE_VLAN=2
```

```
Trigger mapping function: CISCO_PHONE_AUTO_SMARTPORT
```

```
Trigger Id: CISCO_ROUTER_EVENT
```

```
Trigger description: Event for router macro
```

```
Trigger environment: NATIVE_VLAN=1
```

```
Trigger mapping function: CISCO_ROUTER_AUTO_SMARTPORT
```

```
Trigger Id: CISCO_SWITCH_EVENT
```

```

Trigger description: Event for switch macro
Trigger environment: NATIVE_VLAN=1
Trigger mapping function: CISCO_SWITCH_AUTO_SMARTPORT

Trigger Id: CISCO_WIRELESS_AP_EVENT
Trigger description: Event for Wireless Access Point macro
Trigger environment: NATIVE_VLAN=1
Trigger mapping function: CISCO_AP_AUTO_SMARTPORT

Trigger Id: CISCO_WIRELESS_LIGHTWEIGHT_AP_EVENT
Trigger description: Event for Wireless Lightweight Access Point macro
Trigger environment: NATIVE_VLAN=1
Trigger mapping function: CISCO_LWAP_AUTO_SMARTPORT

```

This example shows how to use the **show shell functions** privileged EXEC command to view the built-in macros in the switch software:

```

Switch# show shell functions
#User defined functions:

#Built-in functions:
function CISCO_AP_AUTO_SMARTPORT () {
    if [[ $LINKUP -eq YES ]]; then
        conf t
            interface $INTERFACE
                macro description $TRIGGER
                switchport trunk encapsulation dot1q
                switchport trunk native vlan $NATIVE_VLAN
                switchport trunk allowed vlan ALL
                switchport mode trunk
                switchport nonegotiate
                auto qos voip trust
                mls qos trust cos
            exit
        end
    fi
    if [[ $LINKUP -eq NO ]]; then
        conf t
            interface $INTERFACE
                no macro description
                no switchport nonegotiate
                no switchport trunk native vlan $NATIVE_VLAN
                no switchport trunk allowed vlan ALL
                no auto qos voip trust
                no mls qos trust cos
                if [[ $AUTH_ENABLED -eq NO ]]; then
                    no switchport mode
                    no switchport trunk encapsulation
                fi
            fi
        exit
    end
fi
}

function CISCO_SWITCH_AUTO_SMARTPORT () {
    if [[ $LINKUP -eq YES ]]; then
        conf t
            interface $INTERFACE
                macro description $TRIGGER
                auto qos voip trust
                switchport trunk encapsulation dot1q
                switchport trunk native vlan $NATIVE_VLAN
                switchport trunk allowed vlan ALL
                switchport mode trunk
            exit
        end
    fi
}

```

```
        exit
    end
else
    conf t
        interface $INTERFACE
            no macro description
            no auto qos voip trust
            no switchport mode trunk
            no switchport trunk encapsulation dot1q
            no switchport trunk native vlan $NATIVE_VLAN
            no switchport trunk allowed vlan ALL
        exit
    end
fi
}

<output truncated>
```



Configuring STP and MST

This chapter describes how to configure the Spanning Tree Protocol (STP). This chapter also describes how to configure the IEEE 802.1s Multiple Spanning Tree (MST) protocol. MST is an IEEE standard derived from Cisco's proprietary Multi-Instance Spanning-Tree Protocol (MISTP) implementation. With MST, you can map a single spanning-tree instance to several VLANs.

It includes the following major sections:

- [About STP, page 23-1](#)
- [Default STP Configuration, page 23-7](#)
- [Configuring STP, page 23-7](#)
- [About MST, page 23-22](#)
- [MST Configuration Restrictions and Guidelines, page 23-28](#)
- [Configuring MST, page 23-28](#)
- [About MST-to-PVST+ Interoperability \(PVST+ Simulation\), page 23-35](#)
- [Configuring PVST+ Simulation, page 23-36](#)
- [About Detecting Unidirectional Link Failure, page 23-40](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About STP

STP is a Layer 2 link management protocol that provides path redundancy while preventing undesirable loops in the network. For a Layer 2 Ethernet network to function properly, only one active path can exist between any two stations. A loop-free subset of a network topology is called a spanning tree. The operation of a spanning tree is transparent to end stations, which cannot detect whether they are connected to a single LAN segment or a switched LAN of multiple segments.

Beginning in Cisco IOS Release 15.2(4)E and Cisco IOS Release 3.8.0E, Catalyst 4500 series switches by default, use STP (the IEEE 802.1w RSTP) on all VLANs. By default, a single spanning tree runs on each configured VLAN (provided you do not manually disable the spanning tree). You can enable and disable a spanning tree on a per-VLAN basis.

When you create fault-tolerant internetworks, you must have a loop-free path between all nodes in a network. The spanning tree algorithm calculates the best loop-free path throughout a switched Layer 2 network. Switches send and receive spanning tree frames at regular intervals. The switches do not forward these frames, but use the frames to construct a loop-free path.

Multiple active paths between end stations cause loops in the network. If a loop exists in the network, end stations might receive duplicate messages and switches might learn end station MAC addresses on multiple Layer 2 interfaces. These conditions result in an unstable network.

A spanning tree defines a tree with a root switch and a loop-free path from the root to all switches in the Layer 2 network. A spanning tree forces redundant data paths into a standby (blocked) state. If a network segment in the spanning tree fails and a redundant path exists, the spanning tree algorithm recalculates the spanning tree topology and activates the standby path.

When two ports on a switch are part of a loop, the spanning tree port priority and port path cost setting determine which port is put in the forwarding state and which port is put in the blocking state. The spanning tree port priority value represents the location of an interface in the network topology and how well located it is to pass traffic. The spanning tree port path cost value represents media speed.

These sections describe STP:

- [Understanding the Bridge ID, page 23-2](#)
- [Bridge Protocol Data Units, page 23-3](#)
- [Election of the Root Bridge, page 23-4](#)
- [STP Timers, page 23-4](#)
- [Creating the STP Topology, page 23-5](#)
- [STP Port States, page 23-5](#)
- [MAC Address Allocation, page 23-6](#)
- [STP and IEEE 802.1Q Trunks, page 23-6](#)
- [Per-VLAN Rapid Spanning Tree, page 23-6](#)

Understanding the Bridge ID

Each VLAN on each network device has a unique 64-bit bridge ID consisting of a bridge priority value, an extended system ID, and an STP MAC address allocation.

Bridge Priority Value

The bridge priority value determines whether a given redundant link is given priority and considered part of a given span in a spanning tree. Preference is given to lower values, and if you want to manually configure a preference, assign a lower bridge priority value to a link than to its redundant possibility. With Cisco IOS releases prior to 12.1(12c)EW, the bridge priority is a 16-bit value (see [Table 23-1](#)). With Cisco IOS Release 12.1(12c)EW and later releases, the bridge priority is a 4-bit value when the extended system ID is enabled (see [Table 23-2](#)). See the “[Configuring the Bridge Priority of a VLAN](#)” section on [page 23-17](#).

Extended System ID

Extended system IDs are VLAN IDs between 1025 and 4096. Cisco IOS Releases 12.1(12c)EW and later releases support a 12-bit extended system ID field as part of the bridge ID (see [Table 23-2](#)). Chassis that support only 64 MAC addresses always use the 12-bit extended system ID. On chassis that support 1024 MAC addresses, you can enable use of the extended system ID. STP uses the VLAN ID as the extended system ID. See the “[Enabling the Extended System ID](#)” section on page 23-9.

Table 23-1 Bridge Priority Value with the Extended System ID Disabled

Bridge Priority Value															
Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

Table 23-2 Bridge Priority Value and Extended System ID with the Extended System ID Enabled

Bridge Priority Value				Extended System ID (Set Equal to the VLAN ID)											
Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
32768	16384	8192	4096	VLAN ID											

STP MAC Address Allocation

A Catalyst 4500 series switch chassis has either 64 or 1024 MAC addresses available to support software features like STP. Enter the **show module** command to view the MAC address range on your chassis.

Cisco IOS Release 12.1(12c)EW and later releases support chassis with 64 or 1024 MAC addresses. For chassis with 64 MAC addresses, STP uses the extended system ID plus a MAC address to make the bridge ID unique for each VLAN.

Earlier releases support chassis with 1024 MAC addresses. With earlier releases, STP uses one MAC address per-VLAN to make the bridge ID unique for each VLAN.

Bridge Protocol Data Units

The following elements determine the stable active spanning tree topology of a switched network:

- The unique bridge ID (bridge priority and MAC address) associated with each VLAN on each switch
- The spanning tree path cost (or bridge priority value) to the root bridge
- The port identifier (port priority and MAC address) associated with each Layer 2 interface

Bridge protocol data units (BPDUs) contain information about the transmitting bridge and its ports, including the bridge and MAC addresses, bridge priority, port priority, and path cost. The system computes the spanning tree topology by transmitting BPDUs among connecting switches, and in one direction from the root switch. Each configuration BPDU contains at least the following:

- The unique bridge ID of the switch that the transmitting switch believes to be the root switch
- The spanning tree path cost to the root
- The bridge ID of the transmitting bridge
- The age of the message

- The identifier of the transmitting port
- Values for the *hello*, *forward delay*, and *max-age* protocol timers

When a switch transmits a BPDU frame, all switches connected to the LAN on which the frame is transmitted receive the BPDU. When a switch receives a BPDU, it does not forward the frame but instead uses the information in the frame to calculate a BPDU and, if the topology changes, initiate a BPDU transmission.

A BPDU exchange results in the following:

- One switch is elected as the root bridge.
- The shortest distance to the root bridge is calculated for each switch based on the path cost.
- A designated bridge for each LAN segment is selected. It is the switch closest to the root bridge through which frames are forwarded to the root.
- A root port is selected. It is the port providing the best path from the bridge to the root bridge.
- Ports included in the spanning tree are selected.

Election of the Root Bridge

For each VLAN, the switch with the highest bridge priority (the lowest numerical priority value) is elected as the root bridge. If all switches are configured with the default priority value (32,768), the switch with the lowest MAC address in the VLAN becomes the root bridge.

The spanning tree root bridge is the logical center of the spanning tree topology in a switched network. All paths that are not required to reach the root bridge from anywhere in the switched network are placed in spanning tree blocking mode.

A spanning tree uses the information provided by BPDUs to elect the root bridge and root port for the switched network, as well as the root port and designated port for each switched segment.

STP Timers

Table 23-3 describes the STP timers that affect the performance of the entire spanning tree.

Table 23-3 **Spanning Tree Protocol Timers**

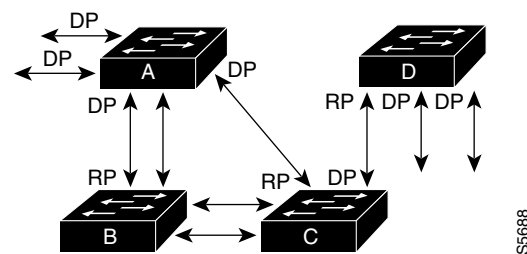
Variable	Description
<i>hello_time</i>	Determines how often the switch broadcasts hello messages to other switches.
<i>forward_time</i>	Determines how long each of the listening and learning states last before the port begins forwarding.
<i>max_age</i>	Determines the amount of time that protocol information received on a port is stored by the switch.

Creating the STP Topology

The goal of the spanning tree algorithm is to make the most direct link the root port. When the spanning tree topology is calculated based on default parameters, the path between source and destination end stations in a switched network might not be optimal according to link speed. For instance, connecting higher-speed links to a port that has a higher number than the current root port can cause a root-port change.

In [Figure 23-1](#), Switch A is elected as the root bridge. (This could happen if the bridge priority of all the switches is set to the default value [32,768] and Switch A has the lowest MAC address.) However, due to traffic patterns, the number of forwarding ports, or link types, Switch A might not be the ideal root bridge. By increasing the STP port priority (lowering the numerical value) of the ideal switch so that it becomes the root bridge, you force a spanning tree recalculation to form a new spanning tree topology with the ideal switch as the root.

Figure 23-1 Spanning Tree Topology



RP = Root Port
DP = Designated Port

For example, assume that one port on Switch B is a fiber-optic link, and another port on Switch B (an unshielded twisted-pair [UTP] link) is the root port. Network traffic might be more efficient over the high-speed fiber-optic link. By changing the spanning tree port priority on the fiber-optic port to a higher priority (lower numerical value) than the priority set for the root port, the fiber-optic port becomes the new root port.

STP Port States

Propagation delays can occur when protocol information passes through a switched LAN. As a result, topology changes can take place at different times and at different places in a switched network. When a Layer 2 interface transitions directly from nonparticipation in the spanning tree topology to the forwarding state, it can create temporary data loops. Ports must wait for new topology information to propagate through the switched LAN before starting to forward frames. They must allow the frame lifetime to expire for frames that have been forwarded under the old topology.

Each Layer 2 interface on a switch that uses spanning tree exists in one of the following five states:

- **Blocking**—In this state, the Layer 2 interface does not participate in frame forwarding.
- **Listening**—This state is the first transitional state after the blocking state when spanning tree determines that the Layer 2 interface should participate in frame forwarding.
- **Learning**—In this state, the Layer 2 interface prepares to participate in frame forwarding.
- **Forwarding**—In this state, the Layer 2 interface forwards frames.

- Disabled—In this state, the Layer 2 interface does not participate in spanning tree and does not forward frames.

MAC Address Allocation

The supervisor engine has a pool of 1024 MAC addresses that are used as the bridge IDs for the VLAN spanning trees. Use the **show module** command to view the MAC address range (allocation range for the supervisor) that the spanning tree uses for the algorithm.

MAC addresses for the Catalyst 4506 switch are allocated sequentially, with the first MAC address in the range assigned to VLAN 1, the second MAC address in the range assigned to VLAN 2, and so forth. For example, if the MAC address range is 00-e0-1e-9b-2e-00 to 00-e0-1e-9b-31-ff, the VLAN 1 bridge ID is 00-e0-1e-9b-2e-00, the VLAN 2 bridge ID is 00-e0-1e-9b-2e-01, the VLAN 3 bridge ID is 00-e0-1e-9b-2e-02, and so on. On other Catalyst 4500 series platforms, all VLANs map to the same MAC address rather than mapping to separate MAC addresses.

STP and IEEE 802.1Q Trunks

802.1Q VLAN trunks impose some limitations on the spanning tree strategy for a network. In a network of Cisco switches connected through 802.1Q trunks, the switches maintain one instance of spanning tree for each VLAN allowed on the trunks. However, non-Cisco 802.1Q switches maintain only one instance of spanning tree for all VLANs allowed on the trunks.

When you connect a Cisco switch to a non-Cisco device (that supports 802.1Q) through an 802.1Q trunk, the Cisco switch combines the spanning tree instance of the 802.1Q native VLAN of the trunk with the spanning tree instance of the non-Cisco 802.1Q switch. However, all per-VLAN spanning tree information is maintained by Cisco switches separated by a network of non-Cisco 802.1Q switches. The non-Cisco 802.1Q network separating the Cisco switches is treated as a single trunk link between the switches.

**Note**

For more information on 802.1Q trunks, see [Chapter 19, “Configuring Layer 2 Ethernet Interfaces.”](#)

Per-VLAN Rapid Spanning Tree

Beginning in Cisco IOS XE Release 3.8.0E and Cisco IOS Release 15.2(4)E, Per-VLAN Rapid Spanning Tree (PVRST+) is the default STP mode on Catalyst 4500 Series Switches.

Per-VLAN Rapid Spanning Tree is the same as PVST+, although PVRST+ utilizes a rapid STP based on IEEE 802.1w rather than 802.1D to provide faster convergence. PVRST+ uses roughly the same configuration as PVST+ and needs only minimal configuration. In PVRST+, dynamic CAM entries are flushed immediately on a per-port basis when any topology change is made. UplinkFast and BackboneFast are enabled but not active in this mode, because the functionality is built into the Rapid STP. PVRST+ provides for rapid recovery of connectivity following the failure of a bridge, bridge port, or LAN.

Like per-VLAN Spanning Tree (PVST+), per-VLAN Rapid Spanning Tree (PVRST+) instances are equal to the number of vlans configured on the switch and can go up to a maximum of 4094 instances.

For enabling information, see “Enabling Per-VLAN Rapid Spanning Tree” on page 20.

Default STP Configuration

Table 23-4 shows the default spanning tree configuration.

Table 23-4 Spanning Tree Default Configuration Values

Feature	Default Value
Enable state	Spanning tree enabled for all VLANs
Bridge priority value	32,768
Spanning tree port priority value (configurable on a per-interface basis—used on interfaces configured as Layer 2 access ports)	128
Spanning tree port cost (configurable on a per-interface basis—used on interfaces configured as Layer 2 access ports)	<ul style="list-style-type: none"> 10-Gigabit Ethernet: 2 Gigabit Ethernet: 4 Fast Ethernet: 19
Spanning tree VLAN port priority value (configurable on a per-VLAN basis—used on interfaces configured as Layer 2 trunk ports)	128
Spanning tree VLAN port cost (configurable on a per-VLAN basis—used on interfaces configured as Layer 2 trunk ports)	<ul style="list-style-type: none"> 10-Gigabit Ethernet: 2 Gigabit Ethernet: 4 Fast Ethernet: 19
Hello time	2 sec
Forward delay time	15 sec
Maximum aging time	20 sec

Configuring STP

The following sections describe how to configure spanning tree on VLANs:

- [Enabling STP, page 23-8](#)
- [Enabling the Extended System ID, page 23-9](#)
- [Configuring the Root Bridge, page 23-9](#)
- [Configuring a Secondary Root Switch, page 23-12](#)
- [Configuring STP Port Priority, page 23-13](#)
- [Configuring STP Port Cost, page 23-15](#)
- [Configuring the Bridge Priority of a VLAN, page 23-17](#)
- [Configuring the Hello Time, page 23-17](#)
- [Configuring the Maximum Aging Time for a VLAN, page 23-18](#)
- [Configuring the Forward-Delay Time for a VLAN, page 23-19](#)
- [Disabling Spanning Tree Protocol, page 23-20](#)
- [Enabling Per-VLAN Rapid Spanning Tree, page 23-20](#)

**Note**

The spanning tree commands described in this chapter can be configured on any interface except those configured with the **no switchport** command.

Enabling STP

**Note**

By default, spanning tree is enabled on all the VLANs.

You can enable a spanning tree on a per-VLAN basis. The switch maintains a separate instance of spanning tree for each VLAN (except on VLANs on which you have disabled a spanning tree).

To enable a spanning tree on a per-VLAN basis, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# spanning-tree vlan <i>vlan_ID</i>	Enables spanning tree for VLAN <i>vlan_id</i> . The <i>vlan_ID</i> value can range from 1 to 4094.
Step 3	Switch(config)# end	Exits configuration mode.
Step 4	Switch# show spanning-tree vlan <i>vlan_ID</i>	Verifies that spanning tree is enabled.

This example shows how to enable a spanning tree on VLAN 200:

```
Switch# configure terminal
Switch(config)# spanning-tree vlan 200
Switch(config)# end
Switch#
```

**Note**

Because spanning tree is enabled by default, entering a **show running** command to view the resulting configuration does not display the command you entered to enable spanning tree.

This example shows how to verify that spanning tree is enabled on VLAN 200:

```
Switch# show spanning-tree vlan 200
```

```
VLAN200 is executing the ieee compatible Spanning Tree protocol
Bridge Identifier has priority 32768, address 0050.3e8d.6401
Configured hello time 2, max age 20, forward delay 15
Current root has priority 16384, address 0060.704c.7000
Root port is 264 (FastEthernet5/8), cost of root path is 38
Topology change flag not set, detected flag not set
Number of topology changes 0 last change occurred 01:53:48 ago
Times: hold 1, topology change 24, notification 2
      hello 2, max age 14, forward delay 10
Timers: hello 0, topology change 0, notification 0
```

```
Port 264 (FastEthernet5/8) of VLAN200 is forwarding
Port path cost 19, Port priority 128, Port Identifier 129.9.
Designated root has priority 16384, address 0060.704c.7000
Designated bridge has priority 32768, address 00e0.4fac.b000
Designated port id is 128.2, designated path cost 19
Timers: message age 3, forward delay 0, hold 0
Number of transitions to forwarding state: 1
```

```
BPDU: sent 3, received 3417
```

```
Switch#
```

Enabling the Extended System ID



Note

The extended system ID is enabled permanently on chassis that support 64 MAC addresses.

Use the **spanning-tree extend system-id** command to enable the extended system ID on chassis that support 1024 MAC addresses. See the [“Understanding the Bridge ID” section on page 23-2](#).

To enable the extended system ID, perform this task:

	Command	Purpose
Step 1	Switch(config) # spanning-tree extend system-id	Enables the extended system ID.
		Disables the extended system ID. Note You cannot disable the extended system ID on chassis that support 64 MAC addresses or when you have configured extended range VLANs (see Table 23-4 on page 23-7).
Step 2	Switch(config) # end	Exits configuration mode.
Step 3	Switch# show spanning-tree vlan <i>vlan_ID</i>	Verifies the configuration.



Note

When you enable or disable the extended system ID, the bridge IDs of all active STP instances are updated, which might change the spanning tree topology.

This example shows how to enable the extended system ID:

```
Switch# configure terminal
Switch(config)# spanning-tree extend system-id
Switch(config)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show spanning-tree summary | include extended
Extended system ID is enabled.
```

Configuring the Root Bridge

A Catalyst 4500 series switch maintains an instance of spanning tree for each active VLAN configured on the switch. A bridge ID, consisting of the bridge priority and the bridge MAC address, is associated with each instance. For each VLAN, the switch with the lowest bridge ID becomes the root bridge for that VLAN. Whenever the bridge priority changes, the bridge ID also changes, resulting in the recomputation of the root bridge for the VLAN.

To configure a switch to become the root bridge for the specified VLAN, use the **spanning-tree vlan *vlan-ID* root** command to modify the bridge priority from the default value (32,768) to a significantly lower value. The bridge priority for the specified VLAN is set to 8192 if this value causes the switch to become the root for the VLAN. If any bridge for the VLAN has a priority lower than 8192, the switch sets the priority to 1 less than the lowest bridge priority.

For example, assume that all the switches in the network have the bridge priority for VLAN 100 set to the default value of 32,768. Entering the **spanning-tree vlan 100 root primary** command on a switch sets the bridge priority for VLAN 100 to 8192, causing this switch to become the root bridge for VLAN 100.



Note

The root switch for each instance of spanning tree should be a backbone or distribution switch. Do not configure an access switch as the spanning tree primary root.

Use the **diameter** keyword to specify the Layer 2 network diameter (the maximum number of bridge hops between any two end stations in the network). When you specify the network diameter, a switch automatically picks an optimal hello time, forward delay time, and maximum age time for a network of that diameter. This action can significantly reduce the spanning tree convergence time.

Use the **hello-time** keyword to override the automatically calculated hello time.



Note

We recommend that you avoid manually configuring the hello time, forward delay time, and maximum age time after configuring the switch as the root bridge.

To configure a switch as the root switch, perform this task:

	Command	Purpose
Step 1	Switch(config)# [no] spanning-tree vlan <i>vlan_ID</i> root primary [diameter <i>hops</i> [hello-time <i>seconds</i>]]	Configures a switch as the root switch.
		Use the no keyword to restore the defaults.
Step 2	Switch(config)# end	Exits configuration mode.

This example shows how to configure a switch as the root bridge for VLAN 10, with a network diameter of 4:

```
Switch# configure terminal
Switch(config)# spanning-tree vlan 10 root primary diameter 4
Switch(config)# end
Switch#
```

This example shows how the configuration changes when a switch becomes a spanning tree root. This configuration is the one before the switch becomes the root for VLAN 1:

```
Switch# show spanning-tree vlan 1

VLAN1 is executing the ieee compatible Spanning Tree protocol
Bridge Identifier has priority 32768, address 0030.94fc.0a00
Configured hello time 2, max age 20, forward delay 15
Current root has priority 32768, address 0001.6445.4400
Root port is 323 (FastEthernet6/3), cost of root path is 19
Topology change flag not set, detected flag not set
Number of topology changes 2 last change occurred 00:02:19 ago
    from FastEthernet6/1
Times: hold 1, topology change 35, notification 2
    hello 2, max age 20, forward delay 15
```



```

Timers:hello 0, topology change 0, notification 0, aging 300

Port 323 (FastEthernet6/3) of VLAN1 is forwarding
  Port path cost 19, Port priority 128, Port Identifier 129.67.
  Designated root has priority 32768, address 0001.6445.4400
  Designated bridge has priority 32768, address 0001.6445.4400
  Designated port id is 129.67, designated path cost 0
  Timers:message age 2, forward delay 0, hold 0
  Number of transitions to forwarding state:1
  BPDU:sent 3, received 91

Port 324 (FastEthernet6/4) of VLAN1 is blocking
  Port path cost 19, Port priority 128, Port Identifier 129.68.
  Designated root has priority 32768, address 0001.6445.4400
  Designated bridge has priority 32768, address 0001.6445.4400
  Designated port id is 129.68, designated path cost 0
  Timers:message age 2, forward delay 0, hold 0
  Number of transitions to forwarding state:0
  BPDU:sent 1, received 89

```

You can set the switch as the root:

```

Switch# configure terminal
Switch(config)# spanning-tree vlan 1 root primary
Switch(config)# spanning-tree vlan 1 root primary
VLAN 1 bridge priority set to 8192
VLAN 1 bridge max aging time unchanged at 20
VLAN 1 bridge hello time unchanged at 2
VLAN 1 bridge forward delay unchanged at 15
Switch(config)# end

```

This configuration is the one after the switch becomes the root:

```

Switch# show spanning-tree vlan 1

VLAN1 is executing the ieee compatible Spanning Tree protocol
  Bridge Identifier has priority 8192, address 0030.94fc.0a00
  Configured hello time 2, max age 20, forward delay 15
  We are the root of the spanning tree
  Topology change flag set, detected flag set
  Number of topology changes 3 last change occurred 00:00:09 ago
  Times: hold 1, topology change 35, notification 2
         hello 2, max age 20, forward delay 15
  Timers:hello 0, topology change 25, notification 0, aging 15

Port 323 (FastEthernet6/3) of VLAN1 is forwarding
  Port path cost 19, Port priority 128, Port Identifier 129.67.
  Designated root has priority 8192, address 0030.94fc.0a00
  Designated bridge has priority 8192, address 0030.94fc.0a00
  Designated port id is 129.67, designated path cost 0
  Timers:message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state:1
  BPDU:sent 9, received 105

Port 324 (FastEthernet6/4) of VLAN1 is listening
  Port path cost 19, Port priority 128, Port Identifier 129.68.
  Designated root has priority 8192, address 0030.94fc.0a00
  Designated bridge has priority 8192, address 0030.94fc.0a00
  Designated port id is 129.68, designated path cost 0
  Timers:message age 0, forward delay 5, hold 0
  Number of transitions to forwarding state:0
  BPDU:sent 6, received 102

Switch#

```



Because the bridge priority is now set at 8192, this switch becomes the root of the spanning tree.

Configuring a Secondary Root Switch

When you configure a switch as the secondary root, the spanning tree bridge priority is modified from the default value (32,768) to 16,384. This means that the switch is likely to become the root bridge for the specified VLANs if the primary root bridge fails (assuming the other switches in the network use the default bridge priority of 32,768).

You can run this command on more than one switch to configure multiple backup root switches. Use the same network diameter and hello time values that you used when configuring the primary root switch.



We recommend that you avoid manually configuring the hello time, forward delay time, and maximum age time after configuring the switch as the root bridge.

To configure a switch as the secondary root switch, perform this task:

	Command	Purpose
Step 1	Switch(config)# [no] spanning-tree vlan vlan_ID root secondary [diameter hops [hello-time seconds]]	Configures a switch as the secondary root switch. Use the no keyword to restore the defaults.
Step 2	Switch(config)# end	Exits configuration mode.

This example shows how to configure the switch as the secondary root switch for VLAN 10, with a network diameter of 4:

```
Switch# configure terminal
Switch(config)# spanning-tree vlan 10 root secondary diameter 4
VLAN 10 bridge priority set to 16384
VLAN 10 bridge max aging time set to 14
VLAN 10 bridge hello time unchanged at 2
VLAN 10 bridge forward delay set to 10
Switch(config)# end
Switch#
```

This example shows how to verify the configuration of VLAN 1:

```
Switch#sh spanning-tree vlan 1

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    32768
             Address     0003.6b10.e800
             This bridge is the root
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32768
             Address     0003.6b10.e800
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300

Interface                Role Sts Cost          Prio.Nbr Status
-----
Fa3/1                    Desg FWD 19          128.129 P2p
```

```

Fa3/2          Desg FWD 19          128.130 P2p
Fa3/48         Desg FWD 19          128.176 Edge P2p

Switch#

```

Configuring STP Port Priority

In the event of a loop, a spanning tree considers port priority when selecting an interface to put into the forwarding state. You can assign higher priority values to interfaces that you want a spanning tree to select first and lower priority values to interfaces that you want a spanning tree to select last. If all interfaces have the same priority value, a spanning tree puts the interface with the lowest interface number in the forwarding state and blocks other interfaces. The possible priority range is 0 through 240, configurable in increments of 16 (the default is 128).



Note

The Cisco IOS software uses the port priority value when the interface is configured as an access port and uses VLAN port priority values when the interface is configured as a trunk port.

To configure the spanning tree port priority of an interface, perform this task:

	Command	Purpose
Step 1	Switch(config)# interface {{ fastethernet gigabitethernet tengigabitethernet } slot/port} { port-channel port_channel_number}	Specifies an interface to configure.
Step 2	Switch(config-if)# [no] spanning-tree port-priority port_priority	Configures the port priority for an interface. The <i>port_priority</i> value can be from 0 to 240, in increments of 16. Use the no keyword to restore the defaults.
Step 3	Switch(config-if)# [no] spanning-tree vlan vlan_ID port-priority port_priority	Configures the VLAN port priority for an interface. The <i>port_priority</i> value can be from 0 to 240, in increments of 16. Use the no keyword to restore the defaults.
Step 4	Switch(config-if)# end	Exits configuration mode.
Step 5	Switch# show spanning-tree interface {{ fastethernet gigabitethernet } slot/port} { port-channel port_channel_number} show spanning-tree vlan vlan_ID	Verifies the configuration.

This example shows how to configure the spanning tree port priority of a Fast Ethernet interface:

```

Switch# configure terminal
Switch(config)# interface fastethernet 5/8
Switch(config-if)# spanning-tree port-priority 100
Switch(config-if)# end
Switch#

```

This example shows how to verify the configuration of a Fast Ethernet interface when it is configured as an access port:

```

Switch# show spanning-tree interface fastethernet 3/1

Vlan                Role Sts Cost        Prio.Nbr Status

```

```

-----
VLAN0001      Desg FWD 19      128.129 P2p
VLAN1002      Desg FWD 19      128.129 P2p
VLAN1003      Desg FWD 19      128.129 P2p
VLAN1004      Desg FWD 19      128.129 P2p
VLAN1005      Desg FWD 19      128.129 P2p
Switch#

```

This example shows how to display the details of the interface configuration when the interface is configured as an access port:

```

Switch# show spanning-tree interface fastethernet 3/1 detail
Port 129 (FastEthernet3/1) of VLAN0001 is forwarding
  Port path cost 19, Port priority 128, Port Identifier 128.129.
  Designated root has priority 32768, address 0003.6b10.e800
  Designated bridge has priority 32768, address 0003.6b10.e800
  Designated port id is 128.129, designated path cost 0
  Timers:message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state:1
  Link type is point-to-point by default
  BPDU:sent 187, received 1

Port 129 (FastEthernet3/1) of VLAN1002 is forwarding
  Port path cost 19, Port priority 128, Port Identifier 128.129.
  Designated root has priority 32768, address 0003.6b10.ebe9
  Designated bridge has priority 32768, address 0003.6b10.ebe9
  Designated port id is 128.129, designated path cost 0
  Timers:message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state:1
  Link type is point-to-point by default
  BPDU:sent 94, received 2

Port 129 (FastEthernet3/1) of VLAN1003 is forwarding
  Port path cost 19, Port priority 128, Port Identifier 128.129.
  Designated root has priority 32768, address 0003.6b10.ebea
  Designated bridge has priority 32768, address 0003.6b10.ebea
  Designated port id is 128.129, designated path cost 0
  Timers:message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state:1
  Link type is point-to-point by default
  BPDU:sent 94, received 2

Port 129 (FastEthernet3/1) of VLAN1004 is forwarding
  Port path cost 19, Port priority 128, Port Identifier 128.129.
  Designated root has priority 32768, address 0003.6b10.ebeb
  Designated bridge has priority 32768, address 0003.6b10.ebeb
  Designated port id is 128.129, designated path cost 0
  Timers:message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state:1
  Link type is point-to-point by default
  BPDU:sent 95, received 2

Port 129 (FastEthernet3/1) of VLAN1005 is forwarding
  Port path cost 19, Port priority 128, Port Identifier 128.129.
  Designated root has priority 32768, address 0003.6b10.ebec
  Designated bridge has priority 32768, address 0003.6b10.ebec
  Designated port id is 128.129, designated path cost 0
  Timers:message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state:1
  Link type is point-to-point by default
  BPDU:sent 95, received 2
Switch#

```

**Note**

The **show spanning-tree port-priority** command displays only information for ports with an active link. If there is no port with an active link, enter a **show running-config interface** command to verify the configuration.

This example shows how to configure the spanning tree VLAN port priority of a Fast Ethernet interface:

```
Switch# configure terminal
Switch(config)# interface fastethernet 5/8
Switch(config-if)# spanning-tree vlan 200 port-priority 64
Switch(config-if)# end
Switch#
```

This example shows how to verify the configuration of VLAN 200 on the interface when it is configured as a trunk port:

```
Switch# show spanning-tree vlan 200
<...output truncated...>

Port 264 (FastEthernet5/8) of VLAN200 is forwarding
Port path cost 19, Port priority 64, Port Identifier 129.8.
  Designated root has priority 32768, address 0010.0d40.34c7
  Designated bridge has priority 32768, address 0010.0d40.34c7
  Designated port id is 128.1, designated path cost 0
  Timers: message age 2, forward delay 0, hold 0
  Number of transitions to forwarding state: 1
  BPDU: sent 0, received 13513

<...output truncated...>
Switch#
```

Configuring STP Port Cost

The default value for spanning tree port path cost is derived from the interface media speed. In the event of a loop, spanning tree considers port cost when selecting an interface to put into the forwarding state. You can assign lower cost values to interfaces that you want spanning tree to select first, and higher cost values to interfaces that you want spanning tree to select last. If all interfaces have the same cost value, spanning tree puts the interface with the lowest interface number in the forwarding state and blocks other interfaces. The possible cost range is 1 through 200,000,000 (the default is media-specific).

Spanning tree uses the port cost value when the interface is configured as an access port and uses VLAN port cost values when the interface is configured as a trunk port.

To configure the spanning tree port cost of an interface, perform this task:

	Command	Purpose
Step 1	Switch(config)# interface {{fastethernet gigabitethernet tengigabitethernet} slot/port} {port-channel port_channel_number}	Specifies an interface to configure.
Step 2	Switch(config-if)# [no] spanning-tree cost port_cost	Configures the port cost for an interface. The <i>port_cost</i> value can be from 1 to 200,000,000. Use the no keyword to restore the defaults.

	Command	Purpose
Step 3	Switch(config-if)# [no] spanning-tree vlan <i>vlan_ID cost port_cost</i>	Configures the VLAN port cost for an interface. The <i>port_cost</i> value can be from 1 to 200,000,000. Use the no keyword to restore the defaults.
Step 4	Switch(config-if)# end	Exits configuration mode.
Step 5	Switch# show spanning-tree interface { fastethernet gigabitethernet } <i>slot/port</i> { port-channel <i>port_channel_number</i> } show spanning-tree vlan <i>vlan_ID</i>	Verifies the configuration.

This example shows how to change the spanning tree port cost of a Fast Ethernet interface:

```
Switch# configure terminal
Switch(config)# interface fastethernet 5/8
Switch(config-if)# spanning-tree cost 18
Switch(config-if)# end
Switch#
```

This example shows how to verify the configuration of the interface when it is configured as an access port:

```
Switch# show spanning-tree interface fastethernet 5/8
Port 264 (FastEthernet5/8) of VLAN200 is forwarding
  Port path cost 18, Port priority 100, Port Identifier 129.8.
  Designated root has priority 32768, address 0010.0d40.34c7
  Designated bridge has priority 32768, address 0010.0d40.34c7
  Designated port id is 128.1, designated path cost 0
  Timers: message age 2, forward delay 0, hold 0
  Number of transitions to forwarding state: 1
  BPDU: sent 0, received 13513
Switch#
```

This example shows how to configure the spanning tree VLAN port cost of a Fast Ethernet interface:

```
Switch# configure terminal
Switch(config)# interface fastethernet 5/8
Switch(config-if)# spanning-tree vlan 200 cost 17
Switch(config-if)# end
Switch#
```

This example shows how to verify the configuration of VLAN 200 on the interface when it is configured as a trunk port:

```
Switch# show spanning-tree vlan 200
<...output truncated...>
Port 264 (FastEthernet5/8) of VLAN200 is forwarding
  Port path cost 17, Port priority 64, Port Identifier 129.8.
  Designated root has priority 32768, address 0010.0d40.34c7
  Designated bridge has priority 32768, address 0010.0d40.34c7
  Designated port id is 128.1, designated path cost 0
  Timers: message age 2, forward delay 0, hold 0
  Number of transitions to forwarding state: 1
  BPDU: sent 0, received 13513

<...output truncated...>
Switch#
```

**Note**

The **show spanning-tree** command displays only information for ports with an active link (green light is on). If there is no port with an active link, you can issue a **show running-config** command to confirm the configuration.

Configuring the Bridge Priority of a VLAN

**Note**

Exercise care when configuring the bridge priority of a VLAN. In most cases, we recommend that you enter the **spanning-tree vlan *vlan_ID* root primary** and the **spanning-tree vlan *vlan_ID* root secondary** commands to modify the bridge priority.

To configure the spanning tree bridge priority of a VLAN, perform this task:

	Command	Purpose
Step 1	Switch(config)# [no] spanning-tree vlan <i>vlan_ID</i> priority <i>bridge_priority</i>	Configures the bridge priority of a VLAN. The <i>bridge_priority</i> value can be from 1 to 65,534. Use the no keyword to restore the defaults.
Step 2	Switch(config)# end	Exits configuration mode.
Step 3	Switch# show spanning-tree vlan <i>vlan_ID</i> bridge [brief]	Verifies the configuration.

This example shows how to configure the bridge priority of VLAN 200 to 33,792:

```
Switch# configure terminal
Switch(config)# spanning-tree vlan 200 priority 33792
Switch(config)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show spanning-tree vlan 200 bridge brief
Vlan                Bridge ID      Hello Time  Max Age  Fwd Delay  Protocol
-----
VLAN200             33792 0050.3e8d.64c8  2    20    15    ieee
Switch#
```

Configuring the Hello Time

**Note**

Exercise care when configuring the hello time. In most cases, we recommend that you use the **spanning-tree vlan *vlan_ID* root primary** and the **spanning-tree vlan *vlan_ID* root secondary** commands to modify the hello time.

To configure the spanning tree hello time of a VLAN, perform this task:

	Command	Purpose
Step 1	Switch(config)# [no] spanning-tree vlan <i>vlan_ID</i> hello-time <i>hello_time</i>	Configures the hello time of a VLAN. The <i>hello_time</i> value can be from 1 to 10 seconds. Use the no keyword to restore the defaults.
Step 2	Switch(config)# end	Exits configuration mode.
Step 3	Switch# show spanning-tree vlan <i>vlan_ID</i> bridge [brief]	Verifies the configuration.

This example shows how to configure the hello time for VLAN 200 to 7 seconds:

```
Switch# configure terminal
Switch(config)# spanning-tree vlan 200 hello-time 7
Switch(config)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show spanning-tree vlan 200 bridge brief

Vlan                Bridge ID      Hello Time  Max Age  Fwd Delay  Protocol
-----
VLAN200             49152 0050.3e8d.64c8    7       20       15    ieee
Switch#
```

Configuring the Maximum Aging Time for a VLAN



Note

Exercise care when configuring aging time. In most cases, we recommend that you use the **spanning-tree vlan *vlan_ID* root primary** and the **spanning-tree vlan *vlan_ID* root secondary** commands to modify the maximum aging time.

To configure the spanning tree maximum aging time for a VLAN, perform this task:

	Command	Purpose
Step 1	Switch(config)# [no] spanning-tree vlan <i>vlan_ID</i> max-age <i>max_age</i>	Configures the maximum aging time of a VLAN. The <i>max_age</i> value can be from 6 to 40 seconds. Use the no keyword to restore the defaults.
Step 2	Switch(config)# end	Exits configuration mode.
Step 3	Switch# show spanning-tree vlan <i>vlan_ID</i> bridge [brief]	Verifies the configuration.

This example shows how to configure the maximum aging time for VLAN 200 to 36 seconds:

```
Switch# configure terminal
Switch(config)# spanning-tree vlan 200 max-age 36
Switch(config)# end
Switch#
```


This example shows how to verify the configuration:

```
Switch# show spanning-tree vlan 200 bridge brief
Hello Max Fwd
Vlan          Bridge ID    Time Age Delay Protocol
-----
VLAN200      49152 0050.3e8d.64c8  2  36  15  ieee
Switch#
```

Configuring the Forward-Delay Time for a VLAN



Note

Exercise care when configuring forward-delay time. In most cases, we recommend that you use the **spanning-tree vlan *vlan_ID* root primary** and the **spanning-tree vlan *vlan_ID* root secondary** commands to modify the forward delay time.

To configure the spanning tree forward delay time for a VLAN, perform this task:

	Command	Purpose
Step 1	Switch(config)# [no] spanning-tree vlan <i>vlan_ID</i> forward-time <i>forward_time</i>	Configures the forward time of a VLAN. The <i>forward_time</i> value can be from 4 to 30 seconds. Use the no keyword to restore the defaults.
Step 2	Switch(config)# end	Exits configuration mode.
Step 3	Switch# show spanning-tree vlan <i>vlan_ID</i> bridge [brief]	Verifies the configuration.

This example shows how to configure the forward delay time for VLAN 200 to 21 seconds:

```
Switch# configure terminal
Switch(config)# spanning-tree vlan 200 forward-time 21
Switch(config)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show spanning-tree vlan 200 bridge brief
Hello Max Fwd
Vlan          Bridge ID    Time Age Delay Protocol
-----
VLAN200      49152 0050.3e8d.64c8  2  20  21  ieee
Switch#
```

This example shows how to display spanning tree information for the bridge:

```
Switch# show spanning-tree bridge
Hello Max Fwd
Vlan          Bridge ID    Time Age Dly Protocol
-----
VLAN200      49152 0050.3e8d.64c8  2  20  15  ieee
VLAN202      49152 0050.3e8d.64c9  2  20  15  ieee
VLAN203      49152 0050.3e8d.64ca  2  20  15  ieee
VLAN204      49152 0050.3e8d.64cb  2  20  15  ieee
VLAN205      49152 0050.3e8d.64cc  2  20  15  ieee
VLAN206      49152 0050.3e8d.64cd  2  20  15  ieee
Switch#
```

Disabling Spanning Tree Protocol

To disable spanning tree on a per-VLAN basis, perform this task:

	Command	Purpose
Step 1	Switch(config)# no spanning-tree vlan <i>vlan_ID</i>	Disables spanning tree on a per-VLAN basis.
Step 2	Switch(config)# end	Exits configuration mode.
Step 3	Switch# show spanning-tree vlan <i>vlan_ID</i>	Verifies that spanning tree is disabled.

This example shows how to disable spanning tree on VLAN 200:

```
Switch# configure terminal
Switch(config)# no spanning-tree vlan 200
Switch(config)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show spanning-tree vlan 200
Spanning tree instance for VLAN 200 does not exist.
Switch#
```

Enabling Per-VLAN Rapid Spanning Tree

Per-VLAN Rapid Spanning Tree (PVRST+) uses the existing PVST+ framework for configuration purposes and for interaction with other features. It also supports some of the PVST+ extensions.



Note

Beginning in Cisco IOS XE Release 3.8.0E and Cisco IOS Release 15.2(4)E, Per-VLAN Rapid Spanning Tree (PVRST+) is the default STP mode on Catalyst 4500 Series Switches.

To enable PVRST+, perform this task:

:

	Command	Purpose
Step 1	Switch(config)# [no] spantree mode rapid-pvst	Enables PVRST+.
Step 2	Switch(config)# interface <i>interface/port</i>	Switches to interface configuration mode.
Step 3	Switch(config-if)# spanning-tree link-type point-to-point	Sets the link-type to point-to-point mode for the port.
Step 4	Switch(config-if)# exit	Exits interface mode.
Step 5	Switch(config)# exit	Exits configuration mode.
Step 6	Switch(config-if)# clear spantree detected-protocols <i>mod/port</i>	Detects any legacy bridges on the port
Step 7	Switch# show spanning-tree summary totals	Verifies the PVRST+ configuration.

The following example shows how to configure PVRST+:

```
Switch# config t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# spanning-tree mode rapid-pvst
Switch(config)# int fa 6/4
```

```
Switch(config-if)# spanning-tree link-type point-to-point
Switch(config-if)# end
Switch(config)# end
Switch#
23:55:32:%SYS-5-CONFIG_I:Configured from console by console
Switch# clear spanning-tree detected-protocols
```

The following example shows how to verify the configuration:

```
Switch# show spanning-tree summary totals
Switch is in rapid-pvst mode
Root bridge for:VLAN0001
Extended system ID          is disabled
Portfast Default            is disabled
PortFast BPDU Guard Default is disabled
Portfast BPDU Filter Default is disabled
Loopguard Default           is disabled
EtherChannel misconfig guard is enabled
UplinkFast                  is disabled
BackboneFast                is disabled
Pathcost method used         is short
Name                         Blocking Listening Learning Forwarding STP Active
-----
1 vlan                       0          0          0          2          2
Switch#
```

Specifying the Link Type

Rapid connectivity is established only on point-to-point links. Spanning tree views a point-to-point link as a segment connecting only two switches running the spanning tree algorithm. Because the switch assumes that all full-duplex links are point-to-point links and that half-duplex links are shared links, you can avoid explicitly configuring the link type. To configure a specific link type, use the **spanning-tree linktype** command.

Restarting Protocol Migration

A switch running both MSTP and RSTP supports a built-in protocol migration process that enables the switch to interoperate with legacy 802.1D switches. If this switch receives a legacy 802.1D configuration BPDU (a BPDU with the protocol version set to 0), it sends only 802.1D BPDUs on that port. When an MSTP switch receives a legacy BPDU, it can also detect the following:

- A port is at the boundary of a region
- An MST BPDU (version 3) that is associated with a different region
- An RST BPDU (version 2)

The switch, however, does not automatically revert to the MSTP mode if it no longer receives 802.1D BPDUs because it cannot determine whether or not the legacy switch has been removed from the link unless the legacy switch is the designated switch. A switch also might continue to assign a boundary role to a port when the switch to which it is connected has joined the region.

To restart the protocol migration process on the entire switch (that is, to force renegotiation with neighboring switches), use the **clear spanning-tree detected-protocols** commands in privileged EXEC mode. To restart the protocol migration process on a specific interface, enter the **clear spanning-tree detected-protocols interface** command in interface-id privileged EXEC mode.

About MST

The following sections describe how MST works on a Catalyst 4500 series switch:

- [IEEE 802.1s MST, page 23-22](#)
- [IEEE 802.1w RSTP, page 23-23](#)
- [MST-to-SST Interoperability, page 23-24](#)
- [Common Spanning Tree, page 23-25](#)
- [MST Instances, page 23-26](#)
- [MST Configuration Parameters, page 23-26](#)
- [MST Regions, page 23-26](#)
- [Message Age and Hop Count, page 23-28](#)

IEEE 802.1s MST

MST extends the IEEE 802.1w rapid spanning tree (RST) algorithm to multiple spanning trees. This extension provides both rapid convergence and load balancing in a VLAN environment. MST converges faster than per-VLAN Spanning Tree Plus (PVST+) and is backward compatible with 802.1D STP, 802.1w (Rapid Spanning Tree Protocol [RSTP]), and the Cisco PVST+ architecture.

MST allows you to build multiple spanning trees over trunks. You can group and associate VLANs to spanning tree instances. Each instance can have a topology independent of other spanning tree instances. This architecture provides multiple forwarding paths for data traffic and enables load balancing. Network fault tolerance is improved because a failure in one instance (forwarding path) does not affect other instances.

In large networks, you can more easily administer the network and use redundant paths by locating different VLAN and spanning tree instance assignments in different parts of the network. A spanning tree instance can exist only on bridges that have compatible VLAN instance assignments. You must configure a set of bridges with the same MST configuration information, which allows them to participate in a specific set of spanning tree instances. Interconnected bridges that have the same MST configuration are referred to as an *MST region*.

MST uses the modified RSTP, MSTP. MST has the following characteristics:

- MST runs a variant of spanning tree called Internal Spanning Tree (IST). IST augments Common Spanning Tree (CST) information with internal information about the MST region. The MST region appears as a single bridge to adjacent single spanning tree (SST) and MST regions.
- A bridge running MST provides interoperability with SST bridges as follows:
 - MST bridges run IST, which augments CST information with internal information about the MST region.
 - IST connects all the MST bridges in the region and appears as a subtree in the CST that includes the whole bridged domain. The MST region appears as a virtual bridge to adjacent SST bridges and MST regions.
 - The Common and Internal Spanning Tree (CIST) is the collection of the following: ISTs in each MST region, the CST that interconnects the MST regions, and the SST bridges. CIST is identical to an IST inside an MST region and identical to a CST outside an MST region. The STP, RSTP, and MSTP together elect a single bridge as the root of the CIST.

- MST establishes and maintains additional spanning trees within each MST region. These spanning trees are termed MST instances (MSTIs). The IST is numbered 0, and the MSTIs are numbered 1, 2, 3, and so on. Any MSTI is local to the MST region and is independent of MSTIs in another region, even if the MST regions are interconnected.

MST instances combine with the IST at the boundary of MST regions to become the CST as follows:

- Spanning tree information for an MSTI is contained in an MSTP record (M-record).

M-records are always encapsulated within MST bridge protocol data units (BPDUs). The original spanning trees computed by MSTP are called M-trees, which are active only within the MST region. M-trees merge with the IST at the boundary of the MST region and form the CST.

- MST provides interoperability with PVST+ by generating PVST+ BPDUs for the non-CST VLANs.
- MST supports some of the PVST+ extensions in MSTP as follows:
 - UplinkFast and BackboneFast are not available in MST mode; they are part of RSTP.
 - PortFast is supported.
 - BPDU filter and BPDU guard are supported in MST mode.
 - Loop guard and root guard are supported in MST. MST preserves the VLAN 1 disabled functionality except that BPDUs are still transmitted in VLAN 1.
 - MST switches operate as if MAC reduction is enabled.
 - For private VLANs (PVLANS), you must map a secondary VLAN to the same instance as the primary.

IEEE 802.1w RSTP

RSTP, specified in 802.1w, supersedes STP specified in 802.1D, but remains compatible with STP. You configure RSTP when you configure the MST feature. For more information, see the [“Configuring MST” section on page 23-28](#).

RSTP provides the structure on which the MST operates, significantly reducing the time to reconfigure the active topology of a network when its physical topology or configuration parameters change. RSTP selects one switch as the root of a spanning-tree-connected active topology and assigns port roles to individual ports of the switch, depending on whether that port is part of the active topology.

RSTP provides rapid connectivity following the failure of a switch, switch port, or a LAN. A new root port and the designated port on the other side of the bridge transition to the forwarding state through an explicit handshake between them. RSTP allows switch port configuration so the ports can transition to forwarding directly when the switch reinitializes.

RSTP provides backward compatibility with 802.1D bridges as follows:

- RSTP selectively sends 802.1D-configured BPDUs and Topology Change Notification (TCN) BPDUs on a per-port basis.
- When a port initializes, the migration delay timer starts and RSTP BPDUs are transmitted. While the migration delay timer is active, the bridge processes all BPDUs received on that port.
- If the bridge receives an 802.1D BPDU after a port’s migration delay timer expires, the bridge assumes it is connected to an 802.1D bridge and starts using only 802.1D BPDUs.
- When RSTP uses 802.1D BPDUs on a port and receives an RSTP BPDU after the migration delay expires, RSTP restarts the migration delay timer and begins using RSTP BPDUs on that port.

RSTP Port Roles

In RSTP, the port roles are defined as follows:

- **Root**—A forwarding port elected for the spanning tree topology.
- **Designated**—A forwarding port elected for every switched LAN segment.
- **Alternate**—An alternate path to the root bridge to that provided by the current root port.
- **Backup**—A backup for the path provided by a designated port toward the leaves of the spanning tree. Backup ports can exist only where two ports are connected together in a loopback mode or bridge with two or more connections to a shared LAN segment.
- **Disabled**—A port that has no role within the operation of spanning tree.

The system assigns port roles as follows:

- A root port or designated port role includes the port in the active topology.
- An alternate port or backup port role excludes the port from the active topology.

RSTP Port States

The port state controls the forwarding and learning processes and provides the values of discarding, learning, and forwarding. [Table 23-5](#) shows the STP port states and RSTP port states.

Table 23-5 Comparison Between STP and RSTP Port States

Operational Status	STP Port State	RSTP Port State	Port Included in Active Topology
Enabled	Blocking ¹	Discarding ²	No
Enabled	Listening	Discarding	No
Enabled	Learning	Learning	Yes
Enabled	Forwarding	Forwarding	Yes
Disabled	Disabled	Discarding	No

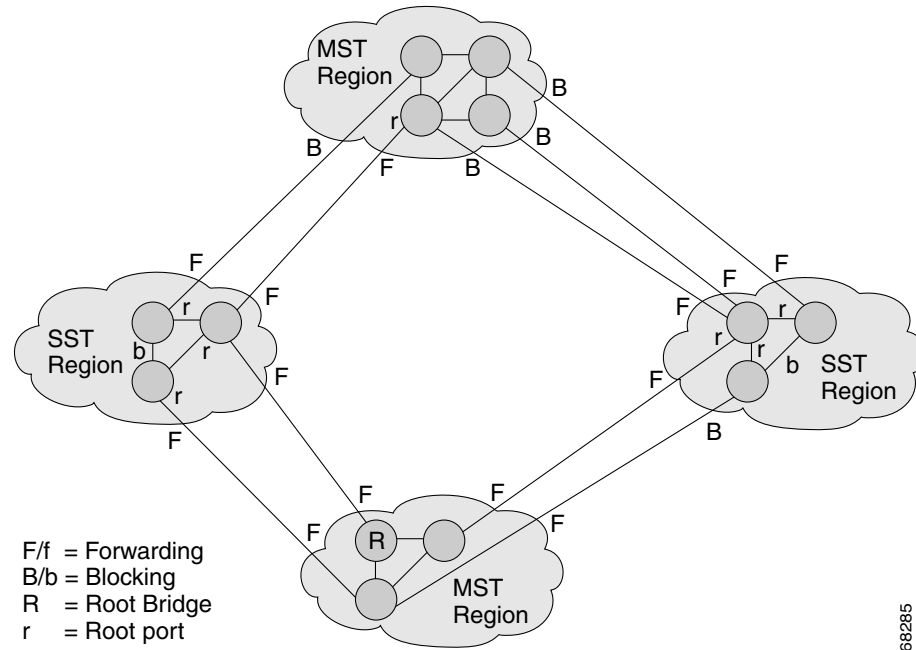
1. IEEE 802.1D port state designation.

2. IEEE 802.1w port state designation. Discarding is the same as blocking in MST.

In a stable topology, RSTP ensures that every root port and designated port transitions to the forwarding state while all alternate ports and backup ports are always in the discarding state.

MST-to-SST Interoperability

A virtual bridged LAN may contain interconnected regions of SST and MST bridges. [Figure 23-2](#) shows this relationship.

Figure 23-2 Network with Interconnected SST and MST Regions

To STP running in the SST region, an MST region appears as a single SST or pseudobridge, which operates as follows:

- Although the values for root identifiers and root path costs match for all BPDUs in all pseudobridges, a pseudobridge differs from a single SST bridge as follows:
 - The pseudobridge BPDUs have different bridge identifiers. This difference does not affect STP operation in the neighboring SST regions because the root identifier and root cost are the same.
 - BPDUs sent from the pseudobridge ports may have significantly different message ages. Because the message age increases by one second for each hop, the difference in the message age is measured in seconds.
- Data traffic from one port of a pseudobridge (a port at the edge of a region) to another port follows a path entirely contained within the pseudobridge or MST region. Data traffic belonging to different VLANs might follow different paths within the MST regions established by MST.
- The system prevents looping by doing either of the following:
 - Blocking the appropriate pseudobridge ports by allowing one forwarding port on the boundary and blocking all other ports.
 - Setting the CST partitions to block the ports of the SST regions.

Common Spanning Tree

CST (802.1Q) is a single spanning tree for all the VLANs. On a Catalyst 4500 series switch running PVST+, the VLAN 1 spanning tree corresponds to CST. On a Catalyst 4500 series switch running MST, IST (instance 0) corresponds to CST.

MST Instances

We support 65 instances including instance 0. Each spanning tree instance is identified by an instance ID that ranges from 0 to 4094. Instance 0 is mandatory and is always present. Rest of the instances are optional.

MST Configuration Parameters

The MST configuration includes these three parts:

- Name—A 32-character string (null padded) that identifies the MST region.
- Revision number—An unsigned 16-bit number that identifies the revision of the current MST configuration.

**Note**

You must set the revision number when required as part of the MST configuration. The revision number is not incremented automatically each time you commit the MST configuration.

- MST configuration table—An array of 4096 bytes. Each byte, interpreted as an unsigned integer, corresponds to a VLAN. The value is the instance number to which the VLAN is mapped. The first byte that corresponds to VLAN 0 and the 4096th byte that corresponds to VLAN 4095 are unused and always set to zero.

You must configure each byte manually. Use SNMP or the CLI to perform the configuration.

MST BPDUs contain the MST configuration ID and the checksum. An MST bridge accepts an MST BPDU only if the MST BPDU configuration ID and the checksum match its own MST region configuration ID and checksum. If either value is different, the MST BPDU is considered to be an SST BPDU.

MST Regions

These sections describe MST regions:

- [MST Region Overview, page 23-26](#)
- [Boundary Ports, page 23-27](#)
- [IST Master, page 23-27](#)
- [Edge Ports, page 23-27](#)
- [Link Type, page 23-28](#)

MST Region Overview

Interconnected bridges that have the same MST configuration are referred to as an MST region. There is no limit on the number of MST regions in the network.

To form an MST region, bridges can be either of the following:

- An MST bridge that is the only member of the MST region.
- An MST bridge interconnected by a LAN. A LAN's designated bridge has the same MST configuration as an MST bridge. All the bridges on the LAN can process MST BPDUs.

If you connect two MST regions with different MST configurations, the MST regions do the following:

- Load balance across redundant paths in the network. If two MST regions are redundantly connected, all traffic flows on a single connection with the MST regions in a network.
- Provide an RSTP handshake to enable rapid connectivity between regions. However, the handshaking is not as fast as between two bridges. To prevent loops, all the bridges inside the region must agree upon the connections to other regions. This situation introduces a delay. We do not recommend partitioning the network into a large number of regions.

Boundary Ports

A boundary port is a port that connects to a LAN, the designated bridge of which is either an SST bridge or a bridge with a different MST configuration. A designated port knows that it is on the boundary if it detects an STP bridge or receives an agreement message from an RST or MST bridge with a different configuration.

At the boundary, the role of MST ports do not matter; their state is forced to be the same as the IST port state. If the boundary flag is set for the port, the MSTP Port Role selection mechanism assigns a port role to the boundary and the same state as that of the IST port. The IST port at the boundary can take up any port role except a backup port role.

IST Master

The IST master of an MST region is the bridge with the lowest bridge identifier and the least path cost to the CST root. If an MST bridge is the root bridge for CST, then it is the IST master of that MST region. If the CST root is outside the MST region, then one of the MST bridges at the boundary is selected as the IST master. Other bridges on the boundary that belong to the same region eventually block the boundary ports that lead to the root.

If two or more bridges at the boundary of a region have an identical path to the root, you can set a slightly lower bridge priority to make a specific bridge the IST master.

The root path cost and message age inside a region stay constant, but the IST path cost is incremented and the IST remaining hops are decremented at each hop. Enter the **show spanning-tree mst** command to display the information about the IST master, path cost, and remaining hops for the bridge.

Edge Ports

A port that is connected to a nonbridging device (for example, a host or a switch) is an edge port. A port that connects to a hub is also an edge port if the hub or any LAN that is connected to it does not have a bridge. An edge port can start forwarding as soon as the link is up.

MST requires that you configure each port connected to a host. To establish rapid connectivity after a failure, you need to block the non-edge designated ports of an intermediate bridge. If the port connects to another bridge that can send back an agreement, then the port starts forwarding immediately. Otherwise, the port needs twice the forward delay time to start forwarding again. You must explicitly configure the ports that are connected to the hosts and switches as edge ports while using MST.

To prevent a misconfiguration, the PortFast operation is turned off if the port receives a BPDU. You can display the configured and operational status of PortFast by using the **show spanning-tree mst interface** command.

Link Type

Rapid connectivity is established only on point-to-point links. You must configure ports explicitly to a host or switch. However, cabling in most networks meets this requirement. By entering the **spanning-tree linktype** command to treating all full-duplex links as point-to-point links, you can avoid explicit configuration.

Message Age and Hop Count

IST and MST instances do not use the message age and maximum age timer settings in the BPDU. IST and MST use a separate hop count mechanism that is very similar to the IP time-to live (TTL) mechanism. You can configure each MST bridge with a maximum hop count. The root bridge of the instance sends a BPDU (or M-record) with the remaining hop count that is equal to the maximum hop count. When a bridge receives a BPDU (or M-record), it decrements the received remaining hop count by one. The bridge discards the BPDU (M-record) and ages out the information held for the port if the count reaches zero after decrementing. The nonroot bridges propagate the decremented count as the remaining hop count in the BPDUs (M-records) they generate.

The message age and maximum age timer settings in the RST portion of the BPDU remain the same throughout the region, and the same values are propagated by the region's designated ports at the boundary.

MST Configuration Restrictions and Guidelines

Follow these restrictions and guidelines to avoid configuration problems:

- Do not disable spanning tree on any VLAN in any of the PVST bridges.
- Do not use PVST bridges as the root of CST.
- Do not connect switches with access links because access links may partition a VLAN.
- Ensure that all PVST root bridges have lower, (numerically higher) priority than the CST root bridge.
- Ensure that trunks carry all of the VLANs mapped to an instance or do not carry any VLANs at all for this instance.
- Complete any MST configuration that incorporates a large number of either existing or new logical VLAN ports during a maintenance window because the complete MST database gets reinitialized for any incremental change (such as adding new VLANs to instances or moving VLANs across instances).

Configuring MST

The following sections describe how to configure MST:

- [Enabling MST, page 23-29](#)

- [Configuring MST Instance Parameters, page 23-30](#)
- [Configuring MST Instance Port Parameters, page 23-31](#)
- [Restarting Protocol Migration, page 23-32](#)
- [Displaying MST Configurations, page 23-32](#)

Enabling MST

To enable and configure MST on a switch, perform this task:

	Command	Purpose
Step 1	Switch(config)# spanning-tree mode mst	Enters MST mode.
Step 2	Switch(config)# spanning-tree mst configuration	Enters MST configuration submenu. Use the no keyword to clear the MST configuration.
Step 3	Switch(config-mst)# show current	Displays the current MST configuration.
Step 4	Switch(config-mst)# name name	Sets the MST region name.
Step 5	Switch(config-mst)# revision revision_number	Sets the MST configuration revision number.
Step 6	Switch(config-mst)# instance instance_number vlan vlan_range	Maps the VLANs to an MST instance. If you do not specify the vlan keyword, use the no keyword to unmap all the VLANs that were mapped to an MST instance. If you specify the vlan keyword, use the no keyword to unmap a specified VLAN from an MST instance.
Step 7	Switch(config-mst)# show pending	Displays the new MST configuration to be applied.
Step 8	Switch(config-mst)# end	Applies the configuration and exit MST configuration submenu.
Step 9	Switch# show spanning-tree mst configuration	Displays the current MST configuration.

This example show how to enable MST:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# spanning-tree mode mst

Switch(config)# spanning-tree mst configuration

Switch(config-mst)# show current
Current MST configuration
Name      []
Revision  0
Instance  Vlan mapped
-----
0          1-4094
-----

Switch(config-mst)# name cisco
Switch(config-mst)# revision 2
Switch(config-mst)# instance 1 vlan 1
Switch(config-mst)# instance 2 vlan 1-1000
Switch(config-mst)# show pending
Pending MST configuration
```

```
Name      [cisco]
Revision  2
Instance  Vlans mapped
-----
0          1001-4094
2          1-1000
-----

Switch(config-mst)# no instance 2
Switch(config-mst)# show pending
Pending MST configuration
Name      [cisco]
Revision  2
Instance  Vlans mapped
-----
0          1-4094
-----

Switch(config-mst)# instance 1 vlan 2000-3000
Switch(config-mst)# no instance 1 vlan 1500
Switch(config-mst)# show pending
Pending MST configuration
Name      [cisco]
Revision  2
Instance  Vlans mapped
-----
0          1-1999,2500,3001-4094
1          2000-2499,2501-3000
-----

Switch(config-mst)# end
Switch(config)# no spanning-tree mst configuration
Switch(config)# end
Switch# show spanning-tree mst configuration
Name      []
Revision  0
Instance  Vlans mapped
-----
0          1-4094
-----
```

Configuring MST Instance Parameters

To configure MST instance parameters, perform this task:

	Command	Purpose
Step 1	Switch(config)# spanning-tree mst X priority Y	Configures the priority for an MST instance.
Step 2	Switch(config)# spanning-tree mst X root [primary secondary]	Configures the bridge as root for an MST instance.
Step 3	Switch(config)# Ctrl-Z	Exits configuration mode.
Step 4	Switch# show spanning-tree mst	Verifies the configuration.

This example shows how to configure MST instance parameters:

```
Switch(config)# spanning-tree mst 1 priority ?
<0-61440>  bridge priority in increments of 4096

Switch(config)# spanning-tree mst 1 priority 1
% Bridge Priority must be in increments of 4096.
% Allowed values are:
```

```

0      4096  8192  12288 16384 20480 24576 28672
32768 36864 40960 45056 49152 53248 57344 61440

Switch(config)# spanning-tree mst 1 priority 49152
Switch(config)#

Switch(config)# spanning-tree mst 0 root primary
mst 0 bridge priority set to 24576
mst bridge max aging time unchanged at 20
mst bridge hello time unchanged at 2
mst bridge forward delay unchanged at 15
Switch(config)# ^Z
Switch#

Switch# show spanning-tree mst

##### MST00          vlans mapped: 11-4094
Bridge      address 00d0.00b8.1400  priority  24576 (24576 sysid 0)
Root        this switch for CST and IST
Configured  hello time 2, forward delay 15, max age 20, max hops 20

Interface    Role Sts Cost      Prio.Nbr Status
-----
Fa4/4        Back BLK 1000      240.196 P2p
Fa4/5        Desg FWD 200000     128.197 P2p
Fa4/48       Desg FWD 200000     128.240 P2p Bound(STP)

##### MST01          vlans mapped: 1-10
Bridge      address 00d0.00b8.1400  priority  49153 (49152 sysid 1)
Root        this switch for MST01

Interface    Role Sts Cost      Prio.Nbr Status
-----
Fa4/4        Back BLK 1000      160.196 P2p
Fa4/5        Desg FWD 200000     128.197 P2p
Fa4/48       Boun FWD 200000     128.240 P2p Bound(STP)

Switch#

```

Configuring MST Instance Port Parameters

To configure MST instance port parameters, perform this task:

	Command	Purpose
Step 1	Switch(config-if)# spanning-tree mst x cost y	Configures the MST instance port cost.
Step 2	Switch(config-if)# spanning-tree mst x port-priority y	Configures the MST instance port priority.
Step 3	Switch(config-if)# Ctrl-Z	Exits configuration mode.
Step 4	Switch# show spanning-tree mst x interface y	Verifies the configuration.

This example shows how to configure MST instance port parameters:

```

Switch(config)# interface fastethernet 4/4
Switch(config-if)# spanning-tree mst 1 ?
cost          Change the interface spanning tree path cost for an instance
port-priority Change the spanning tree port priority for an instance

Switch(config-if)# spanning-tree mst 1 cost 1234567

```

```
Switch(config-if)# spanning-tree mst 1 port-priority 240
Switch(config-if)# ^Z

Switch# show spanning-tree mst 1 interface fastethernet 4/4

FastEthernet4/4 of MST01 is backup blocking
Edge port:no (default) port guard :none (default)
Link type:point-to-point (auto) bpdu filter:disable (default)
Boundary :internal bpdu guard :disable (default)
Bpdus (MRecords) sent 125, received 1782

Instance Role Sts Cost Prio.Nbr Vlans mapped
-----
1 Back BLK 1234567 240.196 1-10

Switch#
```

Restarting Protocol Migration

RSTP and MST have built-in compatibility mechanisms that allow them to interact properly with other regions or other versions of IEEE spanning-tree. For example, an RSTP bridge connected to a legacy bridge can send 802.1D BPDUs on one of its ports. Similarly, when an MST bridge receives a legacy BPDU or an MST BPDU associated with a different region, it is also to detect that a port is at the boundary of a region.

Unfortunately, these mechanisms cannot always revert to the most efficient mode. For example, an RSTP bridge designated for a legacy 802.1D stays in 802.1D mode even after the legacy bridge has been removed from the link. Similarly, an MST port still assumes that it is a boundary port when the bridge(s) to which it is connected have joined the same region. To force a Catalyst 4500 series switch to renegotiate with the neighbors (that is, to restart protocol migration), you must enter the **clear spanning-tree detected-protocols** command, as follows:

```
Switch# clear spanning-tree detected-protocols fastethernet 4/4
Switch#
```

Displaying MST Configurations

To display MST configurations, perform this task:

	Command	Purpose
Step 1	Switch# show spanning-tree mst configuration	Displays the active region configuration information.
Step 2	Switch# show spanning-tree mst [detail]	Displays detailed MST protocol information.
Step 3	Switch# show spanning-tree mst <i>instance-id</i> [detail]	Displays information about a specific MST instance.
Step 4	Switch# show spanning-tree mst interface <i>interface</i> [detail]	Displays information for a given port.
Step 5	Switch# show spanning-tree mst <i>instance-id</i> interface <i>interface</i> [detail]	Displays MST information for a given port and a given instance.
Step 6	Switch# show spanning-tree vlan <i>vlan_ID</i>	Displays VLAN information in MST mode.

The following examples show how to display spanning tree VLAN configurations in MST mode:

```
Switch(config)# spanning-tree mst configuration
Switch(config-mst)# instance 1 vlan 1-10
Switch(config-mst)# name cisco
Switch(config-mst)# revision 1
Switch(config-mst)# Ctrl-D
```

```
Switch# show spanning-tree mst configuration
Name      [cisco]
Revision  1
Instance  Vlan mapped
-----
0         11-4094
1         1-10
-----
```

```
Switch# show spanning-tree mst
```

```
##### MST00          vlans mapped: 11-4094
Bridge      address 00d0.00b8.1400 priority 32768 (32768 sysid 0)
Root        address 00d0.004a.3c1c priority 32768 (32768 sysid 0)
            port Fa4/48 path cost 203100
IST master  this switch
Operational hello time 2, forward delay 15, max age 20, max hops 20
Configured  hello time 2, forward delay 15, max age 20, max hops 20
```

Interface	Role	Sts	Cost	Prio.Nbr	Status
Fa4/4	Back	BLK	1000	240.196	P2p
Fa4/5	Desg	FWD	200000	128.197	P2p
Fa4/48	Root	FWD	200000	128.240	P2p Bound(STP)

```
##### MST01          vlans mapped: 1-10
Bridge      address 00d0.00b8.1400 priority 32769 (32768 sysid 1)
Root        this switch for MST01
```

Interface	Role	Sts	Cost	Prio.Nbr	Status
Fa4/4	Back	BLK	1000	240.196	P2p
Fa4/5	Desg	FWD	200000	128.197	P2p
Fa4/48	Boun	FWD	200000	128.240	P2p Bound(STP)

```
Switch# show spanning-tree mst 1
```

```
##### MST01          vlans mapped: 1-10
Bridge      address 00d0.00b8.1400 priority 32769 (32768 sysid 1)
Root        this switch for MST01
```

Interface	Role	Sts	Cost	Prio.Nbr	Status
Fa4/4	Back	BLK	1000	240.196	P2p
Fa4/5	Desg	FWD	200000	128.197	P2p
Fa4/48	Boun	FWD	200000	128.240	P2p Bound(STP)

Switch# **show spanning-tree mst interface fastethernet 4/4**

FastEthernet4/4 of MST00 is backup blocking
 Edge port:no (default) port guard :none (default)
 Link type:point-to-point (auto) bpdu filter:disable (default)
 Boundary :internal bpdu guard :disable (default)
 Bpdus sent 2, received 368

Instance	Role	Sts	Cost	Prio.Nbr	Vlans mapped
0	Back	BLK	1000	240.196	11-4094
1	Back	BLK	1000	240.196	1-10

Switch# **show spanning-tree mst 1 interface fastethernet 4/4**

FastEthernet4/4 of MST01 is backup blocking
 Edge port:no (default) port guard :none (default)
 Link type:point-to-point (auto) bpdu filter:disable (default)
 Boundary :internal bpdu guard :disable (default)
 Bpdus (MRecords) sent 2, received 364

Instance	Role	Sts	Cost	Prio.Nbr	Vlans mapped
1	Back	BLK	1000	240.196	1-10

Switch# **show spanning-tree mst 1 detail**

MST01 vlans mapped: 1-10
 Bridge address 00d0.00b8.1400 priority 32769 (32768 sysid 1)
 Root this switch for MST01

FastEthernet4/4 of MST01 is backup blocking
 Port info port id 240.196 priority 240 cost 1000
 Designated root address 00d0.00b8.1400 priority 32769 cost 0
 Designated bridge address 00d0.00b8.1400 priority 32769 port id 128.197
 Timers:message expires in 5 sec, forward delay 0, forward transitions 0
 Bpdus (MRecords) sent 123, received 1188

FastEthernet4/5 of MST01 is designated forwarding
 Port info port id 128.197 priority 128 cost 200000
 Designated root address 00d0.00b8.1400 priority 32769 cost 0
 Designated bridge address 00d0.00b8.1400 priority 32769 port id 128.197
 Timers:message expires in 0 sec, forward delay 0, forward transitions 1
 Bpdus (MRecords) sent 1188, received 123

FastEthernet4/48 of MST01 is boundary forwarding
 Port info port id 128.240 priority 128 cost 200000
 Designated root address 00d0.00b8.1400 priority 32769 cost 0
 Designated bridge address 00d0.00b8.1400 priority 32769 port id 128.240
 Timers:message expires in 0 sec, forward delay 0, forward transitions 1
 Bpdus (MRecords) sent 78, received 0

Switch# **show spanning-tree vlan 10**

MST01
 Spanning tree enabled protocol mstp
 Root ID Priority 32769
 Address 00d0.00b8.1400
 This bridge is the root
 Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec


```

Bridge ID Priority    32769 (priority 32768 sys-id-ext 1)
Address      00d0.00b8.1400
Hello Time   2 sec   Max Age 20 sec   Forward Delay 15 sec

Interface      Role Sts Cost      Prio.Nbr Status
-----
Fa4/4          Back BLK 1000     240.196 P2p
Fa4/5          Desg FWD 200000   128.197 P2p

Switch# show spanning-tree summary
Root bridge for:MST01
EtherChannel misconfiguration guard is enabled
Extended system ID is enabled
Portfast is disabled by default
PortFast BPDU Guard is disabled by default
Portfast BPDU Filter is disabled by default
Loopguard is disabled by default
UplinkFast is disabled
BackboneFast is disabled
Pathcost method used is long

Name              Blocking Listening Learning Forwarding STP Active
-----
MST00              1          0          0          2          3
MST01              1          0          0          2          3
-----
2 msts             2          0          0          4          6
Switch#

```

About MST-to-PVST+ Interoperability (PVST+ Simulation)

The PVST+ simulation feature enables seamless interoperability between MST and Rapid PVST+. You can enable or disable this per port, or globally. PVST+ simulation is enabled by default.

However, you may want to control the connection between MST and Rapid PVST+ to protect against accidentally connecting an MST-enabled port to a Rapid PVST+-enabled port. Because Rapid PVST+ is the default STP mode, you may encounter many Rapid PVST+-enabled connections.

Disabling this feature causes the switch to stop the MST region from interacting with PVST+ regions. The MST-enabled port moves to a PVST peer inconsistent (blocking) state once it detects it is connected to a Rapid PVST+-enabled port. This port remains in the inconsistent state until the port stops receiving Shared Spanning Tree Protocol (SSTP) BPDUs, and then the port resumes the normal STP transition process.

You can for instance, disable PVST+ simulation, to prevent an incorrectly configured switch from connecting to a network where the STP mode is not MSTP (the default mode is PVST+).

Observe these guidelines when you configure MST switches (in the same region) to interact with PVST+ switches:

- Configure the root for all VLANs inside the MST region as shown in this example:

```

Switch# show spanning-tree mst interface gigabitethernet 1/1

GigabitEthernet1/1 of MST00 is root forwarding
Edge port: no (trunk) port guard : none (default)
Link type: point-to-point (auto) bpdu filter: disable (default)
Boundary : boundary (PVST) bpdu guard : disable (default)
Bpdus sent 10, received 310

```

Instance	Role	Sts	Cost	Prio.	Nbr Vlans mapped
0	Root	FWD	20000	128.1	1-2, 4-2999, 4000-4094
3	Boun	FWD	20000	128.1	3, 3000-3999

The ports that belong to the MST switch at the boundary simulate PVST+ and send PVST+ BPDUs for all the VLANs.

If you enable loop guard on the PVST+ switches, the ports might change to a loop-inconsistent state when the MST switches change their configuration. To correct the loop-inconsistent state, you must disable and re-enable loop guard on that PVST+ switch.

- Do not locate the root for some or all of the VLANs inside the PVST+ side of the MST switch because when the MST switch at the boundary receives PVST+ BPDUs for all or some of the VLANs on its designated ports, root guard sets the port to the blocking state.
- When you connect a PVST+ switch to two different MST regions, the topology change from the PVST+ switch does not pass beyond the first MST region. In such a case, the topology changes are propagated only in the instance to which the VLAN is mapped. The topology change stays local to the first MST region, and the Cisco Access Manager (CAM) entries in the other region are not flushed. To make the topology change visible throughout other MST regions, you can map that VLAN to IST or connect the PVST+ switch to the two regions through access links.
- When you disable the PVST+ simulation, note that the PVST+ peer inconsistency can also occur while the port is already in other states of inconsistency. For example, the root bridge for all STP instances must all be in either the MST region or the Rapid PVST+ side. If the root bridge for all STP instances are not on one side or the other, the software moves the port into a PVST+ simulation-inconsistent state.



Note We recommend that you put the root bridge for all STP instances in the MST region.

Configuring PVST+ Simulation

PVST+ simulation is enabled by default. This means that all ports automatically interoperate with a connected device that is running in Rapid PVST+ mode. If you disabled the feature and want to re-configure it, refer to the following tasks.

To enable PVST+ simulation globally, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters the global configuration mode.
Step 2	Switch(config)# [no] spanning-tree mst simulate pvst global	Enables PVST+ simulation globally. To prevent the switch from automatically interoperating with a connecting switch that is running Rapid PVST+, enter the no version of the command.

This example shows how to prevent the switch from automatically interoperating with a connecting switch that is running Rapid PVST+:

```
Switch# configure terminal
Switch(config)# no spanning-tree mst simulate pvst global
```

To enable PVST+ simulation on a port, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters the global configuration mode.
Step 2	Switch(config)# interface {type slot/port}	Selects a port to configure.
Step 3	Switch(config-if)# spanning-tree mst simulate pvst	Enables PVST+ simulation on the specified interface. To prevent a specified interface from automatically interoperating with a connecting switch that is not running MST, enter the spanning-tree mst simulate pvst disable command.

This example shows how to prevent a port from automatically interoperating with a connecting device that is running Rapid PVST+:

```
Switch(config)# interface gi3/13
Switch(config-if)# spanning-tree mst simulate pvst disable
```

The following sample output shows the system message you receive when a SSTP BPDU is received on a port and PVST+ simulation is disabled:

```
Message
SPANTREE_PVST_PEER_BLOCK: PVST BPDU detected on port %s [port number].
```

```
Severity
Critical
```

```
Explanation
A PVST+ peer was detected on the specified interface on the switch. PVST+ simulation
feature is disabled, as a result of which the interface was moved to the spanning tree
Blocking state.
```

```
Action
Identify the PVST+ switch from the network which might be configured incorrectly.
```

The following sample output shows the system message you receive when peer inconsistency on the interface is cleared:

```
Message
SPANTREE_PVST_PEER_UNBLOCK: Unblocking port %s [port number].
```

```
Severity
Critical
```

```
Explanation
The interface specified in the error message has been restored to normal spanning tree
state.
```

```
Action
None.
```

This example shows the spanning tree status when port Gi3/14 has been configured to disable PVST+ simulation and is currently in the peer type inconsistent state:

```
Switch# show spanning-tree
VLAN0010
  Spanning tree enabled protocol mstp
    Root ID    Priority    32778
              Address     0002.172c.f400
              This bridge is the root
              Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

    Bridge ID  Priority    32778 (priority 32768 sys-id-ext 10)
              Address     0002.172c.f400
              Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time 300

Interface                Role Sts Cost          Prio.Nbr Type
-----
Gi3/14                   Desg BKN*4      128.270  P2p *PVST_Peer_Inc
```

This example shows the spanning tree summary when PVST+ simulation is enabled in the MSTP mode:

```
Switch# show spanning-tree summary
Switch is in mst mode (IEEE Standard)
Root bridge for: MST0
EtherChannel misconfig guard is enabled
Extended system ID          is enabled
Portfast Default            is disabled
PortFast BPDU Guard Default is disabled
Portfast BPDU Filter Default is disabled
Loopguard Default           is disabled
UplinkFast                  is disabled
BackboneFast                 is disabled
Pathcost method used         is long
PVST Simulation Default      is enabled
```

Name	Blocking	Listening	Learning	Forwarding	STP Active
MST0	2	0	0	0	2
1 mst	2	0	0	0	2

This example shows the spanning tree summary when PVST+ simulation is disabled in any STP mode:

```
Switch# show spanning-tree summary
Switch is in mst mode (IEEE Standard)
Root bridge for: MST0
EtherChannel misconfig guard is enabled
Extended system ID          is enabled
Portfast Default            is disabled
PortFast BPDU Guard Default is disabled
Portfast BPDU Filter Default is disabled
Loopguard Default           is disabled
UplinkFast                  is disabled
BackboneFast                 is disabled
Pathcost method used         is long
PVST Simulation Default      is disabled
```

Name	Blocking	Listening	Learning	Forwarding	STP Active
MST0	2	0	0	0	2
1 mst	2	0	0	0	2

This example shows the spanning tree summary when the switch is not in MSTP mode, that is, the switch is in PVST or Rapid-PVST mode. The output string displays the current STP mode:

```
Switch# show spanning-tree summary
Switch is in rapid-pvst mode
Root bridge for: VLAN0001, VLAN2001-VLAN2002
EtherChannel misconfig guard is enabled
Extended system ID is enabled
Portfast Default is disabled
PortFast BPDU Guard Default is disabled
Portfast BPDU Filter Default is disabled
Loopguard Default is disabled
UplinkFast is disabled
BackboneFast is disabled
Pathcost method used is short
PVST Simulation Default is enabled but inactive in rapid-pvst mode
```

Name	Blocking	Listening	Learning	Forwarding	STP Active
VLAN0001	2	0	0	0	2
VLAN2001	2	0	0	0	2
VLAN2002	2	0	0	0	2
3 vlans	6	0	0	0	6

This example shows the interface details when PVST+ simulation is globally enabled, or the default configuration:

```
Switch# show spanning-tree interface gi3/13 detail
Port 269 (GigabitEthernet3/13) of VLAN0002 is forwarding
  Port path cost 4, Port priority 128, Port Identifier 128.297.
  Designated root has priority 32769, address 0013.5f20.01c0
  Designated bridge has priority 32769, address 0013.5f20.01c0
  Designated port id is 128.297, designated path cost 0
  Timers: message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state: 1
  Link type is point-to-point by default
  PVST Simulation is enabled by default
  BPDU: sent 132, received 1
```

This example shows the interface details when PVST+ simulation is globally disabled:

```
Switch# show spanning-tree interface gi3/13 detail
Port 269 (GigabitEthernet3/13) of VLAN0002 is forwarding
  Port path cost 4, Port priority 128, Port Identifier 128.297.
  Designated root has priority 32769, address 0013.5f20.01c0
  Designated bridge has priority 32769, address 0013.5f20.01c0
  Designated port id is 128.297, designated path cost 0
  Timers: message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state: 1
  Link type is point-to-point by default
  PVST Simulation is disabled by default
  BPDU: sent 132, received 1
```

This example shows the interface details when PVST+ simulation is explicitly enabled on the port:

```
Switch# show spanning-tree interface gi3/13 detail
Port 269 (GigabitEthernet3/13) of VLAN0002 is forwarding
  Port path cost 4, Port priority 128, Port Identifier 128.297.
  Designated root has priority 32769, address 0013.5f20.01c0
  Designated bridge has priority 32769, address 0013.5f20.01c0
  Designated port id is 128.297, designated path cost 0
  Timers: message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state: 1
```

```

Link type is point-to-point by default
PVST Simulation is enabled
BPDU: sent 132, received 1

```

This example shows the interface details when the PVST+ simulation feature is disabled and a PVST Peer inconsistency has been detected on the port:

```

Switch# show spanning-tree interface gi3/13 detail
Port 269 (GigabitEthernet3/13) of VLAN0002 is broken (PVST Peer Inconsistent)
  Port path cost 4, Port priority 128, Port Identifier 128.297.
  Designated root has priority 32769, address 0013.5f20.01c0
  Designated bridge has priority 32769, address 0013.5f20.01c0
  Designated port id is 128.297, designated path cost 0
  Timers: message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state: 1
  Link type is point-to-point by default
  PVST Simulation is disabled
  BPDU: sent 132, received 1

```

About Detecting Unidirectional Link Failure

The dispute mechanism that detects unidirectional link failures is included in the IEEE 802.1D-2004 RSTP and IEEE 802.1Q-2005 MSTP standard, and requires no user configuration.

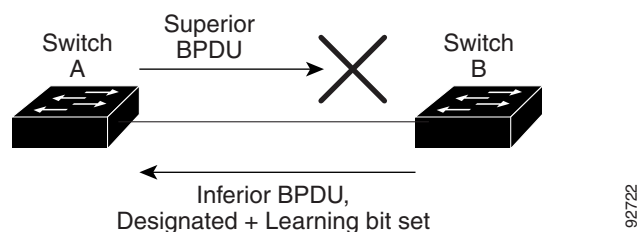
The switch checks the consistency of the port role and state in the BPDUs it receives, to detect unidirectional link failures that could cause bridging loops. When a designated port detects a conflict, it keeps its role, but reverts to a discarding (blocking) state because disrupting connectivity in case of inconsistency is preferable to opening a bridging loop.

For example, in [Figure 23-3](#), Switch A is the root bridge and Switch B is the designated port. BPDUs from Switch A are lost on the link leading to switch B.

Since Rapid PVST+ (802.1w) and MST BPDUs include the role and state of the sending port, Switch A detects (from the inferior BPDU), that switch B does not react to the superior BPDUs it sends, because switch B has the role of a designated port and not the root bridge.

As a result, switch A blocks (or keeps blocking) its port, thus preventing the bridging loop.

Figure 23-3 Detecting Unidirectional Link Failure



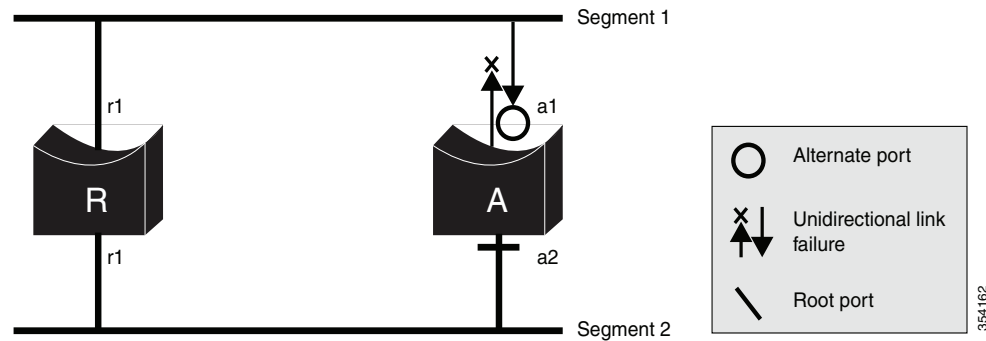
92722

Note these guidelines and limitations relating to the dispute mechanism:

- It works only on switches running RSTP or MST, because the dispute mechanism requires reading the role and state of the port initiating the BPDUs.

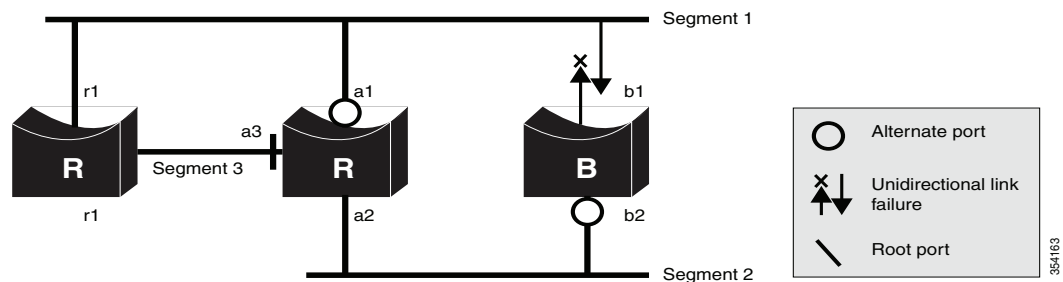
- It may result in loss of connectivity. For example, in Figure 23-4, Bridge A cannot transmit on the port it elected as a root port. As a result of this situation, there is loss of connectivity (r1 and r2 are designated, a1 is root and a2 is alternate). There is only a one way connectivity between A and R.

Figure 23-4 Loss of Connectivity



- It may cause permanent bridging loops on shared segments. For example, in Figure 23-5, suppose that bridge R has the best priority, and that port b1 cannot receive any traffic from the shared segment 1 and sends inferior designated information on segment 1. Both r1 and a1 can detect this inconsistency. However, with the current dispute mechanism, only r1 will revert to discarding while the root port a1 opens a permanent loop. However, this problem does not occur in Layer 2 switched networks that are connected by point-to-point links.

Figure 23-5 Bridging Loops on Shared Segments



This example shows the spanning tree status when port Gi3/14 has been configured to disable PVST+ simulation and the port is currently in the peer type inconsistent state:

```
Switch# show spanning-tree
VLAN0010
  Spanning tree enabled protocol rstp
    Root ID    Priority    32778
              Address     0002.172c.f400
              This bridge is the root
              Hello Time  2 sec    Max Age 20 sec    Forward Delay 15 sec
    Bridge ID  Priority    32778 (priority 32768 sys-id-ext 10)
              Address     0002.172c.f400
              Hello Time  2 sec    Max Age 20 sec    Forward Delay 15 sec
              Aging Time 300
Interface      Role Sts Cost      Prio.Nbr Type
-----
Gi3/14        Desg BKN 4        128.270   P2p Dispute
```

This example shows the interface details when a dispute condition is detected:

```
Switch# show spanning-tree interface gi3/13 detail
Port 269 (GigabitEthernet3/13) of VLAN0002 is designated blocking (dispute)
  Port path cost 4, Port priority 128, Port Identifier 128.297.
  Designated root has priority 32769, address 0013.5f20.01c0
  Designated bridge has priority 32769, address 0013.5f20.01c0
  Designated port id is 128.297, designated path cost 0
  Timers: message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state: 1
  Link type is point-to-point by default
  BPDU: sent 132, received 1
```




Configuring Flex Links and MAC Address-Table Move Update

Flex Links provide a fast and simplified Layer 2 Link redundancy mechanism. This chapter describes how to configure Flex Links on the Catalyst 4500 series switch. It also describes how to configure the MAC address-table move update (MMU) feature, also referred to as the Flex Links bidirectional fast convergence feature.

The chapter consists of these sections:

- [About Flex Links, page 24-1](#)
- [Configuring Flex Links, page 24-5](#)
- [Monitoring Flex Links and the MAC Address-Table Move Update, page 24-12](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About Flex Links

This section describes this information:

- [Flex Links, page 24-1](#)
- [VLAN Flex Links Load Balancing and Support, page 24-2](#)
- [Flex Links Failover Actions, page 24-3](#)

Flex Links

Flex Links are a pair of Layer 2 interfaces (switch ports or port channels) where one interface is configured to act as a backup to the other. Users can disable STP and still retain basic link redundancy. Flex Links are typically configured in service provider or enterprise networks where customers do not want to run STP on some interfaces.

**Note**

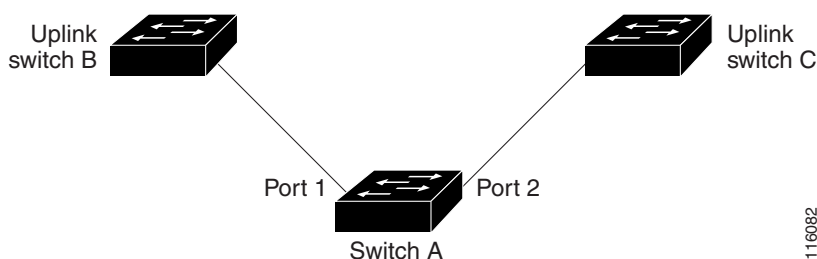
The Catalyst 4500 series switch supports a maximum of 16 Flex Links.

You configure Flex Links on one Layer 2 interface (the active link) by assigning another Layer 2 interface as the Flex Link or backup link. When one of the links is up and forwarding traffic, the other link is in standby mode, ready to begin forwarding traffic if the other link fails. At any given time, only one of the interfaces is in the forwarding state and forwarding traffic. If the primary link fails, the standby link starts forwarding traffic. When the active link reactivates, it enters standby mode and does not forward traffic. STP is disabled on Flex Links interfaces.

In [Figure 24-1](#), ports 1 and 2 on switch A are connected to uplink switches B and C. Because they are configured as Flex Links, only one of the interfaces is forwarding traffic; the other is in standby mode. If port 1 is the active link, it begins forwarding traffic between port 1 and switch B; the link between port 2 (the backup link) and switch C is not forwarding traffic. If port 1 shuts down, port 2 activates and starts forwarding traffic to switch C. When port 1 reactivates, it enters standby mode and does not forward traffic; port 2 continues forwarding traffic.

You can also choose to configure a preemption mechanism, specifying the preferred port for forwarding traffic. In [Figure 24-1](#), for example, you can configure the Flex Links pair with preemption mode so that after port 1 reactivates in the scenario, and it has greater bandwidth than port 2, port 1 begins forwarding after a duration equal to the preemption delay; and port 2 becomes the standby. You do this by entering the interface configuration **switchport backup interface preemption mode bandwidth** and **switchport backup interface preemption delay** commands.

Figure 24-1 Flex Links Configuration Example

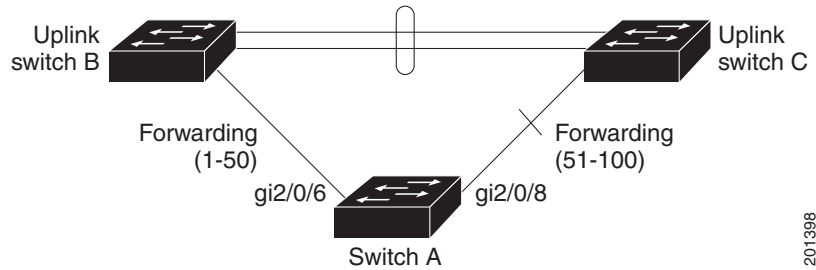


If a primary (forwarding) link shuts down, a trap notifies the network management stations. If the standby link shuts down, a trap notifies the users.

Flex Links are supported only on Layer 2 ports and port channels. Flex Links are compatible with promiscuous trunks. It is not supported on isolated PVLAN trunks.

VLAN Flex Links Load Balancing and Support

VLAN Flex Links load balancing allows you to configure a Flex Links pair so that both ports simultaneously forward the traffic for mutually exclusive VLANs. For example, if Flex Links ports are configured for 1 to 100 VLANs, the traffic of the first 50 VLANs can be forwarded on one port and the rest on the other port. If one of the ports fail, the other active port forwards all the traffic. When the failed port reactivates, it resumes forwarding traffic in the preferred VLANs. In addition to providing the redundancy, this Flex Links pair can be used for load balancing. Also, Flex Links VLAN load balancing also does not impose any restrictions on uplink switches ([Figure 24-2](#)).

Figure 24-2 VLAN Flex Links Load Balancing Configuration Example**Note**

A static MAC address must point to a Flex Links interface that is forwarding for given VLAN. For example, if a backup interface is forwarding VLAN X, then a static MAC address in VLAN X must point to the backup interface. Misconfiguration might cause unexpected results.

Flex Links Failover Actions

When a Flex Links primary fails, the following important actions are taken:

- Detects link failure.
- Moves static unicast MAC addresses that are configured on the primary link to the standby link.
- Moves dynamic unicast MAC addresses that are learned on the primary link to the standby link.
- Moves the standby link to a forwarding state.
- Transmits MAC address-table move updates over a new active link, if you enter the **mac move update transmit** command.
- Transmits dummy multicast packets over a new active interface.

**Note**

Local administrative shut down or a link that starts forwarding again due to preemption is not considered a link failure. In those cases, flush the dynamic hosts and not move them.

Static MAC addresses configured on a Flex Links member interface are moved over to the backup, if it fails. Static MAC addresses configured on a Flex Links member interface are restored when it starts forwarding again.

**Note**

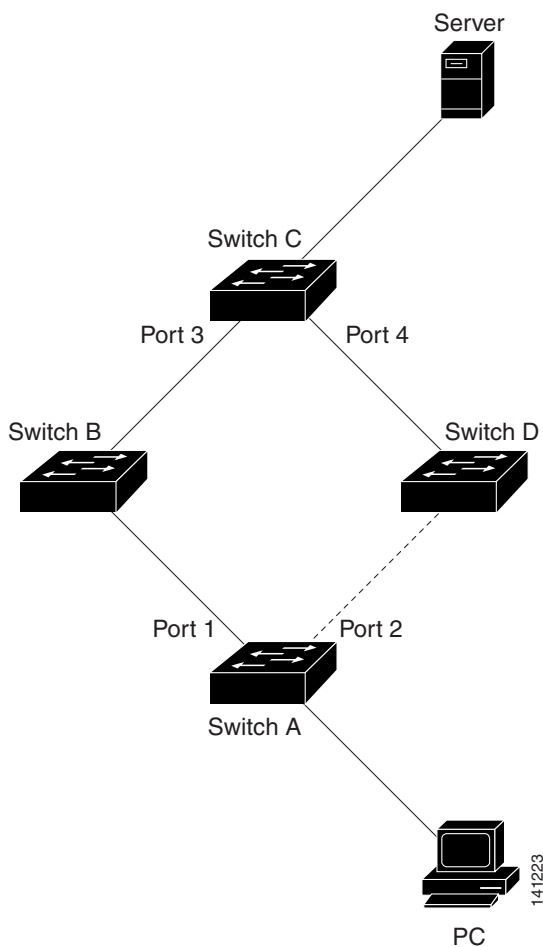
The **show mac address-table** command always shows static MAC addresses as associated with the interface on which it was configured even if it may have been moved to the standby link because of a Flex Links failover.

MAC Address-Table Move Update

In [Figure 24-3](#), ports 1 and 2 on switch A are connected to uplink switches B and D through a Flex Links pair. Port 1 is forwarding traffic, and port 2 is in the blocking state. Traffic from the PC to the server is forwarded from port 1 to port 3. The MAC address of the PC was learned on port 3 of switch C. Traffic from the server to the PC is forwarded from port 3 to port 1.

If port 1 shuts down, port 2 starts forwarding traffic. If there is no traffic from PC to the server after failover to port 2, switch C does not learn MAC address of the PC on port 4. As a result, switch C keeps forwarding traffic from the server to the PC out of port 3. There is traffic loss from server to PC because port 1 is down. This problem is alleviated by sending out a dummy multicast packet with source MAC address of the PC over port 2. Switch C learns the PC MAC address on port 4 and start forwarding traffic from server to the PC out of port 4. One dummy multicast packet is sent out for every MAC address, which is the default Flex Links behavior. The MAC address-table move update (MMU) feature may be enabled to further expedite downstream convergence. MMUs are special packets that carry multiple MAC addresses. Switch A is configured to transmit these packets and switches B, C, and D are configured to receive such packets. If MMU transmit is enabled on Switch A, MAC move updates are transmitted before dummy multicast packets over port 2. Switch D processes and floods MMUs over to Switch C. Switch C processes these packets, and moves the MAC addresses contained within the packets from port 3 to port 4. Because one packet carries multiple MAC addresses, downstream convergence is faster.

Figure 24-3 MAC Address-Table Move Update Example



Configuring Flex Links

These sections contain this configuration information:

- [Default Configuration, page 24-5](#)
- [Configuration Guidelines, page 24-5](#)
- [Configuring Flex Links, page 24-6](#)
- [Configuring VLAN Load Balancing on Flex Links, page 24-8](#)

Default Configuration

The following is the default Flex Links configuration:

- Flex Links are not configured on any interface.
- Preemption mode is off.
- If preemption is enabled, preemption delay is 35 seconds.

Configuration Guidelines

Follow these guidelines to configure Flex Links and associated features:

- You can configure only one Flex Link backup link for any active link, and it must be a different interface from the active interface.
- An interface can belong to only one Flex Links pair. An interface can be a backup link for only one active link, but an active link cannot belong to another Flex Links pair.
- Neither of the links can be a port that belongs to an EtherChannel. However, you can configure two port channels (EtherChannel logical interfaces) as Flex Links. Moreover, you can configure a port channel and a physical interface as Flex Links, with either the port channel or the physical interface as the active link.
- The types (Fast Ethernet, Gigabit Ethernet, or port channel) of the backup link and the active link can differ. However, you should configure both Flex Links with similar characteristics so that no loops exist or changes occur in behavior if the standby link begins to forward traffic.
- STP is disabled on Flex Links ports. A Flex Links port does not participate in STP, even if the VLANs present on the port are configured for STP. When STP is not enabled, ensure that no loops exist in the configured topology.
- Configure any static MAC addresses on a Flex Links member interface after enabling Flex Links.

Follow these guidelines to configure VLAN load balancing on the Flex Links feature. For Flex Links VLAN load balancing, you must choose the preferred VLANs on the backup interface.

Set **switchport backup interface interface-id preempt mode** to forced. A default value of 35 seconds is used as the delay timeout. You cannot configure **switchport backup interface interface-id preempt mode** on the interface. So, mode **bandwidth** and **off** cannot be configured.

Adjust the delay time with the **switchport backup interface interface-id preempt delay delay-time** command.

Configuring Flex Links

To configure a pair of Flex Links, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(conf)# interface <i>interface-id</i>	Specifies the interface, and enters interface configuration mode. The interface might be a physical Layer 2 interface or a port channel (logical interface). The port channel range is 1 to 64.
Step 3	Switch(conf-if)# switchport backup interface <i>interface-id</i>	Configures a physical Layer 2 interface (or port channel) as part of a Flex Links pair with the interface. When one link is forwarding traffic, the other interface is in standby mode.
Step 4	Switch(conf-if)# end	Returns to privileged EXEC mode.
Step 5	Switch# show interface [<i>interface-id</i>] switchport backup	Verifies the configuration.
Step 6	Switch# copy running-config startup config	(Optional) Saves your entries in the switch startup configuration file.

To disable a Flex Links backup interface, enter the **no switchport backup interface** *interface-id* interface configuration command.

This example shows how to configure an interface with a backup interface and to verify the configuration:

```
Switch# configure terminal
Switch(conf)# interface fastethernet1/1
Switch(conf-if)# switchport backup interface fastethernet1/2
Switch(conf-if)# end
Switch# show interface switchport backup
Switch Backup Interface Pairs:

Active Interface      Backup Interface      State
-----
FastEthernet1/1      FastEthernet1/2      Active Up/Backup Standby
FastEthernet1/3      FastEthernet1/4      Active Up/Backup Standby
Port-channel1        GigabitEthernet1/1    Active Up/Backup Standby
```

To configure a preemption scheme for a pair of Flex Links, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(conf)# interface <i>interface-id</i>	Specifies the interface, and enters interface configuration mode. The interface might be a physical Layer 2 interface or a port channel (logical interface). The port channel range is 1 to 64.
Step 3	Switch(conf-if)# switchport backup interface <i>interface-id</i>	Configures a physical Layer 2 interface (or port channel) as part of a Flex Links pair with the interface. When one link is forwarding traffic, the other interface is in standby mode.

	Command	Purpose
Step 4	Switch(conf-if)# switchport backup interface <i>interface-id</i> preemption mode [forced bandwidth off]	Configures a preemption mechanism and delay for a Flex Links interface pair. You can configure the preemption mode as: <ul style="list-style-type: none"> • forced—The active interface always preempts the backup. • bandwidth—The interface with higher bandwidth always acts as the active interface. • off—No preemption occurs from active to backup.
Step 5	Switch(conf-if)# switchport backup interface <i>interface-id</i> preemption delay <i>delay-time</i>	Configures the delay time until a port preempts another port. Note Setting a delay time requires forced or bandwidth mode.
Step 6	Switch(conf)# end	Returns to privileged EXEC mode.
Step 7	Switch# show interface [<i>interface-id</i>] switchport backup	Verifies the configuration.
Step 8	Switch# copy running-config startup config	(Optional) Saves your entries in the switch startup configuration file.

To remove a preemption scheme, enter the **no switchport backup interface** *interface-id* **preemption mode** interface configuration command. To reset the delay time to the default, enter the **no switchport backup interface** *interface-id* **preemption delay** interface configuration command.

This example shows how to configure preemption mode as bandwidth for a backup interface pair and to verify the configuration:

```
Switch# configure terminal
Switch(conf)# interface gigabitethernet1/0/1
Switch(conf-if)# switchport backup interface gigabitethernet1/2
Switch(conf-if)# switchport backup interface gigabitethernet1/2 preemption mode forced
Switch(conf-if)# switchport backup interface gigabitethernet1/2 preemption delay 50
Switch(conf-if)# end
Switch# show interface switchport backup detail
Active Interface      Backup Interface      State
-----
GigabitEthernet1/21  GigabitEthernet1/2    Active Down/Backup Down
Interface Pair       : Gi1/21, Gi1/2
Preemption Mode      : forced
Preemption Delay     : 50 seconds
Bandwidth : 10000 Kbit (Gi1/1), 10000 Kbit (Gi1/2)
Mac Address Move Update Vlan : auto
```

<output truncated>

Configuring VLAN Load Balancing on Flex Links

To configure VLAN load balancing on Flex Links, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Specifies the interface, and enters interface configuration mode. The interface might be a physical Layer 2 interface or a port channel (logical interface). The port channel range is 1 to 48.
Step 3	Switch(config-if)# switchport backup interface <i>interface-id</i> prefer vlan <i>vlan-range</i>	Configures a physical Layer 2 interface (or port channel) as part of a Flex Links pair with the interface, and specifies the VLANs carried on the interface. The VLAN ID range is 1 to 4094.
Step 4	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 5	Switch# show interfaces [<i>interface-id</i>] switchport backup	Verifies the configuration.
Step 6	Switch# copy running-config startup config	(Optional) Saves your entries in the switch startup configuration file.

To disable the VLAN load balancing feature, enter the **no switchport backup interface prefer vlan** interface configuration command.

In this example, VLANs 1 to 50, 60, and 100 to 120 are configured on the switch:

```
Switch(config)# interface fastethernet 1/6
Switch(config-if)# switchport backup interface fastethernet 1/0/8 prefer vlan 60,100-120
```

When both interfaces are up, Fast Ethernet port 1/0/8 forwards traffic for VLANs 60 and 100 to 120 and Fast Ethernet port 1/0/6 forwards traffic for VLANs 1 to 50.

```
Switch# show interfaces switchport backup
Switch Backup Interface Pairs:
```

Active Interface	Backup Interface	State
FastEthernet1/6	FastEthernet1/8	Active Up/Backup Standby

```
Vlans Preferred on Active Interface: 1-50
Vlans Preferred on Backup Interface: 60, 100-120
```

When a Flex Links interface shuts down, VLANs preferred on this interface are moved to the peer interface of the Flex Links pair. In this example, if interface 1/6 shuts down, interface 1/8 carries all VLANs of the Flex Links pair.

```
Switch# show interfaces switchport backup
Switch Backup Interface Pairs:
```

Active Interface	Backup Interface	State
FastEthernet1/6	FastEthernet1/8	Active Down/Backup VLB all

```
Vlans Preferred on Active Interface: 1-50
Vlans Preferred on Backup Interface: 60, 100-120
```


When a Flex Links interface becomes active, VLANs preferred on this interface are blocked on the peer interface and moved to the forwarding state on the interface that has just come up. In this example, if interface Fast Ethernet port 1/6 becomes active, VLANs preferred on this interface are blocked on the peer interface Fast Ethernet port 1/8 and forwarded on Fast Ethernet port 1/6.

```
Switch# show interfaces switchport backup
Switch Backup Interface Pairs:
```

Active Interface	Backup Interface	State
FastEthernet1/6	FastEthernet1/8	Active VLB cfg/Backup VLB cfg

```
Vlans Preferred on Active Interface: 1-50
Vlans Preferred on Backup Interface: 60, 100-120
```

```
Switch# show interfaces switchport backup detail
Switch Backup Interface Pairs:
```

Active Interface	Backup Interface	State
FastEthernet1/6	FastEthernet1/8	Active VLB cfg/Backup VLB cfg

```
Vlans Preferred on Active Interface: 1-50
Vlans Preferred on Backup Interface: 60, 100-120
Preemption Mode : off
Bandwidth : 10000 Kbit (Fa1/6), 100000 Kbit (Fa1/8)
Mac Address Move Update Vlan : auto
```

Configuring MAC Address-Table Move Update

These sections contain this configuration information:

- [Default Configuration, page 24-5](#)
- [Configuration Guidelines, page 24-5](#)
- [Configuring MAC Address-Table Move Update, page 24-9](#)

Default Configuration

By default, the MAC address-table move update feature is disabled.

Configuration Guidelines

Follow these guidelines to configure the MAC address-table move update feature:

- Enable **mac address-table move transmit** on the switch with Flex Links configured to send MAC address-table move updates.
- Enable **mac address-table move receive** on all upstream switches to process MAC address-table move updates.

Configuring the MAC Address-Table Move Update Feature

This section describes the following tasks:

- [Configuring a Switch to Send MAC Address-Table Move Updates, page 24-10](#)
- [Configuring a Switch to Receive MAC Address-Table Move Updates, page 24-11](#)

Configuring a Switch to Send MAC Address-Table Move Updates

To configure an access switch to send MAC address-table move updates, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(conf)# interface <i>interface-id</i>	Specifies the interface, and enters interface configuration mode. The interface might be a physical Layer 2 interface or a port channel (logical interface). The port channel range is 1 to 64.
Step 3	Switch(conf-if)# switchport backup interface <i>interface-id</i> or Switch(conf-if)# switchport backup interface <i>interface-id</i> mmu primary vlan <i>vlan-id</i>	Configures a physical Layer 2 interface (or port channel), as part of a Flex Links pair with the interface. The MAC address-table move update VLAN is the lowest VLAN ID on the interface. Configures a physical Layer 2 interface (or port channel) and specifies the VLAN ID on the interface, which is used for sending the MAC address-table move update. When one link is forwarding traffic, the other interface is in standby mode.
Step 4	Switch(conf-if)# end	Returns to global configuration mode.
Step 5	Switch(conf)# mac address-table move update transmit	Enables the access switch to send MAC address-table move updates to other switches in the network if the primary link shuts down and the switch starts forwarding traffic through the standby link.
Step 6	Switch(conf)# end	Returns to privileged EXEC mode.
Step 7	Switch# show mac address-table move update	Verifies the configuration.
Step 8	Switch# copy running-config startup config	(Optional) Saves your entries in the switch startup configuration file.

To disable the MAC address-table move update feature on the access switch, enter the **no mac address-table move update transmit** interface configuration command. To display the MAC address-table move update information, enter the **show mac address-table move update** command.

This example shows how to configure an access switch to send MAC address-table move update messages and to verify the configuration:

```
Switch# configure terminal
Switch(conf)# interface fastethernet1/1
Switch(conf-if)# switchport backup interface fastethernet1/0/2 mmu primary vlan 2
Switch(conf-if)# end
Switch(conf)# mac address-table move update transmit
Switch(conf)# end
Switch# show mac-address-table move update
```

```

Switch-ID : 01d0.2bfc.3180
Dst mac-address : 0180.c200.0010
Vlans/Macs supported : 4096/55000
Default/Current settings: Rcv Off/Off, Xmt Off/On
Max packets per min : Rcv 100, Xmt 120
Rcv packet count : 0
Rcv conforming packet count : 0
Rcv invalid packet count : 0
Rcv packet count this min : 0
Rcv threshold exceed count : 0
Rcv last sequence# this min : 0
Rcv last interface : None
Rcv last src-mac-address : 0000.0000.0000
Rcv last switch-ID : 0000.0000.0000
Xmt packet count : 0
Xmt packet count this min : 0
Xmt threshold exceed count : 0
Xmt pak buf unavail cnt : 0
Xmt last interface : fa1/2

```

Configuring a Switch to Receive MAC Address-Table Move Updates

To configure a switch to receive and process MAC address-table move update messages, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(conf)# mac address-table move update receive	Enables the switch to receive and process the MAC address-table move updates.
Step 3	Switch(conf)# end	Returns to privileged EXEC mode.
Step 4	Switch# show mac address-table move update	Verifies the configuration.
Step 5	Switch# copy running-config startup config	(Optional) Saves your entries in the switch startup configuration file.

To disable the MAC address-table move update feature on the access switch, enter the **no mac address-table move update receive** configuration command. To display the MAC address-table move update information, enter the **show mac address-table move update** command.

This example shows how to configure a switch to receive and process MAC address-table move update messages:

```

Switch# configure terminal
Switch(conf)# mac address-table move update receive
Switch(conf)# end

```

Monitoring Flex Links and the MAC Address-Table Move Update

Table 24-1 shows the commands for monitoring the Flex Links configuration and the MAC address-table move update information.

Table 24-1 Flex Links and MAC Address-Table Move Update Monitoring Commands

Command	Purpose
Switch# show interface [<i>interface-id</i>] switchport backup	Displays the Flex Link backup interface configured for an interface or all the configured Flex Links and the state of each active and backup interface (up or standby mode).
Switch# show mac address-table move update	Displays the MAC address-table move update information on the switch.



Configuring Resilient Ethernet Protocol

This chapter describes how to use Resilient Ethernet Protocol (REP) on the Catalyst 4500 Series switch. REP is a Cisco proprietary protocol that provides an alternative to Spanning Tree Protocol (STP) to control network loops, handle link failures, and improve convergence time. REP controls a group of ports connected in a segment, ensures that the segment does not create any bridging loops, and responds to link failures within the segment. REP provides a basis for constructing more complex networks and supports VLAN load balancing.

This chapter includes these sections:

- [About REP, page 25-1](#)
- [Configuring REP, page 25-7](#)
- [Monitoring REP, page 25-14](#)



Note

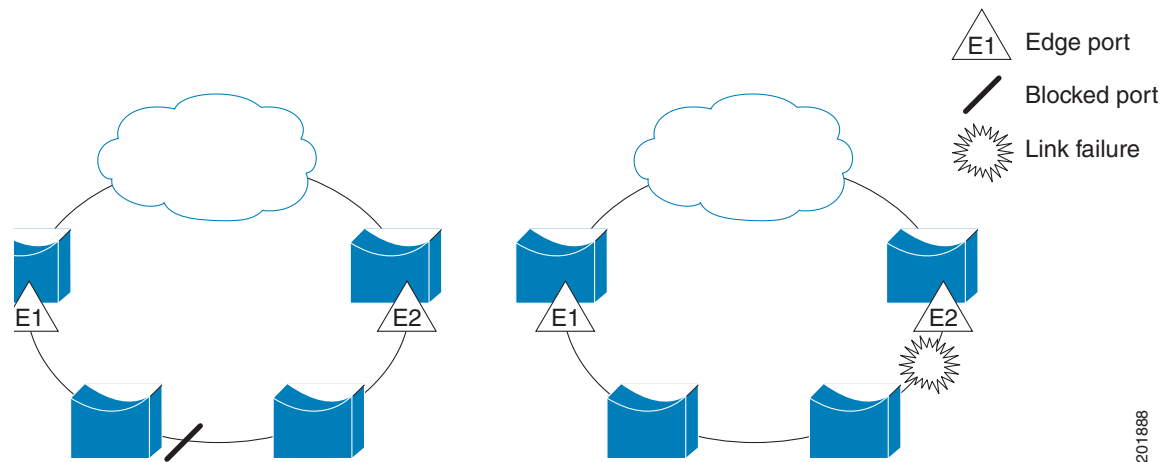
For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About REP

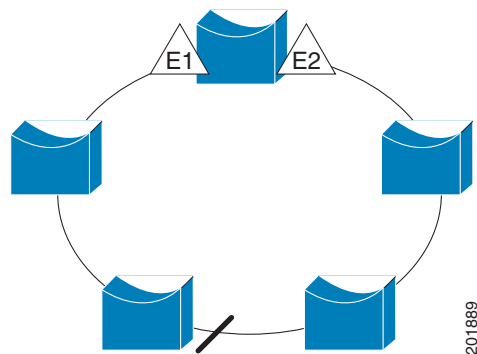
One REP segment is a chain of ports connected to each other and configured with a segment ID. Each segment consists of standard (non-edge) segment ports and two user-configured edge ports. A switch can have no more than two ports that belong to the same segment, and each segment port can have only one external neighbor. A segment can go through a shared medium, but on any link only two ports can belong to the same segment. REP is supported only on Layer 2 trunk and PVLAN promiscuous trunk interfaces.

[Figure 25-1](#) shows an example of a segment consisting of six ports spread across four switches. Ports E1 and E2 are configured as edge ports. When all ports are operational (as in the segment on the left), a single port is blocked, shown by the diagonal line. When there is a failure in the network, as shown in the diagram on the right, the blocked port returns to the forwarding state to minimize network disruption.

Figure 25-1 REP Open Segments

The segment shown in Figure 25-1 is an open segment; there is no connectivity between the two edge ports. The REP segment cannot cause a bridging loop and it is safe to connect the segment edges to any network. All hosts connected to switches inside the segment have two possible connections to the rest of the network through the edge ports, but only one connection is accessible at any time. If a failure causes a host to be unable to access its usual gateway, REP unblocks all ports to ensure that connectivity is available through the other gateway.

The segment shown in Figure 25-2, with both edge ports located on the same switch, is a ring segment. In this configuration, there is connectivity between the edge ports through the segment. With this configuration, you can create a redundant connection between any two switches in the segment.

Figure 25-2 REP Ring Segment

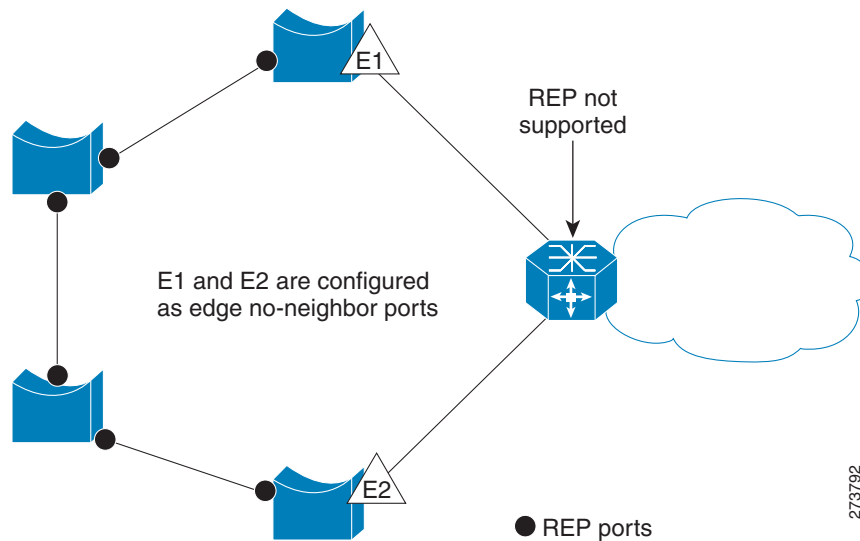
REP segments have these characteristics:

- If all ports in the segment are operational, one port (referred to as the *alternate* port) is in the blocked state for each VLAN. If VLAN load balancing is configured, two ports in the segment control the blocked state of VLANs.
- If one or more ports in a segment is not operational, causing a link failure, all ports forward traffic on all VLANs to ensure connectivity.
- In case of a link failure, the alternate ports are unblocked as quickly as possible. When the failed link comes back up, a logically blocked port per-VLAN is selected with minimal disruption to the network.

You can construct almost any type of network based on REP segments. REP also supports VLAN load balancing, controlled by the primary edge port but occurring at any port in the segment.

In access ring topologies, the neighboring switch might not support REP, as shown in [Figure 25-3](#). Starting with Cisco IOS Release 15.0(2)SG, you can configure the non-REP facing ports (E1 and E2) as edge no-neighbor ports. These ports inherit all properties of edge ports, and you can configure them the same as any edge port, including configuring them to send STP or REP topology change notices to the aggregation switch. In this case the STP topology change notice (TCN) that is sent is a multiple spanning-tree (MST) STP message.

Figure 25-3 Edge No-Neighbor Ports



REP has these limitations:

- You must configure each segment port; an incorrect configuration can cause forwarding loops in the networks.
- REP can manage only a single failed port within the segment; multiple port failures within the REP segment cause loss of network connectivity.
- You should configure REP only in networks with redundancy. Configuring REP in a network without redundancy causes loss of connectivity.

To use REP effectively, you should understand the following topics:

- [Link Integrity, page 25-4](#)
- [Fast Convergence, page 25-4](#)
- [VLAN Load Balancing, page 25-4](#)
- [Spanning Tree Interaction, page 25-6](#)
- [REP Ports, page 25-6](#)

Link Integrity

REP does not use an end-to-end polling mechanism between edge ports to verify link integrity. It implements local link failure detection. When enabled on an interface, the REP Link Status Layer (LSL) detects its REP-aware neighbor and establishes connectivity within the segment. All VLANs are blocked on an interface until it detects the neighbor. After the neighbor is identified, REP determines which neighbor port should become the alternate port and which ports should forward traffic.

Each port in a segment has a unique port ID. The port ID format is similar to that used by the spanning tree algorithm: a port number (unique on the bridge), associated to a MAC address (unique in the network). When a segment port is coming up, its LSL starts sending packets that include the segment ID and the port ID. The port is declared as operational after it performs a three-way handshake with a neighbor in the same segment. A segment port does not become operational if these situations occur:

- No neighbor has the same segment ID.
- More than one neighbor has the same segment ID.
- The neighbor does not acknowledge the local port as a peer.

Each port creates an adjacency with its immediate neighbor. Once the neighbor adjacencies are created, the ports negotiate to determine one blocked port for the segment, the alternate port. All other ports become unblocked. By default, REP packets are sent to a BPDU class MAC address. The packets can also be sent to the Cisco multicast address, which at present is used only to send blocked port advertisement (BPA) messages when there is a failure in the segment. The packets are dropped by devices not running REP.

Fast Convergence

Because REP runs on a physical link basis and not a per-VLAN basis, only one hello message is required for all VLANs, reducing the load on the protocol. We recommend that you create VLANs consistently on all switches in a given segment and configure the same allowed VLANs on the REP trunk and PVLAN promiscuous trunk ports. To avoid the delay introduced by relaying messages in software, REP also allows some packets to be flooded to a regular multicast address. These messages operate at the hardware flood layer (HFL) and are flooded to the whole network, not just the REP segment. Switches that do not belong to the segment treat them as data traffic. You can control flooding of these messages by configuring a dedicated administrative VLAN for the whole domain.

The estimated convergence recovery time is less than 200 milliseconds for the local segment.

VLAN Load Balancing

One edge port in the REP segment acts as the primary edge port; the other as the secondary edge port. It is the primary edge port that always participates in VLAN load balancing in the segment. REP VLAN balancing is achieved by blocking some VLANs at a configured alternate port and all other VLANs at the primary edge port. When you configure VLAN load balancing, you can specify the alternate port in one of three ways:

- By entering the port ID of the interface.
To identify the port ID of a port in the segment, enter the **show interface rep detail** interface configuration command for the port.
- By entering the neighbor offset number of a port in the segment, which identifies the downstream neighbor port of an edge port.

The neighbor offset number range is -256 to +256; a value of 0 is invalid. The primary edge port has an offset number of 1; positive numbers above 1 identify downstream neighbors of the primary edge port. Negative numbers indicate the secondary edge port (offset number -1) and its downstream neighbors.

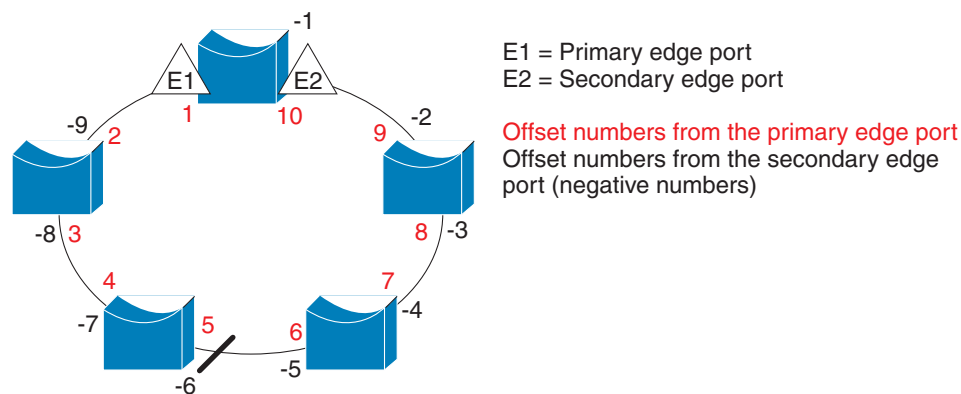
**Note**

You configure offset numbers on the primary edge port by identifying a port's downstream position from the primary (or secondary) edge port. You never enter an offset value of 1 because that is the offset number of the primary edge port itself.

Figure 25-4 shows neighbor offset numbers for a segment where E1 is the primary edge port and E2 is the secondary edge port. The red numbers inside the ring are numbers offset from the primary edge port; the black numbers outside of the ring show the offset numbers from the secondary edge port. Note that you can identify all ports (except the primary edge port) by either a positive offset number (downstream position from the primary edge port) or a negative offset number (downstream position from the secondary edge port). If E2 became the primary edge port, its offset number is then 1 and E1 is then -1.

- By entering the **preferred** keyword to select the port that you previously configured as the preferred alternate port with the **rep segment preferred** interface configuration command.

Figure 25-4 Neighbor Offset Numbers in a Segment



When the REP segment is complete, all VLANs are blocked. When you configure VLAN load balancing, it is triggered in one of two ways:

- You can manually trigger VLAN load balancing at any time by entering the **rep preempt segment segment-id** privileged EXEC command on the switch that has the primary edge port.
- You can configure a preempt delay time by entering the **rep preempt delay seconds** interface configuration command. After a link failure and recovery, VLAN load balancing begins after the configured preemption time period elapses. Note that the delay timer restarts if another port fails before the time has elapsed.

**Note**

When VLAN load balancing is configured, it does not start working until triggered by either manual intervention or a link failure and recovery.

When VLAN load balancing is triggered, the primary edge port then sends out a message to alert all interfaces in the segment about the preemption. When the message is received by the secondary edge port, it is reflected into the network to notify the alternate port to block the set of VLANs specified in the message and to notify the primary edge port to block the remaining VLANs.

You can also configure a particular port in the segment to block all VLANs. VLAN load balancing is initiated only by the primary edge port and is not possible if the segment is not terminated by an edge port on each end. The primary edge port determines the local VLAN load balancing configuration.

To reconfigure load balancing, you reconfigure the primary edge port. When you change the load balancing configuration, the primary edge port again waits for the **rep preempt segment** command or for the configured preempt delay period after a port failure and recovery before executing the new configuration. If you change an edge port to a regular segment port, the existing VLAN load balancing status does not change. Configuring a new edge port might cause a new topology configuration.

Spanning Tree Interaction

REP does not interact with STP, but can coexist with it. A port that belongs to a segment is removed from spanning tree control and STP BPDUs are not accepted or sent from segment ports. STP can not run on a segment.

To migrate from an STP ring configuration to REP segment configuration, begin by configuring a single port in the ring as part of the segment and continue by configuring contiguous ports to minimize the number of segments. Each segment always contains a blocked port, so multiple segments means multiple blocked ports and a potential loss of connectivity. When the segment has been configured in both directions up to the location of the edge ports, you then configure the edge ports.

REP Ports

Ports in REP segments take one of three roles or states: Failed, Open, or Alternate.

- A port configured as a regular segment port starts as a failed port.
- Once the neighbor adjacencies are determined, the port transitions to alternate port state, blocking all VLANs on the interface. Blocked port negotiations occur and when the segment settles, one blocked port remains in the alternate role and all other ports become open ports.
- When a failure occurs in a link, all ports move to the failed state. When the alternate port receives the failure notification, it changes to the open state, forwarding all VLANs.

A regular segment port converted to an edge port, or an edge port converted to a regular segment port, does not always result in a topology change. If you convert an edge port into a regular segment port, VLAN load balancing is not implemented unless it has been configured. For VLAN load balancing, you must configure two edge ports in the segment.

A segment port that is reconfigured as a spanning tree port restarts according the spanning tree configuration (by default, a designated blocking port). If PortFast is configured or if STP is disabled, the port goes into the forwarding state.

Configuring REP

A segment is a collection of ports connected one to the other in a chain and configured with a segment ID. To configure REP segments, you should configure the REP administrative VLAN and then add the ports to the segment using interface configuration mode. You should configure two edge ports in the segment, with one of them the primary edge port and the other by default the secondary edge port. A segment has only one primary edge port. If you configure two ports in a segment as the primary edge port, for example ports on different switches, the REP selects one of them to serve as the segment primary edge port. You can also optionally configure where to send segment topology change notices (STCNs) and VLAN load balancing.

This section includes this information:

- [Default REP Configuration, page 25-7](#)
- [REP Configuration Guidelines, page 25-7](#)
- [Configuring the REP Administrative VLAN, page 25-8](#)
- [Configuring REP Interfaces, page 25-10](#)
- [Setting Manual Preemption for VLAN Load Balancing, page 25-13](#)
- [Configuring SNMP Traps for REP, page 25-14](#)

Default REP Configuration

REP is disabled on all interfaces. When enabled, the interface is a regular segment port unless it is configured as an edge port.

When REP is enabled, the task of sending segment topology change notices (STCNs) is disabled, all the VLANs are blocked, and the default administrative VLAN is VLAN 1.

When VLAN load balancing is enabled, the default is manual pre-emption with the delay timer disabled. If VLAN load balancing is not configured, the default after manual pre-emption is to block all the VLANs in the primary edge port.

REP Configuration Guidelines

Follow these guidelines when configuring REP:

- We recommend that you begin by configuring one port and then configure the contiguous ports to minimize the number of segments and the number of blocked ports.
- If more than two ports in a segment fail when no external neighbors are configured, one port goes into a forwarding state for the data path to help maintain connectivity during configuration. In the **show rep interface privileged EXEC** command output, the Port Role for this port shows as *Fail Logical Open*; the Port Role for the other failed port shows as *Fail No Ext Neighbor*. When the external neighbors for the failed ports are configured, the ports go through the alternate port state transitions and eventually go to an open state or remain as the alternate port, based on the alternate port election mechanism.
- REP ports must be Layer 2 dot1Q trunk or PVLAN promiscuous trunk ports.
- Be careful when configuring REP through a Telnet connection. Because REP blocks all VLANs until another REP interface sends a message to unblock it, you might lose connectivity to the switch if you enable REP in a Telnet session that accesses the switch through the same interface.

- You cannot run REP and STP on the same segment or interface.
- If you connect an STP network to the REP segment, be sure that the connection is at the segment edge. An STP connection that is not at the edge could cause a bridging loop because STP does not run on REP segments. All STP BPDUs are dropped at REP interfaces.
- You must configure all trunk and PVLAN promiscuous trunk ports in the segment with the same set of allowed VLANs, or a misconfiguration occurs.
- REP ports follow these rules:
 - If REP is enabled on two ports on a switch, both ports must be either regular segment ports or edge ports.
 - If only one port on a switch is configured in a segment, the port should be an edge port.
 - If two ports on a switch belong to the same segment, both ports must be edge ports or both ports must be regular segment ports.
 - If two ports on a switch belong to the same segment, they must be both edge ports, both regular segment ports, or one regular port and one edge no-neighbor port. An edge port and regular segment port on a switch cannot belong to the same segment.
 - If two ports on a switch belong to the same segment and one is configured as an edge port and one as a regular segment port (a misconfiguration), the edge port is treated as a regular segment port.
- REP interfaces come up in a blocked state and remains in a blocked state until notified that it is safe to unblock. You need to be aware of this to avoid sudden connection losses.
- REP sends all LSL PDUs in untagged frames on the native VLAN. The BPA message sent to the Cisco multicast address is sent on the administration VLAN, which is VLAN 1 by default.
- REP ports can not be configured as one of these port types:
 - SPAN destination port
 - Private VLAN port
 - Tunnel port
 - Access port
- REP is supported on EtherChannels, but not on an individual port that belongs to an EtherChannel.
- There is a maximum of 384 REP segments per switch.

Configuring the REP Administrative VLAN

To avoid the delay introduced by relaying messages in software for link-failure notification or VLAN-blocking notification during load balancing, REP floods packets at the hardware flood layer (HFL) to a regular multicast address. These messages are flooded to the whole network, not just the REP segment. You can control message flooding by configuring an administrative VLAN for the entire domain, or for a particular segment.

Follow these guidelines when configuring a REP administrative VLAN:

- If you do not configure an administrative VLAN, the default is VLAN 1.
- You can create any number of administrative VLANs as long as it is per segment.
- The administrative VLAN cannot be a Remote Switched Port Analyzer (RSPAN) VLAN.

To configure the REP administrative VLAN, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# rep admin vlan <i>vlan-id</i>	Specifies the administrative VLAN ID for the entire domain (all segments). The VLAN ID range is 2 to 4094. The default is VLAN 1. To specify the administrative VLAN ID per segment, enter the rep admin vlan <i>vlan-id segment segment-id</i> global configuration command. To set the admin VLAN ID to 1, enter the no rep admin vlan global configuration command.
Step 3	Switch(config)# end	Returns to privileged EXEC mode.
Step 4	Switch# show interface [<i>interface-id</i>] rep detail	Verifies configuration on the specified REP interface.
Step 5	Switch# copy running-config startup config	(Optional) Saves your entries in the switch startup configuration file.

This example shows how to configure the administrative VLAN as VLAN 100 and verify the configuration by entering the **show interface rep detail** command on one of the REP interfaces:

```
Switch# configure terminal
Switch (config)# rep admin vlan 100
Switch (config)# end

Switch# show interface gigabitethernet1/1 rep detail
GigabitEthernet1/1 REP enabled
Segment-id: 2 (Edge)
PortID: 00010019E7144680
Preferred flag: No
Operational Link Status: TWO_WAY
Current Key: 0002001121A2D5800E4D
Port Role: Open
Blocked Vlan: <empty>
Admin-vlan: 100
Preempt Delay Timer: disabled
Load-balancing block port: none
Load-balancing block vlan: none
STCN Propagate to: none
LSL PDU rx: 3322, tx: 1722
HFL PDU rx: 32, tx: 5
BPA TLV rx: 16849, tx: 508
BPA (STCN, LSL) TLV rx: 0, tx: 0
BPA (STCN, HFL) TLV rx: 0, tx: 0
EPA-ELECTION TLV rx: 118, tx: 118
EPA-COMMAND TLV rx: 0, tx: 0
EPA-INFO TLV rx: 4214, tx: 4190
```

This example shows how to create an administrative VLAN per segment. Here VLAN ID 2 is configured as the administrative VLAN only for REP segment 2. All remaining segments have VLAN 1 as the administrative VLAN.

```
Switch# configure terminal
Switch (config)# rep admin vlan 2 segment 2
Switch (config)# end
```

Configuring REP Interfaces

For REP operation, you need to enable it on each segment interface and identify the segment ID. This step is required and must be done before other REP configuration. You must also configure a primary and secondary edge port on each segment. All other steps are optional.

To enable and configure REP on an interface, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Specifies the interface, and enter interface configuration mode. The interface can be a physical Layer 2 interface or a port channel (logical interface). The port-channel range is 1 to 48.
Step 3	Switch(config-if)# switchport mode trunk or, switchport mode private-vlan trunk promiscuous	<p>Configures the Layer 2 interface as a Layer 2 trunk port.</p> <p>Configures the Layer 2 interface as a PVLAN promiscuous trunk port.</p> <p>For information on command options for PVLAN promiscuous trunk ports, refer to “Configuring a Layer 2 Interface as a Promiscuous PVLAN Trunk Port” on page 21.</p> <p>Note With REP, only the switchport mode private-vlan trunk promiscuous command is supported; other PVLAN trunk related configurations <i>are not</i> supported.</p>

	Command	Purpose
Step 4	<pre>Switch(config-if)# rep segment segment-id [edge [no-neighbor] [primary]] [preferred]</pre>	<p>Enables REP on the interface, and identifies a segment number. The segment ID range is from 1 to 1024. These optional keywords are available.</p> <p>Note You must configure two edge ports, including one primary edge port for each segment.</p> <ul style="list-style-type: none"> Enter edge to configure the port as an edge port. Entering edge without the primary keyword configures the port as the secondary edge port. Each segment has only two edge ports. (Optional) Enter no-neighbor to configure a port with no external REP neighbors as an edge port. The port inherits all properties of edge ports. You can configure them as you would any edge port. (Optional) On an edge port, enter primary to configure the port as the primary edge port, the port on which you can configure VLAN load balancing. <p>Note Although each segment can have only one primary edge port, if you configure edge ports on two different switches and enter the primary keyword on both switches, the configuration is allowed. However, REP selects only one of these ports as the segment primary edge port. You can identify the primary edge port for a segment by entering the show rep topology privileged EXEC command.</p> <ul style="list-style-type: none"> (Optional) Enter preferred to indicate that the port is the preferred alternate port or the preferred port for VLAN load balancing. <p>Note Configuring a port as preferred does not guarantee that it becomes the alternate port; this only provides a slight advantage among equal contenders. The alternate port is usually a previously failed port.</p>
Step 5	<pre>Switch(config-if)# rep stcn {interface interface-id segment id-list stp}</pre>	<p>(Optional) Configures the edge port to send segment topology change notices (STCNs).</p> <ul style="list-style-type: none"> Enter interface <i>interface-id</i> to designate a physical interface or port channel to receive STCNs. Enter segment <i>id-list</i> to identify one or more segments to receive STCNs. The range is 1 to 1024. Enter stp to send STCNs to STP networks.

	Command	Purpose
Step 6	Switch(config-if)# rep block port { id <i>port-id</i> <i>neighbor_offset</i> preferred } vlan { <i>vlan-list</i> all }	<p>(Optional) Configures VLAN load balancing on the primary edge port, identify the REP alternate port in one of three ways, and configure the VLANs to be blocked on the alternate port.</p> <ul style="list-style-type: none"> Enter the id <i>port-id</i> to identify the alternate port by port ID. The port ID is automatically generated for each port in the segment. You can view interface port IDs by entering the show interface interface-id rep [detail] privileged EXEC command. Enter a <i>neighbor_offset</i> number to identify the alternate port as a downstream neighbor from an edge port. The range is from -256 to 256, with negative numbers indicating the downstream neighbor from the secondary edge port. A value of 0 is invalid. Enter -1 to identify the secondary edge port as the alternate port. For an example of neighbor offset numbering, see Figure 25-4 on page 25-5. <p>Note Because you enter this command at the primary edge port (offset number 1), you never enter an offset value of 1 to identify an alternate port.</p> <ul style="list-style-type: none"> Enter preferred to select the regular segment port previously identified as the preferred alternate port for VLAN load balancing. Enter vlan <i>vlan-list</i> to block one VLAN or a range of VLANs. Enter vlan all to block all VLANs. <p>Note Enter this command only on the REP primary edge port.</p>
Step 7	Switch(config-if)# rep preempt delay <i>seconds</i>	<p>(Optional) You must enter this command and configure a preempt time delay if you want VLAN load balancing to automatically trigger after a link failure and recovery. The time delay range is 15 to 300 seconds. The default is manual preemption with no time delay.</p> <p>Note Enter this command only on the REP primary edge port.</p>
Step 8	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 9	Switch# show interface [<i>interface-id</i>] rep [detail]	Verifies the REP interface configuration.
Step 10	Switch# copy running-config startup config	(Optional) Saves your entries in the switch startup configuration file.

Enter the **no** form of each command to return to the default configuration. Enter the **show rep topology** privileged EXEC command to see which port in the segment is the primary edge port.

This example shows how to configure an interface as the primary edge port for segment 1, to send STCNs to segments 2 through 5, and to configure the alternate port as the port with port ID 0009001818D68700 to block all VLANs after a preemption delay of 60 seconds after a segment port failure and recovery.

```
Switch# configure terminal
Switch(config)# interface gigabitethernet1/1
Switch(config-if)# rep segment 1 edge primary
Switch(config-if)# rep stcn segment 2-5
Switch(config-if)# rep block port 0009001818D68700 vlan all
Switch(config-if)# rep preempt delay 60
Switch(config-if)# end
```

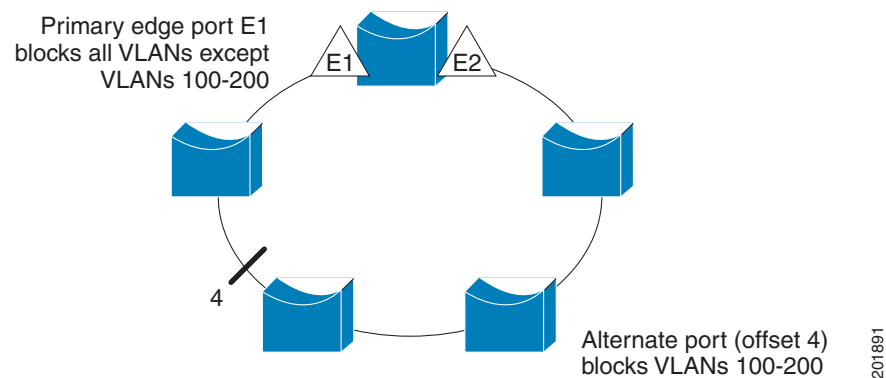

This example shows how to configure the same configuration when the interface has no external REP neighbor:

```
Switch# configure terminal
Switch (config)# interface gigabitethernet1/1
Switch (config-if)# rep segment 1 edge no-neighbor primary
Switch (config-if)# rep stcn segment 2-5
Switch (config-if)# rep block port 0009001818D68700 vlan all
Switch (config-if)# rep preempt delay 60
Switch (config-if)# rep lsl-age-timer 6000
```

This example shows how to configure the VLAN blocking configuration shown in Figure 25-5. The alternate port is the neighbor with neighbor offset number 4. After manual preemption, VLANs 100 to 200 are blocked at this port and all other VLANs are blocked at the primary edge port E1 (Gigabit Ethernet port 1/1).

```
Switch# configure terminal
Switch (config)# interface gigabitethernet1/1
Switch (config-if)# rep segment 1 edge primary
Switch (config-if)# rep block port 4 vlan 100-200
Switch (config-if)# end
```

Figure 25-5 Example of VLAN Blocking



Setting Manual Preemption for VLAN Load Balancing

If you do not enter the **rep preempt delay** *seconds* interface configuration command on the primary edge port to configure a preemption time delay, the default is to manually trigger VLAN load balancing on the segment. Be sure that all other segment configuration has been completed before manually preempting VLAN load balancing. When you enter the **rep preempt segment** command, a confirmation message appears before the command is executed because preemption can cause network disruption.

To manually trigger VLAN load balancing on a segment on the switch that has the segment primary edge port, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Specifies the interface, and enter interface configuration mode. The interface can be a physical Layer 2 interface or a port channel (logical interface). The port-channel range is 1 to 48.

	Command	Purpose
Step 3	Switch(config-if)# rep preempt segment <i>segment-id</i>	Manually triggers VLAN load balancing on the segment. You must confirm the command before it is executed.
Step 4	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 5	Switch# show rep topology	Displays REP topology information.

Configuring SNMP Traps for REP

To configure the switch to send REP-specific traps to notify the SNMP server of link operational status changes and port role changes, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# snmp mib rep trap-rate <i>value</i>	Enables the switch to send REP traps, and set the number of traps sent per second. The range is from 0 to 1000. The default is 0 (no limit imposed; a trap is sent at every occurrence).
Step 3	Switch(config)# end	Returns to privileged EXEC mode.
Step 4	Switch# show running-config	Verifies the REP trap configuration.
Step 5	Switch# copy running-config startup config	(Optional) Saves your entries in the switch startup configuration file.

To remove the trap, enter the **no snmp mib rep trap-rate** global configuration command.

This example configures the switch to send REP traps at a rate of 10 per second:

```
Switch(config)# snmp mib rep trap-rate 10
```

Monitoring REP

To monitor REP, enter the following privileged EXEC commands ([Table 25-1](#)).

Table 25-1 REP Monitoring Commands

Command	Purpose
Switch# show interface [<i>interface-id</i>] rep [<i>detail</i>]	Displays REP configuration and status for a specified interface or for all interfaces.
Switch# show rep topology [<i>segment segment-id</i>] [<i>archive</i>] [<i>detail</i>]	Displays REP topology information for a segment or for all segments, including the primary and secondary edge ports in the segment.



Configuring Optional STP Features

This chapter describes the Spanning Tree Protocol (STP) features supported on the switch. It also provides guidelines, procedures, and configuration examples. To configure STP, see [Chapter 23, “Configuring STP and MST.”](#)

This chapter includes the following major sections:

- [About Root Guard, page 26-2](#)
- [Enabling Root Guard, page 26-2](#)
- [About Loop Guard, page 26-3](#)
- [Enabling Loop Guard, page 26-5](#)
- [About EtherChannel Guard, page 26-6](#)
- [Enabling EtherChannel Guard \(Optional\), page 26-6](#)
- [About STP PortFast Port Types, page 26-7](#)
- [Enabling PortFast Port Types, page 26-8](#)
- [About Bridge Assurance, page 26-11](#)
- [Configuring Bridge Assurance, page 26-13](#)
- [About BPDU Guard, page 26-15](#)
- [Enabling BPDU Guard, page 26-15](#)
- [About PortFast Edge BPDU Filtering, page 26-16](#)
- [Enabling PortFast Edge BPDU Filtering, page 26-17](#)
- [About UplinkFast, page 26-19](#)
- [Enabling UplinkFast, page 26-20](#)
- [About BackboneFast, page 26-21](#)
- [Enabling BackboneFast, page 26-23](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About Root Guard

Spanning Tree root guard forces an interface to become a designated port, to protect the current root status and prevent surrounding switches from becoming the root switch.

When you enable root guard on a per-port basis, it is automatically applied to all of the active VLANs to which that port belongs. When you disable root guard, it is disabled for the specified port and the port automatically goes into the listening state.

When a switch that has ports with root guard enabled detects a new root, the ports enter the root-inconsistent state. The switch no longer detects a new root and its ports automatically go into the listening state.

Enabling Root Guard

To enable root guard on a Layer 2 access port (to force it to become a designated port), perform this task:

	Command	Purpose
Step 1	Switch(config)# interface {{fastethernet gigabitethernet tengigabitethernet} slot/port}	Specifies an interface to configure.
Step 2	Switch(config-if)# [no] spanning-tree guard root	Enables root guard. Use the no keyword to disable root guard.
Step 3	Switch(config-if)# end	Exits configuration mode.
Step 4	Switch# show spanning-tree	Verifies the configuration.

This example shows how to enable root guard on Fast Ethernet interface 5/8:

```
Switch(config)# interface fastethernet 5/8
Switch(config-if)# spanning-tree guard root
Switch(config-if)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show running-config interface fastethernet 5/8
Building configuration...

Current configuration: 67 bytes
!
interface FastEthernet5/8
  switchport mode access
  spanning-tree guard root
end

Switch#
```

This example shows how to determine whether any ports are in root inconsistent state:

```
Switch# show spanning-tree inconsistentports
```

Name	Interface	Inconsistency
VLAN0001	FastEthernet3/1	Root Inconsistent
VLAN0001	FastEthernet3/2	Root Inconsistent
VLAN1002	FastEthernet3/1	Root Inconsistent
VLAN1002	FastEthernet3/2	Root Inconsistent
VLAN1003	FastEthernet3/1	Root Inconsistent
VLAN1003	FastEthernet3/2	Root Inconsistent
VLAN1004	FastEthernet3/1	Root Inconsistent
VLAN1004	FastEthernet3/2	Root Inconsistent
VLAN1005	FastEthernet3/1	Root Inconsistent
VLAN1005	FastEthernet3/2	Root Inconsistent

```
Number of inconsistent ports (segments) in the system :10
```

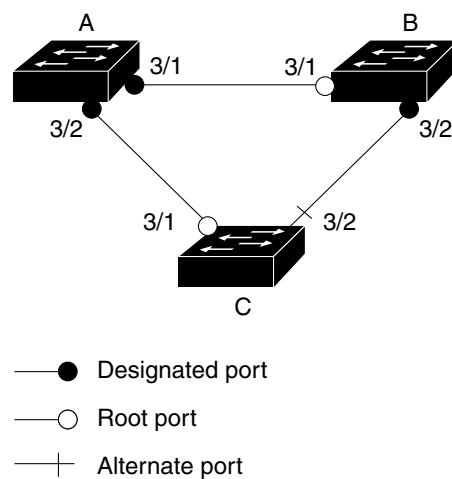
About Loop Guard

Loop guard helps prevent bridging loops that could occur because of a unidirectional link failure on a point-to-point link. When enabled globally, loop guard applies to all point-to-point ports on the system. Loop guard detects root ports and blocked ports and ensures that they keep receiving BPDUs from their designated port on the segment. If a loop-guard-enabled root or blocked port stop receiving BPDUs from its designated port, it transitions to the blocking state, assuming there is a physical link error on this port. The port recovers from this state as soon as it receives a BPDU.

You can enable loop guard on a per-port basis. When you enable loop guard, it is automatically applied to all of the active instances or VLANs to which that port belongs. When you disable loop guard, it is disabled for the specified ports. Disabling loop guard moves all loop-inconsistent ports to the listening state.

If you enable loop guard on a channel and the first link becomes unidirectional, loop guard blocks the entire channel until the affected port is removed from the channel. [Figure 26-1](#) shows loop guard in a triangular switch configuration.

Figure 26-1 Triangular Switch Configuration with Loop Guard



55772

Figure 26-1 illustrates the following configuration:

- Switches A and B are distribution switches.
- Switch C is an access switch.
- Loop guard is enabled on ports 3/1 and 3/2 on Switches A, B, and C.

Enabling loop guard on a root switch has no effect but provides protection when a root switch becomes a nonroot switch.

Follow these guidelines when using loop guard:

- Do not enable loop guard on PortFast edge-enabled or dynamic VLAN ports.
- Do not enable loop guard if root guard is enabled.

Loop guard interacts with other features as follows:

- Loop guard does not affect the functionality of UplinkFast or BackboneFast.
- Enabling loop guard on ports that are not connected to a point-to-point link does not work.
- Root guard forces a port to always be the root port. Loop guard is effective only if the port is a root port or an alternate port. You cannot enable loop guard and root guard on a port at the same time.
- Loop guard uses the ports known to spanning tree. Loop guard can take advantage of logical ports provided by the Port Aggregation Protocol (PAgP). However, to form a channel, all the physical ports grouped in the channel must have compatible configurations. PAgP enforces uniform configurations of root guard or loop guard on all the physical ports to form a channel.
 - Spanning tree always chooses the first operational port in the channel to send the BPDUs. If that link becomes unidirectional, loop guard blocks the channel, even if other links in the channel are functioning properly.
 - If a set of ports that are already blocked by loop guard are grouped together to form a channel, spanning tree loses all the state information for those ports and the new channel port may obtain the forwarding state with a designated role.
 - If a channel is blocked by loop guard and the channel breaks, spanning tree loses all the state information. The individual physical ports may obtain the forwarding state with the designated role, even if one or more of the links that formed the channel are unidirectional.

**Note**

You can enable UniDirectional Link Detection (UDLD) to help isolate the link failure. A loop may occur until UDLD detects the failure, but loop guard is not able to detect it.

- Loop guard has no effect on a disabled spanning tree instance or a VLAN.

Enabling Loop Guard

You can enable loop guard globally or per-port.

Loop Guard can be enabled only on network and normal spanning tree port types.

To enable loop guard globally on the switch, perform this task:

	Command	Purpose
Step 1	Switch(config)# spanning-tree loopguard default	Enables loop guard globally on the switch.
Step 2	Switch(config)# end	Exits configuration mode.
Step 3	Switch# show spanning tree interface 4/4	Verifies the configuration impact on a port.

This example shows how to enable loop guard globally:

```
Switch(config)# spanning-tree loopguard default
Switch(config)# Ctrl-Z
```

This example shows how to verify the previous configuration of port 4/4:

```
Switch# show spanning-tree interface fastethernet 4/4 detail
Port 196 (FastEthernet4/4) of VLAN0010 is forwarding
  Port path cost 1000, Port priority 160, Port Identifier 160.196.
  Designated root has priority 32768, address 00d0.00b8.140a
  Designated bridge has priority 32768, address 00d0.00b8.140a
  Designated port id is 160.196, designated path cost 0
  Timers:message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state:1
  The port is in the portfast mode by portfast trunk configuration
  Link type is point-to-point by default
  Bpdu filter is enabled
  Loop guard is enabled by default on the port
  BPDU:sent 0, received 0
```

To enable loop guard on an interface, perform this task:

	Command	Purpose
Step 1	Switch(config)# interface {type slot/port} {port-channel port_channel_number}	Selects an interface to configure.
Step 2	Switch(config-if)# spanning-tree guard loop	Configures loop guard.
Step 3	Switch(config)# end	Exits configuration mode.
Step 4	Switch# show spanning tree interface 4/4	Verifies the configuration impact on that port.

This example shows how to enable loop guard on port 4/4:

```
Switch(config)# interface fastEthernet 4/4
Switch(config-if)# spanning-tree guard loop
Switch(config-if)# ^Z
```

This example shows how to verify the configuration impact on port 4/4:

```
Switch# show spanning-tree interface fastEthernet 4/4 detail
Port 196 (FastEthernet4/4) of VLAN0010 is forwarding
  Port path cost 1000, Port priority 160, Port Identifier 160.196.
  Designated root has priority 32768, address 00d0.00b8.140a
  Designated bridge has priority 32768, address 00d0.00b8.140a
  Designated port id is 160.196, designated path cost 0
  Timers:message age 0, forward delay 0, hold 0
  Number of transitions to forwarding state:1
  The port is in the portfast mode by portfast trunk configuration
  Link type is point-to-point by default
  Bpdu filter is enabled
  Loop guard is enabled on the port
  BPDU:sent 0, received 0
Switch#
```

About EtherChannel Guard

EtherChannel guard allows you to detect an EtherChannel misconfiguration between the switch and a connected device. A misconfiguration can occur if the interfaces of a switch are manually configured in an EtherChannel, and one or more interfaces on the other device are not. For EtherChannel configuration guidelines, see the [“EtherChannel Configuration Guidelines and Restrictions” section on page 27-6](#).



Note

EtherChannel guard applies only to EtherChannels in forced mode (that is, manually configured) rather than through PAgP or LACP.

If the switch detects a misconfiguration on the other device, EtherChannel guard error-disables all interfaces in the EtherChannel bundle, and displays an error message.

You can enable this feature with the **spanning-tree etherchannel guard misconfig** global configuration command.

Enabling EtherChannel Guard (Optional)

You can enable EtherChannel guard to detect an EtherChannel misconfiguration if your switch is running PVST+, rapid PVST+, or MSTP.

To enable EtherChannel guard, perform this task:

	Command	Purpose
Step 1	Switch(config)# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# spanning-tree etherchannel guard misconfig	Enables EtherChannel guard.
Step 3	Switch(config)# end	Returns to privileged EXEC mode.
Step 4	Switch(config)# show spanning-tree summary	Verifies your entries.
Step 5	Switch(config)# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To disable the EtherChannel guard feature, use the **no spanning-tree etherchannel guard misconfig** global configuration command.

Use the **show interfaces status err-disabled** privileged EXEC command to show which switch ports are disabled because of an EtherChannel misconfiguration. On the remote device, you can enter the **show etherchannel summary** privileged EXEC command to verify the EtherChannel configuration.

After the configuration is corrected, enter the **shutdown** and **no shutdown** interface configuration commands on the port-channel interfaces that were misconfigured.

About STP PortFast Port Types

You can configure a spanning tree port as an edge port, a network port, or a normal port. A port can be in only one of these states at a given time. The default spanning tree port type is normal. You can configure the port type either globally or per interface.

Depending on the type of device to which the interface is connected, you can configure a spanning tree port as one of these port types:

- A PortFast edge port—is connected to a Layer 2 host. This can be either an access port or an edge trunk port (**portfast edge trunk**). This type of port interface immediately transitions to the forwarding state, bypassing the listening and learning states. Use PortFast edge on Layer 2 access ports connected to a single workstation or server to allow those devices to connect to the network immediately, rather than waiting for spanning tree to converge.

Even if the interface receives a bridge protocol data unit (BPDU), spanning tree does not place the port into the blocking state. Spanning tree sets the port's operating state to *non-port fast* even if the configured state remains *port fast edge* and starts participating in the topology change.



Note If you configure a port connected to a Layer 2 switch or bridge as an edge port, you might create a bridging loop.

- A PortFast network port—is connected only to a Layer 2 switch or bridge.
Bridge Assurance is enabled only on PortFast network ports. For more information, see [About Bridge Assurance, page 26-11](#).



Note If you configure a port that is connected to a Layer 2 host as a spanning tree network port, the port will automatically move into the blocking state.

- A PortFast normal port—is the default type of spanning tree port.



Note

Beginning with Cisco IOS Release 15.2(4)E, or IOS XE 3.8.0E, if you enter the **spanning-tree portfast** [trunk] command in the global or interface configuration mode, the system automatically saves it as **spanning-tree portfast edge** [trunk].

Enabling PortFast Port Types

- [Configuring the PortFast Default State Globally, page 26-8](#)
- [Configuring a PortFast Edge Port on a Specified Interface, page 26-8](#)
- [Configuring a PortFast Network Port on a Specified Interface, page 26-10](#)

Configuring the PortFast Default State Globally

To configure the default PortFast state, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 1	Switch(config)# spanning-tree portfast [edge network normal] default	Configures the default state for all interfaces on the switch. You have these options: <ul style="list-style-type: none"> • (Optional) edge—Configures all interfaces as edge ports. This assumes all ports are connected to hosts/servers. • (Optional) network—Configures all interfaces as spanning tree network ports. This assumes all ports are connected to switches and bridges. Bridge Assurance is enabled on all network ports by default. • (Optional) normal—Configures all interfaces as normal spanning tree ports. These ports can be connected to any type of device. • default—The default port type is normal.
Step 2	Switch(config)# end	Exits configuration mode.

Configuring a PortFast Edge Port on a Specified Interface

Interfaces configured as edge ports immediately transition to the forwarding state, without passing through the blocking or learning states, on linkup. To configure an edge port on a specified interface, perform this task:



Note

Because the purpose of this type of port is to minimize the time that access ports must wait for spanning tree to converge, it is most effective when used on access ports. If you enable PortFast edge on a port connecting to another switch, you risk creating a spanning tree loop.

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface {{ fastethernet gigabitethernet tengigabitethernet } slot/port } { port-channel port_channel_number }	Specifies an interface to configure.

	Command	Purpose
Step 3	Switch(config-if)# spanning-tree portfast edge [trunk]	Enables edge behavior on a Layer 2 access port connected to an end workstation or server. (Optional) trunk —Enables edge behavior on a trunk port. Use this keyword if the link is a trunk. Use this command only on ports that are connected to end host devices that terminate VLANs and from which the port should never receive STP BPDUs. Such end host devices include workstations, servers, and ports on routers that are not configured to support bridging. Use the no version of the command to disable PortFast edge.
Step 4	Switch(config-if)# end	Exits global configuration mode
Step 5	Switch# show running interface {{ fastethernet gigabitethernet tengigabitethernet } <i>slot/port</i> } { port-channel <i>port_channel_number</i> }	Verifies the configuration.
Step 6	Switch# show spanning-tree interface {{ fastethernet gigabitethernet tengigabitethernet } <i>slot/port</i> } { port-channel <i>port_channel_number</i> } portfast edge	Displays spanning-tree PortFast information for the specified interface.

This example shows how to enable edge behavior on GigabitEthernet interface 5/7 and verify configuration:

```
Switch# configure terminal
Switch(config)# interface fastethernet 5/7
Switch(config-if)# spanning-tree portfast edge
Switch(config-if)# end
Switch#

Switch# show running-config interface fastethernet 5/7
Building configuration...
Current configuration:
!
interface GigabitEthernet5/7
 no ip address
 switchport
 switchport access vlan 200
 switchport mode access
 spanning-tree portfast edge
end
```

This example shows how you can display that port GigabitEthernet 5/8 is currently in the edge state:

```
Switch# show spanning-tree vlan 200
VLAN0200
Spanning tree enabled protocol rstp
Root ID Priority 2
Address 001b.2a68.5fc0
Cost 3
Port 125 (GigabitEthernet5/9)
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 2 (priority 0 sys-id-ext 2)
Address 7010.5c9c.5200
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Aging Time 0 sec
```

Interface	Role	Sts	Cost	Prio.	Nbr	Type
-----	-----	-----	-----	-----	-----	-----
Gi5/7	Desg	FWD	4	128.1	P2p	Edge

Configuring a PortFast Network Port on a Specified Interface

Ports that are connected to Layer 2 switches and bridges can be configured as network ports.



Note Bridge Assurance is enabled only on PortFast network ports. For more information, see [About Bridge Assurance, page 26-11](#)

To configure a port as a network port, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface {{fastethernet gigabitethernet tengigabitethernet} slot/port} {port-channel port_channel_number}	Specifies an interface to configure.
Step 3	Switch(config-if)# spanning-tree portfast network	Configures the port as a network port. If you have enabled Bridge Assurance globally, it automatically runs on a spanning tree network port. Use the no keyword to disable PortFast.
Step 4	Switch(config-if)# end	Exits configuration mode.
Step 5	Switch# show running interface {{fastethernet gigabitethernet tengigabitethernet} slot/port} {port-channel port_channel_number}	Verifies the configuration.

This example shows how to configure GigabitEthernet interface 5/8 as a network port and verify configuration:

```
Switch# configure terminal
Switch(config)# interface gigabitethernet 5/8
Switch(config-if)# spanning-tree portfast network
Switch(config-if)# end
Switch#

Switch# show running-config interface gigabitethernet 5/8
Building configuration...
Current configuration:
!
interface GigabitEthernet5/8
 no ip address
 switchport
 switchport access vlan 200
 switchport mode access
 spanning-tree portfast network
end
```

About Bridge Assurance

You can use Bridge Assurance to help prevent looping conditions that are caused by unidirectional links (one-way traffic on a link or port), or a malfunction in a neighboring switch. Here, a malfunction refers to a switch that is not able to run STP any more, while still forwarding traffic (a brain dead switch).

BPDUs are sent out on all operational network ports, including alternate and backup ports, for each hello time period. Bridge Assurance monitors the receipt of BPDUs on point-to-point links on all network ports. When a port does not receive BPDUs within the allotted hello time period, the port is put into a blocked state (the same as a port inconsistent state, which stops forwarding of frames). When the port resumes receipt of BPDUs, the port resumes normal spanning tree operations.



Note Only Rapid PVST+ and MST spanning tree protocols support Bridge Assurance. PVST+ does not support Bridge Assurance.

This example shows how Bridge Assurance protects your network from bridging loops. Here, [Figure 26-2](#) shows a normal STP topology, and [Figure 26-3](#) demonstrates a potential network problem when the device fails (brain dead) and Bridge Assurance is not enabled on the network.

Figure 26-2 Network with Normal STP Topology

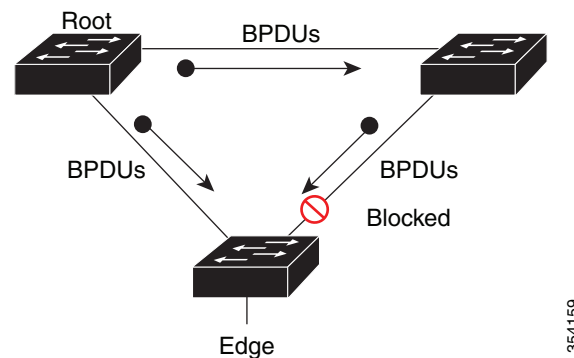


Figure 26-3 Network Loop Due to a Malfunctioning Switch

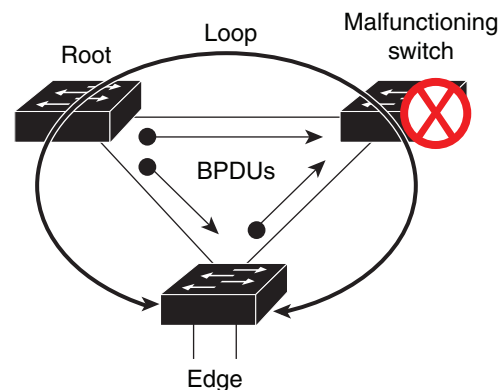


Figure 26-4 shows the network with Bridge Assurance enabled, and the STP topology progressing normally with bidirectional BDPUs issuing from every STP network port. Figure 26-5 shows how the potential network problem shown in Figure 26-3 does not occur when you have Bridge Assurance enabled on your network.

Figure 26-4 Network with STP Topology Running Bridge Assurance

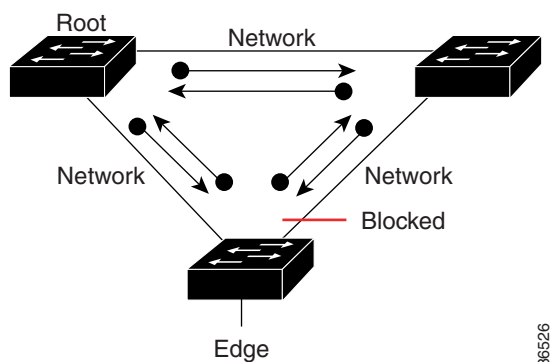
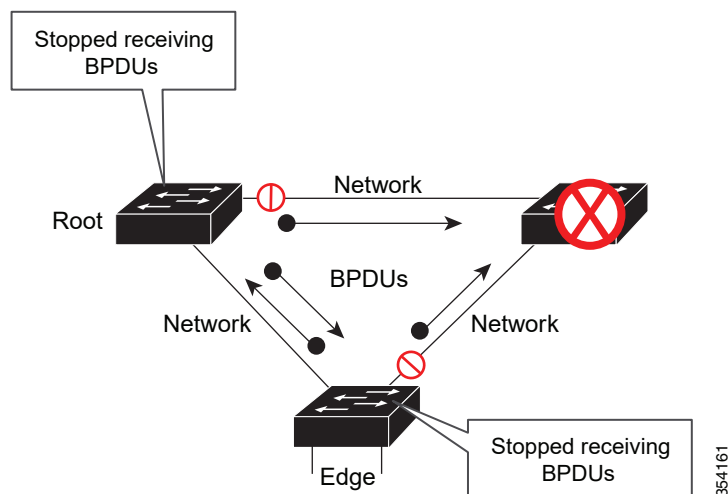


Figure 26-5 Network Problem Averted with Bridge Assurance Enabled



The system generates syslog messages when a port is block or unblocked. The following sample outputs show the log that is generated for each of these states:

Blocked port:

```
Sep 17 09:48:16.249 PDT: %SPANTREE-2-BRIDGE_ASSURANCE_BLOCK: Bridge Assurance blocking
port GigabitEthernet5/8 on VLAN0200. (stack-dut-R4-4)
```

Unblocked Port:

```
Sep 17 09:48:58.426 PDT: %SPANTREE-2-BRIDGE_ASSURANCE_UNBLOCK: Bridge Assurance
unblocking port GigabitEthernet5/8 on VLAN0200. (stack-dut-R4-4)
```

Observe these guidelines when configuring Bridge Assurance:

- It can be enabled or disabled globally.
- It applies to all operational network ports, including alternate and backup ports.
- Only Rapid PVST+ and MST spanning tree protocols support Bridge Assurance. PVST+ does not support Bridge Assurance.
- For Bridge Assurance to work properly, it must be supported and configured on both ends of a point-to-point link. If the device on one side of the link has Bridge Assurance enabled and the device on the other side does not, then the connecting port is blocked (a Bridge Assurance inconsistent state). We recommend that you enable Bridge Assurance throughout your network.
- To enable Bridge Assurance on a port, BPDU filtering and BPDU Guard must be disabled.
- You can enable Bridge Assurance in conjunction with Loop Guard.
- You can enable Bridge Assurance in conjunction with Root Guard. The latter is designed to provide a way to enforce the root bridge placement in the network.

Configuring Bridge Assurance

	Command	Purpose
Step 1	Switch # configure terminal	Enters the global configuration mode.
Step 2	Switch(config)# spanning-tree bridge assurance	Enables Bridge Assurance on all network ports on the switch. Bridge Assurance is enabled by default. Use the no version of the command to disable the feature. Disabling Bridge Assurance causes all configured network ports to behave as normal spanning tree ports.
Step 3	Switch(config-if)# end	Exits configuration mode.
Step 4	Switch# show spanning-tree summary	Displays spanning tree information and shows if Bridge Assurance is enabled

This example show how to display spanning tree information and verify if Bridge Assurance is enabled. Look for these details in the output:

- Portfast Default—Network
- Bridge Assurance—Enabled

```
Switch# show spanning-tree summary
Switch is in rapid-pvst mode
Root bridge for: VLAN0199-VLAN0200, VLAN0128
EtherChannel misconfig guard      is enabled
Extended system ID                is enabled
Portfast Default                  is network
Portfast Edge BPDU Guard Default  is disabled
Portfast Edge BPDU Filter Default is disabled
Loopguard Default                 is enabled
PVST Simulation Default           is enabled but inactive in rapid-pvst mode
Bridge Assurance                  is enabled
UplinkFast                        is disabled
BackboneFast                      is disabled
```

Configured Pathcost method used is short

Name	Blocking	Listening	Learning	Forwarding	STP Active
VLAN0199	0	0	0	5	5
VLAN0200	0	0	0	4	4
VLAN0128	0	0	0	4	4
3 vlans	0	0	0	13	13

This example shows how to verify if GigabitEthernet 5/8 (configured as a network port), is in a normal state. (From the **show spanning-tree summary** output above, we know that Bridge Assurance is enabled on GigabitEthernet 5/8).

Switch# **show spanning-tree vlan**

Sep 17 09:51:36.370 PDT: %SYS-5-CONFIG_I: Configured from console by console2

VLAN0200

Spanning tree enabled protocol rstp

Root ID Priority 2
 Address 7010.5c9c.5200
 This bridge is the root
 Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 2 (priority 0 sys-id-ext 2)
 Address 7010.5c9c.5200
 Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
 Aging Time 0 sec

Interface	Role	Sts	Cost	Prio.Nbr	Type
Gi5/7	Desg	FWD	4	128.1	P2p Edge
Gi5/8	Desg	FWD	3	128.480	P2p Network
Gi5/9	Desg	FWD	4	128.169	P2p Edge
Gi5/10	Desg	FWD	4	128.215	P2p Network

This example shows how port GigabitEthernet 5/8 (configured as a network port), is currently in the Bridge Assurance inconsistent state:



Note The output shows the port type as network and *BA_Inc, indicating that the port is in an inconsistent state.

Switch# **show spanning-tree vlan**

VLAN200

Spanning tree enabled protocol rstp

Root ID Priority 32778
 Address 0002.172c.f400
 This bridge is the root
 Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32778 (priority 32768 sys-id-ext 10)
 Address 0002.172c.f400
 Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
 Aging Time 300

Interface	Role	Sts	Cost	Prio.Nbr	Type
Gia5/8	Desg	BKN*4		128.270	Network, P2p *BA_Inc

About BPDU Guard

Spanning Tree BPDU guard shuts down PortFast edge-configured interfaces that receive BPDUs, rather than putting them into the spanning tree blocking state.

When configured globally, BPDU Guard is only effective on ports in the operational PortFast edge state. In a valid configuration, PortFast edge-configured interfaces do not receive BPDUs. Reception of a BPDU by a PortFast edge-configured interface signals an invalid configuration, such as connection of an unauthorized device.

BPDU guard provides a secure response to invalid configurations, because the administrator must manually put the interface back in service.



Note

When the BPDU guard feature is enabled, spanning tree applies the BPDU guard feature to all PortFast-configured interfaces. BPDU Guard shuts down that interface if a BPDU is received, regardless of the PortFast port type configuration.



Note

To prevent the port from shutting down, use the **errdisable detect cause bpduguard shutdown vlan** global configuration command to shut down only the offending VLAN on the port where the violation occurred.

Enabling BPDU Guard

Enabling BPDU Guard Globally

To globally enable BPDU guard on edge ports that receive BPDUs, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	
Step 2	Switch(config)# spanning-tree portfast edge bpduguard default	Enables BPDU Guard globally by default on all edge ports of the switch. Use the no version of the command to disable BPDU guard.
Step 3	Switch(config)# end	Exits configuration mode.
Step 4	Switch# show spanning-tree summary	Verifies the BPDU configuration.

This example shows how to enable BPDU guard:

```
Switch(config)# spanning-tree portfast edge bpduguard default
Switch(config)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show spanning-tree summary
Root bridge for: Bridge VLAN0025
EtherChannel misconfiguration guard is enabled
```

```
Extended system ID                is enabled
PortFast Edge BPDU Guard Default  is enabled
Portfast Edge BPDU Filter Default is disabled
Portfast Default                  is edge
Bridge Assurance                  is enabled
Loopguard                        is disabled
UplinkFast                       is disabled
BackboneFast                     is disabled
Pathcost method used is short

Name                               Blocking Listening Learning Forwarding STP Active
-----
2 vlans                           0           0           0           3           3
```

Enabling BPDU Guard on a Specified Interface

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface {{fastethernet gigabitethernet tengigabitethernet} slot/port} {port-channel port_channel_number}	Specifies an interface to configure.
Step 3	Switch(config-if)# spanning-tree portfast edge bpduguard default	Enables BPDU Guard on the specified edge port. Use the no keyword to disable BPDU guard.
Step 4	Switch(config)# end	Exits configuration mode.
Step 5	Switch# show spanning-tree summary	Verifies the BPDU configuration.

About PortFast Edge BPDU Filtering

Cisco IOS Release 12.2(31)SGA and later support PortFast edge BPDU filtering, which allows the administrator to prevent the system from sending or even receiving BPDUs on specified ports.

When configured globally, PortFast edge BPDU filtering applies to all operational PortFast edge ports. Ports in an operational PortFast edge state are supposed to be connected to hosts that typically drop BPDUs. If an operational PortFast edge port receives a BPDU, it immediately loses its operational PortFast edge status. In that case, PortFast edge BPDU filtering is disabled on this port and STP resumes sending BPDUs on this port.

PortFast edge BPDU filtering can also be configured on a per-port basis. When PortFast edge BPDU filtering is explicitly configured on a port, it does not send any BPDUs and drops all BPDUs it receives.



Caution

Explicitly configuring PortFast edge BPDU filtering on a port that is not connected to a host can result in bridging loops, because the port ignores any BPDU it receives and goes to the forwarding state.

When you enable PortFast edge BPDU filtering globally and set this port configuration as the default for PortFast edge BPDU filtering (see the [“Enabling BackboneFast” section on page 26-23](#)), PortFast enables or disables PortFast edge BPDU filtering.

If the port configuration is not set to default, then the PortFast edge configuration does not affect PortFast edge BPDU filtering. Table 26-1 lists all the possible PortFast edge BPDU filtering combinations. PortFast edge BPDU filtering allows access ports to move directly to the forwarding state as soon as the end hosts are connected.

Table 26-1 PortFast Edge BPDU Filtering Port Configurations

Per-Port Configuration	Global Configuration	PortFast Edge State	PortFast Edge BPDU Filtering State
Default	Enable	Enable	Enable ¹
Default	Enable	Disable	Disable
Default	Disable	Not applicable	Disable
Disable	Not applicable	Not applicable	Disable
Enable	Not applicable	Not applicable	Enable

1. The port transmits at least 10 BPDUs. If this port receives any BPDUs, then PortFast edge and PortFast edge BPDU filtering are disabled.

Enabling PortFast Edge BPDU Filtering

Enabling PortFast Edge BPDU Filtering Globally

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# spanning-tree portfast edge bpdupfilter default	Enables BPDU filtering globally by default on all edge ports of the switch. Use the no prefix to disable BPDU filtering by default on all edge ports of the switch.
Step 3	Switch# show spanning-tree summary totals	Verifies the BPDU configuration.

This example shows how to enable PortFast edge BPDU filtering as default on all edge ports and verify the configuration in PVST+ mode :

```
Switch(config)# spanning-tree portfast edge bpdupfilter default
Switch(config)# exit
```

```
Switch# show spanning-tree summary totals
Root bridge for: Bridge VLAN0025
EtherChannel misconfiguration guard is enabled
Extended system ID           is enabled
PortFast Edge BPDU Guard Default is disabled
Portfast Edge BPDU Filter Default is enabled
Portfast Default              is edge
Bridge Assurance               is enabled
Loopguard                     is disabled
UplinkFast                    is disabled
BackboneFast                   is disabled
Pathcost method used is long
```

```
Name                Blocking Listening Learning Forwarding STP Active
```

2 vlans	0	0	0	3	3
---------	---	---	---	---	---



Note

For PVST+ information, see [Chapter 23, “Configuring STP and MST.”](#)

Enabling PortFast Edge BPDU Filtering on a Specified Interface

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface {{ fastethernet gigabitethernet tengigabitethernet } <i>slot/port</i> } {port-channel port_channel_number}	Specifies an interface to configure.
Step 3	Switch(config-if)# spanning-tree bpdufilter [enable disable]	Enables or Disables BPDU filtering.
Step 4	Switch# show spanning-tree interface { <i>type slot/port</i> }	Verifies the configuration.

This example shows how to enable PortFast edge BPDU filtering on port 4/4:

```
Switch(config)# interface fastethernet 4/4
Switch(config-if)# spanning-tree bpdufilter enable
Switch(config-if)# end
```

This example shows how to verify that PortFast edge BPDU filtering is enabled:

```
Switch# show spanning-tree interface fastethernet 4/4

Vlan          Role Sts Cost      Prio.Nbr Status
-----
VLAN0010      Desg FWD 1000     160.196 Edge P2p
```

This example shows more detail on the port:

```
Switch# show spanning-tree interface fastEthernet 4/4 detail
Port 196 (FastEthernet4/4) of VLAN0002 is forwarding
Port path cost 4, Port priority 128, Port Identifier 128.269.
Designated root has priority 32770, address 0002.172c.f400
Designated bridge has priority 32770, address 0002.172c.f400
Designated port id is 128.269, designated path cost 0
Timers: message age 0, forward delay 0, hold 0
Number of transitions to forwarding state: 1
Link type is point-to-point by default
Loop guard is enabled by default on the port
The port is in portfast edge trunk mode
Link type is point-to-point by default
BPDU:sent 2183, received 0
Switch#
```

About UplinkFast



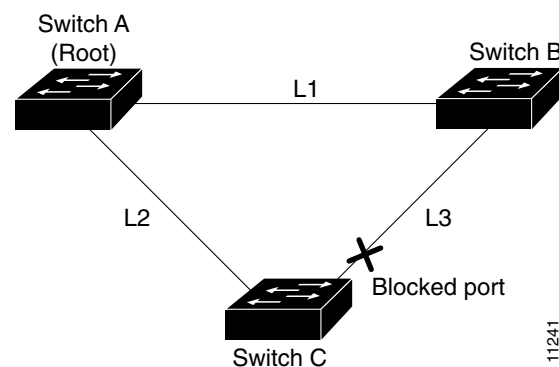
Note

UplinkFast is most useful in wiring-closet switches. This feature might not be useful for other types of applications.

Spanning Tree UplinkFast provides fast convergence after a direct link failure and uses uplink groups to achieve load balancing between redundant Layer 2 links. Convergence is the speed and ability of a group of internetworking devices running a specific routing protocol to agree on the topology of an internetwork after a change in that topology. An uplink group is a set of Layer 2 interfaces (per VLAN), only one of which is forwarding at any given time. Specifically, an uplink group consists of the root port (which is forwarding) and a set of blocked ports, except for self-looping ports. The uplink group provides an alternate path in case the currently forwarding link fails.

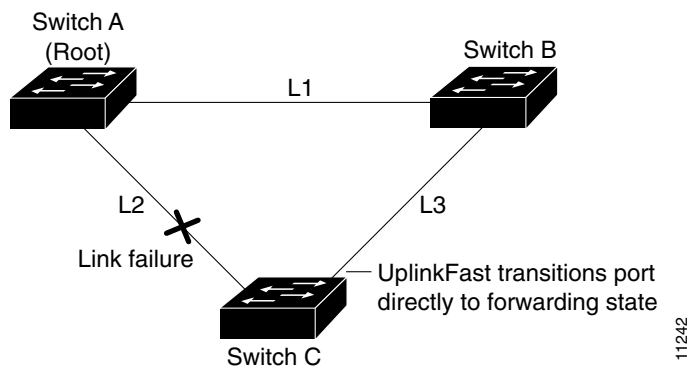
Figure 26-6 shows an example of a topology with no link failures. Switch A, the root switch, is connected directly to Switch B over link L1 and to Switch C over link L2. The Layer 2 interface on Switch C that is connected directly to Switch B is in the blocking state.

Figure 26-6 UplinkFast Before Direct Link Failure



If Switch C detects a link failure on the currently active link L2 on the root port (a direct link failure), UplinkFast unblocks the blocked port on Switch C and transitions it to the forwarding state without going through the listening and learning states, as shown in Figure 26-7. This switchover takes approximately one to five seconds.

Figure 26-7 UplinkFast After Direct Link Failure



Enabling UplinkFast

UplinkFast increases the bridge priority to 49,152 and adds 3000 to the spanning tree port cost of all interfaces on the switch, making it unlikely that the switch becomes the root switch. The *max_update_rate* value represents the number of multicast packets transmitted per second (the default is 150 packets per second [pps]).

UplinkFast cannot be enabled on VLANs that have been configured for bridge priority. To enable UplinkFast on a VLAN with bridge priority configured, restore the bridge priority on the VLAN to the default value by entering a **no spanning-tree vlan *vlan_ID* priority** command in global configuration mode.



Note

When you enable UplinkFast, it affects all VLANs on the switch. You cannot configure UplinkFast on an individual VLAN.

To enable UplinkFast, perform this task:

	Command	Purpose
Step 1	Switch(config)# [no] spanning-tree uplinkfast [max-update-rate <i>max_update_rate</i>]	Enables UplinkFast. Use the no keyword to disable UplinkFast and restore the default rate, use the command.
Step 2	Switch(config)# end	Exits configuration mode.
Step 3	Switch# show spanning-tree vlan <i>vlan_ID</i>	Verifies that UplinkFast is enabled on that VLAN.

This example shows how to enable UplinkFast with a maximum update rate of 400 pps:

```
Switch(config)# spanning-tree uplinkfast max-update-rate 400
Switch(config)# exit
Switch#
```

This example shows how to verify which VLANS have UplinkFast enabled:

```
Switch# show spanning-tree uplinkfast
UplinkFast is enabled
```

Station update rate set to 150 packets/sec.

```
UplinkFast statistics
-----
Number of transitions via uplinkFast (all VLANs)           :14
Number of proxy multicast addresses transmitted (all VLANs) :5308

Name                Interface List
-----
VLAN1                Fa6/9 (fwd) , Gi5/7
VLAN2                Gi5/7 (fwd)
VLAN3                Gi5/7 (fwd)
VLAN4
VLAN5
VLAN6
VLAN7
VLAN8
VLAN10
VLAN15
VLAN1002            Gi5/7 (fwd)
```

```

VLAN1003          Gi5/7 (fwd)
VLAN1004          Gi5/7 (fwd)
VLAN1005          Gi5/7 (fwd)
Switch#

```

About BackboneFast

BackboneFast is a complementary technology to UplinkFast. UplinkFast is designed to quickly respond to failures on links directly connected to leaf-node switches, but it does not help with indirect failures in the backbone core. BackboneFast optimizes the topology based on the Max Age setting. It allows the default convergence time for indirect failures to be reduced from 50 seconds to 30 seconds. However, it never eliminates forward delays and offers no assistance for direct failures.



Note

BackboneFast should be enabled on every switch in your network.

Sometimes a switch receives a BPDU from a designated switch that identifies the root bridge and the designated bridge as the same switch. Because this should not happen, the BPDU is considered inferior.

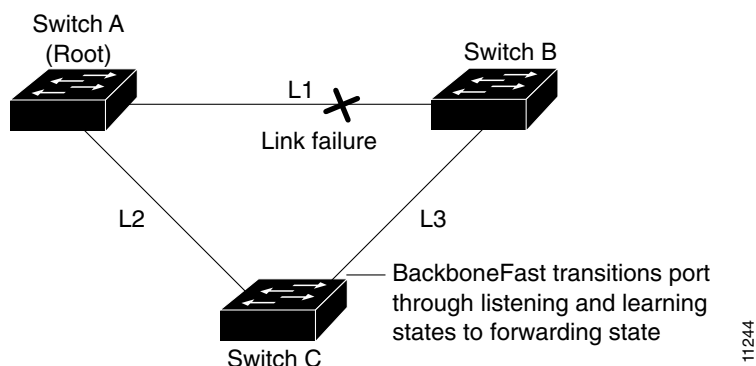
BPDU is considered inferior when a link from the designated switch has lost its link to the root bridge. The designated switch transmits the BPDU with the information that it is now the root bridge as well as the designated bridge. The receiving switch ignores the inferior BPDU for the time defined by the Max Age setting.

After receiving inferior BPDUs, the receiving switch tries to determine if there is an alternate path to the root bridge.

- If the port that the inferior BPDUs are received on is already in blocking mode, then the root port and other blocked ports on the switch become alternate paths to the root bridge.
- If the inferior BPDUs are received on a root port, then all presently blocking ports become the alternate paths to the root bridge. Also, if the inferior BPDUs are received on a root port and no other blocking ports exist on the switch, the receiving switch assumes that the link to the root bridge is down and the time defined by the Max Age setting expires, which turns the switch into the root switch.

If the switch finds an alternate path to the root bridge, it uses this new alternate path. This new path, and any other alternate paths, are used to send a Root Link Query (RLQ) BPDU. When BackboneFast is enabled, the RLQ BPDUs are sent out as soon as an inferior BPDU is received. This process can enable faster convergence in the event of a backbone link failure.

Figure 26-8 shows an example of a topology with no link failures. Switch A, the root switch, connects directly to Switch B over link L1 and to Switch C over link L2. In this example, because switch B has a lower priority than A but higher than C, switch B becomes the designated bridge for L3. Consequently, the Layer 2 interface on Switch C that connects directly to Switch B must be in the blocking state.

Figure 26-8 BackboneFast Before Indirect Link Failure

Next, assume that L1 fails. Switch A and Switch B, the switches directly connected to this segment, instantly know that the link is down. The blocking interface on Switch C must enter the forwarding state for the network to recover. However, because L1 is not directly connected to Switch C, Switch C does not start sending any BPDUs on L3 under the normal rules of STP until the time defined by the Max Age setting has expired.

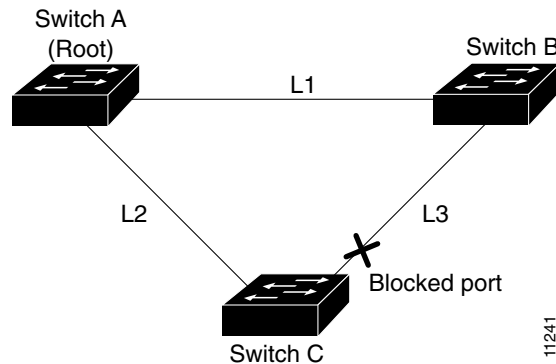
In an STP environment without BackboneFast, if L1 should fail, Switch C cannot detect this failure because it is not connected directly to link L1. However, because Switch B is directly connected to the root switch over L1, Switch B detects the failure and elects itself the root. Switch B begins sending configuration BPDUs to Switch C, listing itself as the root.

The following actions also occur when you use BackboneFast to eliminate the time defined by the Max Age setting (20-second) delay:

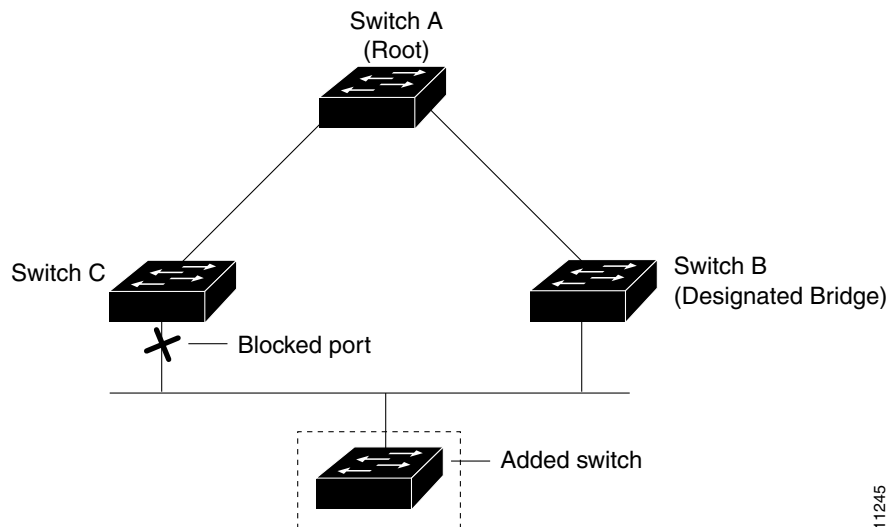
1. When Switch C receives the inferior configuration BPDUs from Switch B, Switch C infers that an indirect failure has occurred.
2. Switch C then sends out an RLQ.
3. Switch A receives the RLQ. Because Switch A is the root bridge, it replies with an RLQ response, listing itself as the root bridge.
4. When Switch C receives the RLQ response on its existing root port, it knows that it still has a stable connection to the root bridge. Because Switch C originated the RLQ request, it does not need to forward the RLQ response on to other switches.
5. BackboneFast allows the blocked port on Switch C to move immediately to the listening state without waiting for the time defined by the Max Age setting for the port to expire.
6. BackboneFast transitions the Layer 2 interface on Switch C to the forwarding state, providing a path from Switch B to Switch A.

This switchover takes approximately 30 seconds, twice the Forward Delay time if the default forward delay time of 15 seconds is set.

Figure 26-9 shows how BackboneFast reconfigures the topology to account for the failure of link L1.

Figure 26-9 BackboneFast after Indirect Link Failure

If a new switch is introduced into a shared-medium topology as shown in [Figure 26-10](#), BackboneFast is not activated, because the inferior BPDUs did not come from the recognized designated bridge (Switch B). The new switch begins sending inferior BPDUs that say it is the root switch. However, the other switches ignore these inferior BPDUs, and the new switch learns that Switch B is the designated bridge to Switch A, the root switch.

Figure 26-10 Adding a Switch in a Shared-Medium Topology

Enabling BackboneFast



Note

For BackboneFast to work, you must enable it on all switches in the network. BackboneFast is supported for use with third-party switches but it is not supported on Token Ring VLANs.

To enable BackboneFast, perform this task:

	Command	Purpose
Step 1	Switch(config)# [no] spanning-tree backbonefast	Enables BackboneFast. Use You can use the no keyword to disable BackboneFast.
Step 2	Switch(config)# end	Exits configuration mode.
Step 3	Switch# show spanning-tree backbonefast	Verifies that BackboneFast is enabled.

This example shows how to enable BackboneFast:

```
Switch(config)# spanning-tree backbonefast
Switch(config)# end
Switch#
```

This example shows how to verify that BackboneFast is enabled:

```
Switch# show spanning-tree backbonefast
BackboneFast is enabled
```

```
BackboneFast statistics
-----
Number of transition via backboneFast (all VLANs) : 0
Number of inferior BPDUs received (all VLANs)    : 0
Number of RLQ request PDUs received (all VLANs)  : 0
Number of RLQ response PDUs received (all VLANs) : 0
Number of RLQ request PDUs sent (all VLANs)      : 0
Number of RLQ response PDUs sent (all VLANs)     : 0
Switch#
```

This example shows how to display a summary of port states:

```
Switch# show spanning-tree summary
Switch is in rapid-pvst mode
Root bridge for:VLAN0001, VLAN1002-VLAN1005
EtherChannel misconfiguration guard is enabled
Extended system ID      is enabled
PortFast Edge BPDUs Guard Default is disabled
Portfast Edge BPDUs Filter Default is disabled
Portfast Default        is disabled
Bridge Assurance         is enabled
Loopguard Default       is disabled
UplinkFast               is disabled
BackboneFast             is disabled
Pathcost method used is short
```

Name	Blocking	Listening	Learning	Forwarding	STP Active
VLAN0001	0	0	0	3	3
VLAN1002	0	0	0	2	2
VLAN1003	0	0	0	2	2
VLAN1004	0	0	0	2	2
VLAN1005	0	0	0	2	2
5 vlans	0	0	0	11	11

```
BackboneFast statistics
-----
Number of transition via backboneFast (all VLANs) :0
Number of inferior BPDUs received (all VLANs)    :0
Number of RLQ request PDUs received (all VLANs)  :0
```

```

Number of RLQ response PDUs received (all VLANs)      :0
Number of RLQ request PDUs sent (all VLANs)           :0
Number of RLQ response PDUs sent (all VLANs)          :0
Switch#

```

This example shows how to display the total lines of the spanning tree state section:

```

Switch# show spanning-tree summary totals
Root bridge for:VLAN0001, VLAN1002-VLAN1005
Extended system ID is disabled
PortFast Edge BPDU Guard Default is disabled
Portfast Edge BPDU Filter Default is enabled
Portfast Default is network
Bridge Assurance is enabled
Loopguard is disabled by default
EtherChannel misconfiguration guard is enabled
UplinkFast is enabled
BackboneFast is enabled
Pathcost method used is short

Name                  Blocking Listening Learning Forwarding STP Active
-----
5 vlans                0          0          0          11          11

BackboneFast statistics
-----
Number of transition via backboneFast (all VLANs)      :0
Number of inferior BPDUs received (all VLANs)         :0
Number of RLQ request PDUs received (all VLANs)       :0
Number of RLQ response PDUs received (all VLANs)      :0
Number of RLQ request PDUs sent (all VLANs)           :0
Number of RLQ response PDUs sent (all VLANs)          :0
Switch#

```




Configuring EtherChannel and Link State Tracking

This chapter describes how to use the command-line interface (CLI) to configure EtherChannel on the Layer 2 or Layer 3 interfaces of the switch. It also provides guidelines, procedures, and configuration examples.

EtherChannel provides automatic recovery for the loss of a link by redistributing the load across the remaining links. If a link fails, EtherChannel redirects traffic from the failed link to the remaining links in the channel without intervention. This chapter also describes how to configure link-state tracking.

This chapter includes the following major sections:

- [About EtherChannel, page 27-1](#)
- [EtherChannel Configuration Guidelines and Restrictions, page 27-6](#)
- [Configuring EtherChannel, page 27-7](#)
- [Displaying EtherChannel to a Virtual Switch System, page 27-20](#)
- [Understanding Link-State Tracking, page 27-23](#)
- [Configuring Link-State Tracking, page 27-26](#)



Note

The commands in the following sections can be used on all Ethernet interfaces of the switch, including the uplink ports on the supervisor engine.



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About EtherChannel

EtherChannel bundles up to eight individual Ethernet links into a single logical link that provides an aggregate bandwidth of up to 800 Mbps (Fast EtherChannel), 8 Gbps (Gigabit EtherChannel), or 80 Gbps (10 Gigabit EtherChannel) between a Catalyst 4500 or 4500X Series Switch and another switch or host.

**Note**

Because some linecards have a maximum bandwidth capacity toward the backplane, they can limit the aggregate bandwidth of an EtherChannel when all the EtherChannel members belong to the same linecard.

The switch supports a maximum of 254 EtherChannels on standalone switches, and 256 EtherChannels in virtual switching system (VSS) mode. You can form an EtherChannel with up to eight compatibly configured Ethernet interfaces across modules on a Catalyst 4500 series switch. All interfaces in each EtherChannel must be the same speed and must be configured as either Layer 2 or Layer 3 interfaces.

**Note**

The network device to which a Catalyst 4500 series switch is connected may impose its own limits on the number of interfaces in an EtherChannel.

If a segment within an EtherChannel fails, traffic previously carried over the failed link switches to the remaining segments within the EtherChannel. When the segment fails, an SNMP trap is sent, identifying the switch, the EtherChannel, and the failed link. Inbound broadcast and multicast packets on one segment in an EtherChannel are blocked from returning on any other segment of the EtherChannel.

**Note**

The port channel link failure switchover for the Catalyst 4500 series switch was measured at 50 milliseconds, which provides SONET-like link failure switchover time.

These subsections describe how EtherChannel works:

- [Port Channel Interfaces, page 27-2](#)
- [Configuring EtherChannels, page 27-2](#)
- [Load Balancing, page 27-6](#)

Port Channel Interfaces

Each EtherChannel has a numbered port channel interface. A configuration applied to the port channel interface affects all physical interfaces assigned to that interface.

**Note**

QoS does not propagate to members. The defaults, QoS cos = 0 and QoS dscp = 0, apply on the port channel. Input or output policies applied on individual interfaces are ignored.

After you configure an EtherChannel, the configuration that you apply to the port channel interface affects the EtherChannel; the configuration that you apply to the physical interfaces affects only the interface where you apply the configuration. To change the parameters of all ports in an EtherChannel, apply configuration commands to the port channel interface (such commands can be STP commands or commands to configure a Layer 2 EtherChannel as a trunk).

Configuring EtherChannels

These subsections describe how EtherChannels are configured:

- [EtherChannel Configuration Overview, page 27-3](#)
- [Manual EtherChannel Configuration, page 27-3](#)

- [PAgP EtherChannel Configuration, page 27-4](#)
- [IEEE 802.3ad LACP EtherChannel Configuration, page 27-5](#)

EtherChannel Configuration Overview

You can configure EtherChannels manually or use the Port Aggregation Control Protocol (PAgP) or the Link Aggregation Control Protocol (LACP) (Cisco IOS Release 12.2(31)SGA and later), to form EtherChannels. The EtherChannel protocols allow ports with similar characteristics to form an EtherChannel through dynamic negotiation with connected network devices. PAgP is a Cisco-proprietary protocol and LACP is defined in IEEE 802.3ad.

PAgP and LACP do not interoperate. Ports configured to use PAgP cannot form EtherChannels with ports configured to use LACP and vice versa.

[Table 27-1](#) lists the user-configurable EtherChannel modes.

Table 27-1 *EtherChannel Modes*

Mode	Description
on	Mode that forces the LAN port to channel unconditionally. In the on mode, a usable EtherChannel exists only when a LAN port group in the on mode is connected to another LAN port group in the on mode. Because ports configured in the on mode do not negotiate, there is no negotiation traffic between the ports.
auto	PAgP mode that places a LAN port into a passive negotiating state in which the port responds to PAgP packets it receives but does not initiate PAgP negotiation.
desirable	PAgP mode that places a LAN port into an active negotiating state in which the port initiates negotiations with other LAN ports by sending PAgP packets.
passive	LACP mode that places a port into a passive negotiating state in which the port responds to LACP packets it receives but does not initiate LACP negotiation.
active	LACP mode that places a port into an active negotiating state in which the port initiates negotiations with other ports by sending LACP packets.

Manual EtherChannel Configuration

Manually configured EtherChannel ports do not exchange EtherChannel protocol packets. A manually configured EtherChannel forms only when you configure all ports compatibly in the EtherChannel.

Auto-LAG

You can automatically create EtherChannels on ports connected to a switch, using auto-LAG. By default, auto-LAG is disabled globally, but is enabled on all port interfaces. To configure auto-LAG on your switch, ensure that you enable auto-LAG globally.

When auto-LAG is enabled globally:

- All port interfaces participate in the creation of auto EtherChannels if the partner port interfaces have EtherChannel configured on them. For more information, see [Table 27-1Supported auto-LAG configuration on actor and partner devices, page 27-4](#).
- Ports that are already part of manual EtherChannels cannot participate in creation of auto EtherChannels.

- When auto-LAG is disabled on a port interface that is already a part of an auto created EtherChannel, the port interface will unbundle from the auto EtherChannel.

The following table shows the supported auto-LAG configurations between the actor and partner devices:

Table 27-1 Supported auto-LAG configuration on actor and partner devices

Actor/Partner	Active	Passive	Auto
Active	Yes	Yes	Yes
Passive	Yes	No	Yes
Auto	Yes	Yes	Yes

On disabling auto-LAG globally, all auto created Etherchannels become manual EtherChannels. You cannot add any configurations in an existing auto created EtherChannel. To add an new configuration, convert the auto-created EtherChannel into a manual EtherChannel using the **port-channel channel-number persistent** command.



Note

Auto-LAG uses the LACP protocol to create an auto EtherChannel. Only one EtherChannel can be automatically created with the unique partner devices.

Auto-LAG Configuration Guidelines

Follow these guidelines when configuring auto-LAG:

- When auto-LAG is enabled globally and on the port interface, and you do not want the port interface to become a member of the auto EtherChannel, disable the auto-LAG on the port interface.
- A port interface does not bundle to an auto EtherChannel when it is already a member of a manual EtherChannel. To allow it to bundle with the auto EtherChannel, first unbundle the manual EtherChannel on the port interface.
- When auto-LAG is enabled and an auto EtherChannel is created, you can create multiple EtherChannels manually with the same partner device. But by default, the port tries to create auto EtherChannel with the partner device.
- Auto-LAG is supported only on Layer 2 EtherChannels. It is not supported on Layer 3 interfaces and Layer 3 EtherChannels.

PAgP EtherChannel Configuration

PAgP supports the automatic creation of EtherChannels by exchanging PAgP packets between LAN ports. PAgP packets are exchanged only between ports in **auto** and **desirable** modes.

The protocol learns the capabilities of LAN port groups dynamically and informs the other LAN ports. Once PAgP identifies correctly matched Ethernet links, it facilitates grouping the links into an EtherChannel. The EtherChannel is then added to the spanning tree as a single bridge port.

Both the **auto** and **desirable** modes allow PAgP to negotiate between LAN ports to determine if they can form an EtherChannel, based on criteria such as port speed and trunking state. Layer 2 EtherChannels also use VLAN numbers.

LAN ports can form an EtherChannel when they are in different PAgP modes if the modes are compatible. For example:

- A LAN port in **desirable** mode can form an EtherChannel successfully with another LAN port that is in **desirable** mode.
- A LAN port in **desirable** mode can form an EtherChannel with another LAN port in **auto** mode.
- A LAN port in **auto** mode cannot form an EtherChannel with another LAN port that is also in **auto** mode because neither port initiates negotiation.

IEEE 802.3ad LACP EtherChannel Configuration

Cisco IOS Release 12.2(31)SGA and later releases support IEEE 802.3ad LACP EtherChannels. LACP supports the automatic creation of EtherChannels by exchanging LACP packets between LAN ports. LACP packets are exchanged only between ports in **passive** and **active** modes.

The protocol learns the capabilities of LAN port groups dynamically and informs the other LAN ports. Once LACP identifies correctly matched Ethernet links, it facilitates grouping the links into an EtherChannel. The EtherChannel is then added to the spanning tree as a single bridge port.

Both the **passive** and **active** modes allow LACP to negotiate between LAN ports to determine if they can form an EtherChannel, based on criteria such as port speed and trunking state. Layer 2 EtherChannels also use VLAN numbers.

LAN ports can form an EtherChannel when they are in different LACP modes as long as the modes are compatible. For example:

- A LAN port in **active** mode can form an EtherChannel successfully with another LAN port that is in **active** mode.
- A LAN port in **active** mode can form an EtherChannel with another LAN port in **passive** mode.
- A LAN port in **passive** mode cannot form an EtherChannel with another LAN port that is also in **passive** mode, because neither port initiates negotiation.

LACP uses the following parameters:

- LACP system priority—You may configure an LACP system priority on each switch running LACP. The system priority can be configured automatically or through the CLI. See the [“Configuring the LACP System Priority and System ID”](#) section on page 27-16. LACP uses the system priority with the switch MAC address to form the system ID and also during negotiation with other systems.



Note The LACP system ID is the combination of the LACP system priority value and the MAC address of the switch.

- LACP port priority—You must configure an LACP port priority on each port configured to use LACP. The port priority can be configured automatically or through the CLI. See the [“Configuring Layer 2 EtherChannels”](#) section on page 27-11. LACP uses the port priority with the port number to form the port identifier.
- LACP administrative key—LACP automatically configures an administrative key value equal to the channel group identification number on each port configured to use LACP. The administrative key defines the ability of a port to aggregate with other ports. A port’s ability to aggregate with other ports is determined by these factors:
 - Port physical characteristics, such as data rate, duplex capability, and point-to-point or shared medium
 - Configuration restrictions that you establish

LACP tries to configure the maximum number of compatible ports in an EtherChannel up to the maximum allowed by the hardware (eight ports). If a port cannot be actively included in a channel, it is not included automatically if a channeled port fails.

You can change the LACP timer rate to modify the duration of the LACP timeout. Use the `lACP rate` command to set the rate at which LACP control packets are sent to an LACP-supported interface. You can change the timeout rate from the default rate (30 seconds) to the fast rate (1 second). This command is supported only on LACP-enabled interfaces. To configure the LACP fast time rate, see *Configuring the LACP Fast Timer Rate*.

ISSU and stateful switchover cannot be guaranteed with LACP fast timers.



Note Standby and “sub-channeling” are not supported in LACP and PagP.

Load Balancing

EtherChannel can balance the traffic load across the links in the channel by reducing part of the binary pattern formed from the addresses or ports in the frame to a numerical value that selects one of the links in the channel. To balance the load, EtherChannel uses MAC addresses, IP addresses, or Layer 4 port numbers, and either the message source or message destination, or both.

Use the option that provides the greatest variety in your configuration. For example, if the traffic on a channel is going only to a single MAC address, using the destination MAC address always chooses the same link in the channel; using source addresses or IP addresses might result in better load balancing.



Note Load balancing can only be configured globally. As a result, all channels (manually configured, PagP, or LACP) use the same load-balancing method.

For additional information on load balancing, see the [“Configuring EtherChannel Load Balancing” section on page 27-18](#).

EtherChannel Configuration Guidelines and Restrictions

If improperly configured, some EtherChannel interfaces are disabled automatically to avoid network loops and other problems. Follow these guidelines and restrictions to avoid configuration problems:

- All Ethernet interfaces on all modules support EtherChannel (maximum of eight interfaces) with no requirement that interfaces be physically contiguous or on the same module.
- Configure all interfaces in an EtherChannel to operate at the same speed and duplex mode.
- Enable all interfaces in an EtherChannel. Disabling an interface in an EtherChannel is treated as a link failure, and its traffic is transferred to one of the remaining interfaces in the EtherChannel.
- An EtherChannel does not form if one of the interfaces is a Switched Port Analyzer (SPAN) destination port.
- For Layer 3 EtherChannels:
 - Assign Layer 3 addresses to the port channel logical interface, not to the physical interfaces in the channel.
- For Layer 2 EtherChannels:

- Assign all interfaces in the EtherChannel to the same VLAN, or configure them as trunks.
- If you configure an EtherChannel from trunk interfaces, verify that the trunking mode and the native VLAN is the same on all the trunks. Interfaces in an EtherChannel with different trunk modes or different native VLANs can have unexpected results.
- An EtherChannel supports the same allowed range of VLANs on all the interfaces in a trunking Layer 2 EtherChannel. If the allowed ranges differ for selected interface differ, they do not form an EtherChannel.
- Interfaces with different Spanning Tree Protocol (STP) port path costs can form an EtherChannel as long they are otherwise compatibly configured. Setting different STP port path costs does not make interfaces incompatible for the formation of an EtherChannel.
- After you configure an EtherChannel, any configuration that you apply to the port channel interface affects the EtherChannel; any configuration that you apply to the physical interfaces affects only the interface you configure.

Storm Control is an exception to this rule. For example, you cannot configure Storm Control on some of the members of an EtherChannel; Storm Control must be configured on all or none of the ports. If you configure Storm Control on only some of the ports, those ports are dropped from the EtherChannel interface (put in suspended state). You should configure Storm Control at the port channel interface level, and not at the physical interface level.

- A physical interface with port security enabled can join a Layer 2 EtherChannel only if port security is also enabled on the EtherChannel; otherwise the command is rejected by the CLI.
- You cannot configure a 802.1X port in an EtherChannel.

Configuring EtherChannel

These sections describe how to configure EtherChannel:

- [Configuring Layer 3 EtherChannels, page 27-7](#)
- [Configuring Layer 2 EtherChannels, page 27-11](#)
- [Configuring LACP Standalone or Independent Mode, page 27-13](#)
- [Configuring LACP Port Channel Min-links, page 27-14](#)
- [Configuring the LACP System Priority and System ID, page 27-16](#)
- [Configuring EtherChannel Load Balancing, page 27-18](#)
- [Removing an Interface from an EtherChannel, page 27-19](#)
- [Removing an EtherChannel, page 27-20](#)

**Note**

Ensure that the interfaces are configured correctly. See the “[EtherChannel Configuration Guidelines and Restrictions](#)” section on page 27-6.

Configuring Layer 3 EtherChannels

To configure Layer 3 EtherChannels, create the port channel logical interface and then put the Ethernet interfaces into the port channel.

These sections describe Layer 3 EtherChannel configuration:

- [Creating Port Channel Logical Interfaces, page 27-8](#)
- [Configuring Physical Interfaces as Layer 3 EtherChannels, page 27-8](#)

Creating Port Channel Logical Interfaces



Note

To move an IP address from a physical interface to an EtherChannel, you must delete the IP address from the physical interface before configuring it on the port channel interface.

To create a port channel interface for a Layer 3 EtherChannel, perform this task:

	Command	Purpose
Step 1	Switch(config)# interface port-channel <i>port_channel_number</i>	Creates the port channel interface. The value for <i>port_channel_number</i> can range from 1 to 254 for a standalone switch, and from 1 to 256 for switches in VSS mode.
Step 2	Switch(config-if)# ip address <i>ip_address mask</i>	Assigns an IP address and subnet mask to the EtherChannel.
Step 3	Switch(config-if)# end	Exits configuration mode.
Step 4	Switch# show running-config interface port-channel <i>port_channel_number</i>	Verifies the configuration.

This example shows how to create port channel interface 1:

```
Switch# configure terminal
Switch(config)# interface port-channel 1
Switch(config-if)# ip address 172.32.52.10 255.255.255.0
Switch(config-if)# end
```

This example shows how to verify the configuration of port channel interface 1:

```
Switch# show running-config interface port-channel 1
Building configuration...

Current configuration:
!
interface Port-channel1
 ip address 172.32.52.10 255.255.255.0
 no ip directed-broadcast
end

Switch#
```

Configuring Physical Interfaces as Layer 3 EtherChannels

To configure physical interfaces as Layer 3 EtherChannels, perform this task for each interface:

	Command	Purpose
Step 1	Switch(config)# interface { fastethernet gigabitethernet tengigabitethernet } <i>slot/port</i>	Selects a physical interface to configure.
Step 2	Switch(config-if)# no switchport	Makes this a Layer 3 routed port.
Step 3	Switch(config-if)# no ip address	Ensures that no IP address is assigned to the physical interface.
Step 4	Switch(config-if)# channel-group <i>port_channel_number</i> mode { active on auto passive desirable }	Configures the interface in a port channel and specifies the PAgP or LACP mode. If you use PAgP, enter the keywords auto or desirable . If you use LACP, enter the keywords active or passive .
Step 5	Switch(config-if)# end	Exits configuration mode.
Step 6	Switch# show running-config interface port-channel <i>port_channel_number</i> Switch# show running-config interface { fastethernet gigabitethernet tengigabitethernet } <i>slot/port</i> Switch# show interfaces { fastethernet gigabitethernet tengigabitethernet } <i>slot/port</i> etherchannel Switch# show etherchannel 1 port-channel	Verifies the configuration.

This example shows how to configure Fast Ethernet interfaces 5/4 and 5/5 into port channel 1 with PAgP mode **desirable**:

```
Switch# configure terminal
Switch(config)# interface range fastethernet 5/4 - 5 (Note: Space is mandatory.)
Switch(config-if)# no switchport
Switch(config-if)# no ip address
Switch(config-if)# channel-group 1 mode desirable
Switch(config-if)# end
```



Note

See the “[Configuring a Range of Interfaces](#)” section on page 9-5 for information about the **range** keyword.

The following two examples show how to verify the configuration of Fast Ethernet interface 5/4:

```
Switch# show running-config interface fastethernet 5/4
Building configuration...
```

```
Current configuration:
!
interface FastEthernet5/4
 no ip address
 no switchport
 no ip directed-broadcast
 channel-group 1 mode desirable
end
```

```
Switch# show interfaces fastethernet 5/4 etherchannel
Port state      = EC-Enbld Up In-Bndl Usr-Config
```

```

Channel group = 1          Mode = Desirable      Gcchange = 0
Port-channel  = Po1       GC   = 0x00010001     Pseudo-port-channel = Po1
Port index    = 0         Load = 0x55

Flags: S - Device is sending Slow hello.  C - Device is in Consistent state.
      A - Device is in Auto mode.         P - Device learns on physical port.
Timers: H - Hello timer is running.       Q - Quit timer is running.
      S - Switching timer is running.     I - Interface timer is running.

Local information:

Port      Flags State   Timers   Hello    Partner  PAgP    Learning  Group
Fa5/4     SC   U6/S7      30s      Interval Count  Priority  Method  Ifindex
Partner's information:

Port      Partner      Partner      Partner      Partner Group
Fa5/4     Name          Device ID    Port          Age  Flags  Cap.
JAB031301 0050.0f10.230c 2/45         1s  SAC   2D

Age of the port in the current state: 00h:54m:52s

Switch#

```

This example shows how to verify the configuration of port channel interface 1 after the interfaces have been configured:

```

Switch# show etherchannel 1 port-channel

Channel-group listing:
-----
Group: 1
-----

Port-channels in the group:
-----
Port-channel: Po1
-----

Age of the Port-channel   = 01h:56m:20s
Logical slot/port        = 10/1          Number of ports = 2
GC                       = 0x00010001    HotStandBy port = null
Port state                = Port-channel L3-Ag Ag-Inuse

Ports in the Port-channel:

Index  Load  Port
-----
1      00    Fa5/6
0      00    Fa5/7

Time since last port bundled: 00h:23m:33s  Fa5/6

Switch#

```

This example shows how to display a one-line summary per channel group:

```

Switch# show etherchannel summary

Flags: D - down          P - bundled in port-channel
      I - stand-alone s - suspended
      H - Hot-standby (LACP only)
      R - Layer3          S - Layer2
      U - in use          f - failed to allocate aggregator

```

```

M - not in use, minimum links not met
u - unsuitable for bundling
w - waiting to be aggregated
d - default port

Number of channel-groups in use: 2
Number of aggregators:          2

Group  Port-channel  Protocol    Ports
-----+-----+-----+-----
1      Po1(SD)          LACP        Gi1/23(H)  Gi1/24(H)
Switch#

```

Configuring Layer 2 EtherChannels

To configure Layer 2 EtherChannels, configure the Ethernet interfaces with the **channel-group** command. This operation creates the port channel logical interface.



Note

Cisco IOS software creates port channel interfaces for Layer 2 EtherChannels when you configure Layer 2 Ethernet interfaces with the **channel-group** command.

To configure Layer 2 Ethernet interfaces as Layer 2 EtherChannels, perform this task for each interface:

	Command	Purpose
Step 1	Switch(config)# interface { fastethernet gigabitethernet tengigabitethernet } <i>slot/port</i>	Selects a physical interface to configure.
Step 2	Switch(config-if)# channel-group <i>port_channel_number</i> mode { active on auto passive desirable }	Configures the interface in a port channel and specifies the PAGP or LACP mode. If you use PAGP, enter the keywords auto or desirable . If you use LACP, enter the keywords active or passive .
Step 3	Switch(config-if)# end	Exits configuration mode.
Step 4	Switch# show running-config interface { fastethernet gigabitethernet } <i>slot/port</i> Switch# show interface { fastethernet gigabitethernet tengigabitethernet } <i>slot/port</i> etherchannel	Verifies the configuration.

This example shows how to configure Fast Ethernet interfaces 5/6 and 5/7 into port channel 2 with PAGP mode **desirable**:

```

Switch# configure terminal
Switch(config)# interface range fastethernet 5/6 - 7 (Note: Space is mandatory.)
Switch(config-if-range)# channel-group 2 mode desirable
Switch(config-if-range)# end
Switch# end

```



Note

See the [“Configuring a Range of Interfaces”](#) section on page 9-5 for information about the **range** keyword.

This example shows how to verify the configuration of port channel interface 2:

```
Switch# show running-config interface port-channel 2
Building configuration...
```

```
Current configuration:
!
interface Port-channel2
 switchport access vlan 10
 switchport mode access
end
```

```
Switch#
```

The following two examples show how to verify the configuration of Fast Ethernet interface 5/6:

```
Switch# show running-config interface fastethernet 5/6
Building configuration...
```

```
Current configuration:
!
interface FastEthernet5/6
 switchport access vlan 10
 switchport mode access
 channel-group 2 mode desirable
end
```

```
Switch# show interfaces fastethernet 5/6 etherchannel
Port state      = EC-Enbld Up In-Bndl Usr-Config
Channel group = 1          Mode = Desirable      Gcchange = 0
Port-channel   = Po1       GC      = 0x00010001
Port indx      = 0         Load = 0x55
```

```
Flags: S - Device is sending Slow hello.  C - Device is in Consistent state.
       A - Device is in Auto mode.        P - Device learns on physical port.
       d - PAgP is down.
```

```
Timers: H - Hello timer is running.      Q - Quit timer is running.
        S - Switching timer is running.  I - Interface timer is running.
```

Local information:

Port	Flags	State	Timers	Hello Interval	Partner Count	PAgP Priority	Learning Method	Group Ifindex
Fa5/6	SC	U6/S7		30s	1	128	Any	56

Partner's information:

Port	Partner Name	Partner Device ID	Partner Port	Partner Age	Partner Flags	Partner Group Cap.
Fa5/6	JAB031301	0050.0f10.230c	2/47	18s	SAC	2F

Age of the port in the current state: 00h:10m:57s

This example shows how to verify the configuration of port channel interface 2 after the interfaces have been configured:

```
Switch# show etherchannel 2 port-channel
Port-channels in the group:
-----
```

```
Port-channel: Po2
-----
```

```
Age of the Port-channel   = 00h:23m:33s
Logical slot/port        = 10/2          Number of ports in agport = 2
GC                        = 0x00020001    HotStandBy port = null
```



```

Port state          = Port-channel Ag-Inuse

Ports in the Port-channel:

Index    Load    Port
-----
    1      00      Fa5/6
    0      00      Fa5/7

Time since last port bundled:    00h:23m:33s    Fa5/6

Switch#

```

Configuring LACP Standalone or Independent Mode

This feature is particularly relevant when a port (A) in a Layer 2 LACP EtherChannel is connected to an unresponsive port (B) on the peer. When LACP standalone is disabled on the EtherChannel, all traffic arriving on A is blocked (the default behavior on a switch). In some scenarios, you might want to allow management traffic on such ports. You can do this by enabling LACP standalone (or independent) mode.



Note

This **port-channel standalone-disable** command only applies to Layer 2 EtherChannels



Note

LACP Standalone Disable is enabled by default.

To configure the LACP Standalone or Independent mode, perform this task:

	Command	Purpose
Step 1	Switch(config)# no port-channel standalone-disable	Enables the LACP standalone or independent mode.
	Switch(config)# port-channel standalone-disable	Reverts to the default.
Step 2	Switch(config-if)# end	Exits configuration mode.
Step 3	Switch# show running configuration {fastethernet gigabitethernet} slot/port port-channel port_channel_number	Verifies the configuration.

This example shows how to configure the LACP Standalone mode:

```

Switch# configure terminal
Switch(config)# interface port-channel 1
Switch(config-if)# switchport
Switch(config-if)# exit
Switch(config)# int gi3/1
Switch(config-if)# channel-group 1 mode active
Switch(config-if)# exit
Switch(config)# interface port-channel 1
Switch(config-if)# no port-channel standalone-disable
Ports of Po12 already in suspend (S) mode require a shut/no shut.
Switch(config-if)# end

```

This example shows how to verify the configuration of port channel interface 1:

```

Switch# show running-config interface port-channel 1

```

```
Building configuration...

Current configuration:
!
interface Port-channel1
  switchport
  no port-channel standalone-disable
end

Switch#
```

This example shows how to verify the state of port channel interface 1:

```
Switch# show etherchannel 1 port-channel
      Port-channels in the group:
      -----
Port-channel: Po13      (Primary Aggregator)
-----
Age of the Port-channel   = 0d:00h:07m:57s
Logical slot/port        = 11/13           Number of ports = 0
Port state                = Port-channel Ag-Not-Inuse
Protocol                  = LACP
Port security             = Disabled
Standalone                = Enabled (independent mode)
Switch#
```

Configuring LACP Port Channel Min-links

Beginning in Cisco IOS Release 15.2(4)E, and Cisco IOS XE Release 3.8.0E, you can specify the minimum number of active ports that must be in the link-up state and bundled in an EtherChannel for the port channel interface to transition to the link-up state. Using EtherChannel min-links, you can prevent low-bandwidth LACP EtherChannels from becoming active. Port channel min-links also cause LACP EtherChannels to become inactive if they have too few active member ports to supply the required minimum bandwidth.



Note

For a Multichassis EtherChannel (MEC), the LACP *min-links* command argument defines the minimum number of physical links in each chassis for the MEC to be operational.

To configure the minimum number of links that are required for a port channel, perform the following task:

	Command	Purpose
Step 1	Switch(config)# interface port-channel <i>channel-number</i>	Enters interface configuration mode for a port-channel. The valid range for <i>channel-number</i> is 1 to 256.

	Command	Purpose
Step 2	Switch(config-if)# port-channel min-links <i>min-links-number</i>	Specifies the minimum number of member ports that must be in the link-up state and bundled in the EtherChannel for the port channel interface to transition to the link-up state. The valid range for <i>min-links-number</i> is 2 to 8. For a switch in VSS mode, when setting up min-links, ensure that the port-channel consists of the same number of min-links on the active switch and on the standby switch. For example, if you set min-links to 2 for a port-channel, ensure that there are 2 links, each, available on the active and the standby switch.
Step 3	Switch# end	Returns to privileged EXEC mode

This example shows how to configure LACP port channel min-links:

```
Switch# configure terminal
Switch(config)# interface port-channel 25
switch(config-if)# port-channel min-links 3
Switch# show etherchannel 25 summary
switch# end
```

For a switch in VSS mode, when the minimum links requirement on a port channel is not met, the local ports in the EtherChannel are assigned m state, and all traffic is redirected out through the virtual switch link, via the member ports of the port channel on the peer switch, for example:

```
Switch# show etherchannel 25 summary
Flags: D - down P - bundled in port-channel
I - stand-alone s - suspended
H - Hot-standby (LACP only)
R - Layer3 S - Layer2
U - in use N - not in use, no aggregation
f - failed to allocate aggregator
M - not in use, no aggregation due to minimum links not met
m- not in use, port not aggregated due to minimum links not met
u - unsuitable for bundling
w - waiting to be aggregated
d - default port
Number of channel-groups in use: 125
Number of aggregators: 125
Group Port-channel Protocol Ports
-----+-----+-----+-----+-----+-----
25 Po25(RU) LACP Gi1/3/1(D) Gi1/3/2(m) Gi2/2/25(P) Gi2/2/26(P)
```

When the minimum links requirement is not met in standalone switches, the port channel is flagged and assigned the SM/SN or RM/RN state, for example:

```
Switch# show etherchannel 25 summary
Flags: D - down P - bundled in port-channel
I - stand-alone s - suspended
H - Hot-standby (LACP only)
R - Layer3 S - Layer2
U - in use N- not in use, no aggregation
f - failed to allocate aggregator
M - not in use, no aggregation due to minimum links not met
m- not in use, port not aggregated due to minimum links not met
u - unsuitable for bundling
```

```

w - waiting to be aggregated
d - default port
Number of channel-groups in use: 125
Number of aggregators: 125
Group Port-channel Protocol Ports
-----+-----+-----+-----+-----+-----
25 Po25(RM) LACP Gi1/3/1(D) Gi1/3/2(D) Gi2/2/25(D) Gi2/2/26(w)

```

Configuring the LACP System Priority and System ID

The LACP system ID is the LACP system priority value combined with the MAC address of the switch. To configure the LACP system priority and system ID, perform this task:

	Command	Purpose
Step 1	Switch(config)# lACP system-priority <i>priority_value</i>	(Optional for LACP) Sets the LACP system priority and system ID. Valid values are 1 through 65535. Higher numbers have lower priority. The default is 32768.
	Switch(config)# no system port-priority	Reverts to the default.
Step 2	Switch(config)# end	Exits configuration mode.
Step 3	Switch# show lACP sys-id	Verifies the configuration.

This example shows how to configure the LACP system priority:

```

Switch# configure terminal
Switch(config)# lACP system-priority 23456
Switch(config)# end
Switch# show module

Mod  Ports Card Type                               Model                Serial No.
-----+-----+-----+-----+-----+-----
1      2  1000BaseX (GBIC) Supervisor(active)    WS-X4014              JAB063808YZ
2     48  10/100BaseTX (RJ45)                    WS-X4148-RJ           JAB0447072W
3     48  10/100BaseTX (RJ45)V                    WS-X4148-RJ45V        JAE061704J6
4     48  10/100BaseTX (RJ45)V                    WS-X4148-RJ45V        JAE061704ML

M MAC addresses                               Hw  Fw      Sw              Status
-----+-----+-----+-----+-----+-----
1 0005.9a39.7a80 to 0005.9a39.7a81 2.1 12.1(12r)EW 12.1(13)EW(0.26) Ok
2 0002.fd80.f530 to 0002.fd80.f55f 0.1                               Ok
3 0009.7c45.67c0 to 0009.7c45.67ef 1.6                               Ok
4 0009.7c45.4a80 to 0009.7c45.4aaf 1.6                               Ok

```

This example shows how to verify the configuration:

```

Switch# show lACP sys-id
23456,0050.3e8d.6400
Switch#

```

The system priority is displayed first, followed by the MAC address of the switch.

Configuring LACP Fast Rate Timer

You can change the LACP timer rate to modify the duration of the LACP timeout. Use the **lacp rate** command to set the rate at which LACP control packets are received by an LACP-supported interface. You can change the timeout rate from the default rate (30 seconds) to the fast rate (1 second). This command is supported only on LACP-enabled interfaces.

	Command	Purpose
Step 4	Switch# configure terminal	Enters configuration mode.
Step 5	Switch(config)# interface { fastethernet gigabitethernet tengigabitethernet } <i>slot/port</i>	Selects the interface to configure.
Step 6	Switch(config-if)# lacp rate { normal fast }	Configures the rate at which LACP control packets are received by an LACP-supported interface. To reset the timeout rate to its default, use the no lacp rate command.

This example shows how to configure the LACP rate:

```
Switch# configure terminal
Switch(config)# interface gigabitEthernet 2/1
Switch(config-if)# lacp rate fast
Switch(config-if)# exit
```

The **show lacp internal** command displays similar output:

```
Flags: S - Device is requesting Slow LACPDU
F - Device is requesting Fast LACPDU
A - Device is in Active mode P - Device is in Passive mode
Channel group 25
LACP port Admin Oper Port Port
Port Flags State Priority Key Key Number State
Tel/49 FA bndl 32768 0x19 0x19 0x32 0x3F
Tel/50 FA bndl 32768 0x19 0x19 0x33 0x3F
Tel/51 FA bndl 32768 0x19 0x19 0x34 0x3F
Tel/52 FA bndl 32768 0x19 0x19 0x35 0x3F
```

The **show lacp counters** command displays similar output

```
Switch# show lacp counters
LACPDU Marker Marker Response LACPDU
Port Sent Recv Sent Recv Sent Recv Pkts Err
-----
Channel group: 24
Tel/1/27 2 2 0 0 0 0 0
Te2/1/25 2 2 0 0 0 0 0
```

Configuring Auto-LAG Globally



Note

By default, auto-LAG is disabled globally, on your device.

To configure auto-LAG globally, perform this task:

	Command	Purpose
Step 1	Switch(config)# port-channel auto	Configures auto-LAG globally on the device.
Step 2	Switch(config)# end	Exits configuration mode.
Step 3	Switch# show etherchannel auto	Displays the EtherChannel created automatically.

Configuring Auto-LAG on a Port Interface

To configure auto-LAG globally, perform this task:

	Command	Purpose
Step 1	Switch(config)# interface <i>interface-id</i>	Specifies the port interface to be enabled for auto-LAG, and enters interface configuration mode.
Step 2	Switch(config-if)# channel-group auto	(Optional) Enables auto-LAG feature on individual port interface. By default, the auto-LAG feature is enabled on the port.
Step 3	Switch(config)# end	Exits configuration mode.
Step 4	Switch# show etherchannel auto	Displays the EtherChannel created automatically.

Configuring Persistence with Auto-LAG

To convert the automatically created EtherChannel back to a manual one, perform this task:

	Command	Purpose
Step 1	Switch# port-channel <i>channel-number</i> persistent	Specifies the port interface to be enabled for auto-LAG, and enters interface configuration mode.
Step 2	Switch(config-if)# channel-group auto	Converts the auto created EtherChannel into a manual one and allows you to add configuration on the EtherChannel.
Step 3	Switch(config)# end	Exits configuration mode.
Step 4	Switch# show etherchannel summary	Displays EtherChannel information.

Configuring EtherChannel Load Balancing



Note

Load balancing can only be configured globally. As a result, all channels (manually configured, PagP, or LACP) use the same load-balancing method.

To configure EtherChannel load balancing, perform this task:

	Command	Purpose
Step 1	Switch(config)# [no] port-channel load-balance { src-mac dst-mac src-dst-mac src-ip dst-ip src-dst-ip src-port dst-port src-dst-port }	Configures EtherChannel load balancing. Use the no keyword to return EtherChannel load balancing to the default configuration.
Step 2	Switch(config)# end	Exits configuration mode.
Step 3	Switch# show etherchannel load-balance	Verifies the configuration.

The load-balancing keywords indicate these values:

- **src-mac**—Source MAC addresses
- **dst-mac**—Destination MAC addresses
- **src-dst-mac**—Source and destination MAC addresses
- **src-ip**—Source IP addresses
- **dst-ip**—Destination IP addresses
- **src-dst-ip**—Source and destination IP addresses (Default)
- **src-port**—Source Layer 4 port
- **dst-port**—Destination Layer 4 port
- **src-dst-port**—Source and destination Layer 4 port

This example shows how to configure EtherChannel to use source and destination IP addresses:

```
Switch# configure terminal
Switch(config)# port-channel load-balance src-dst-ip
Switch(config)# end
Switch#
```

This example shows how to verify the configuration:

```
Switch# show etherchannel load-balance
EtherChannel Load-Balancing Configuration:
    src-dst-ip

EtherChannel Load-Balancing Addresses Used Per-Protocol:
Non-IP: Source XOR Destination MAC address
IPv4: Source XOR Destination IP address
IPv6: Source XOR Destination IP address
Switch#
```

Removing an Interface from an EtherChannel

To remove an Ethernet interface from an EtherChannel, perform this task:

	Command	Purpose
Step 1	Switch(config)# interface { fastethernet gigabitethernet tengigabitethernet } <i>slot/port</i>	Selects a physical interface to configure.
Step 2	Switch(config-if)# no channel-group	Removes the interface from the port channel interface.

	Command	Purpose
Step 3	Switch(config-if)# end	Exits configuration mode.
Step 4	Switch# show running-config interface {fastethernet gigabitethernet tengigabitethernet} slot/port Switch# show interface {fastethernet gigabitethernet tengigabitethernet} slot/port etherchannel	Verifies the configuration.

This example shows how to remove Fast Ethernet interfaces 5/4 and 5/5 from port channel 1:

```
Switch# configure terminal
Switch(config)# interface range fastethernet 5/4 - 5 (Note: Space is mandatory.)
Switch(config-if)# no channel-group 1
Switch(config-if)# end
```

Removing an EtherChannel

If you remove an EtherChannel, the member ports are shut down and removed from the channel group.



Note

If you want to change an EtherChannel from Layer 2 to Layer 3, or Layer 3 to Layer 2, you must remove the EtherChannel and recreate it in the desired configuration.

To remove an EtherChannel, perform this task:

	Command	Purpose
Step 1	Switch(config)# no interface port-channel port_channel_number	Removes the port channel interface.
Step 2	Switch(config)# end	Exits configuration mode.
Step 3	Switch# show etherchannel summary	Verifies the configuration.

This example shows how to remove port channel 1:

```
Switch# configure terminal
Switch(config)# no interface port-channel 1
Switch(config)# end
```

Displaying EtherChannel to a Virtual Switch System

Catalyst 4500 series switches support enhanced PAgP. If a Catalyst 4500 series switch is connected to a Catalyst 6500 series Virtual Switch System (VSS) by using a PAgP EtherChannel, the Catalyst 4500 series switch automatically serve as a VSS client, using enhanced PAgP on this EtherChannel for dual-active detection. This VSS client feature has no impact on the performance of Catalyst 4500 series switch and does not require any user configuration.

This section includes these topics:

- [Understanding VSS Client, page 27-21](#)
- [Displaying EtherChannel Links to VSS, page 27-23](#)

Understanding VSS Client

This section describes these topics:

- [Virtual Switch System, page 27-21](#)
- [Dual-Active Scenarios, page 27-21](#)
- [Dual-Active Detection Using Enhanced PAgP, page 27-21](#)

Virtual Switch System

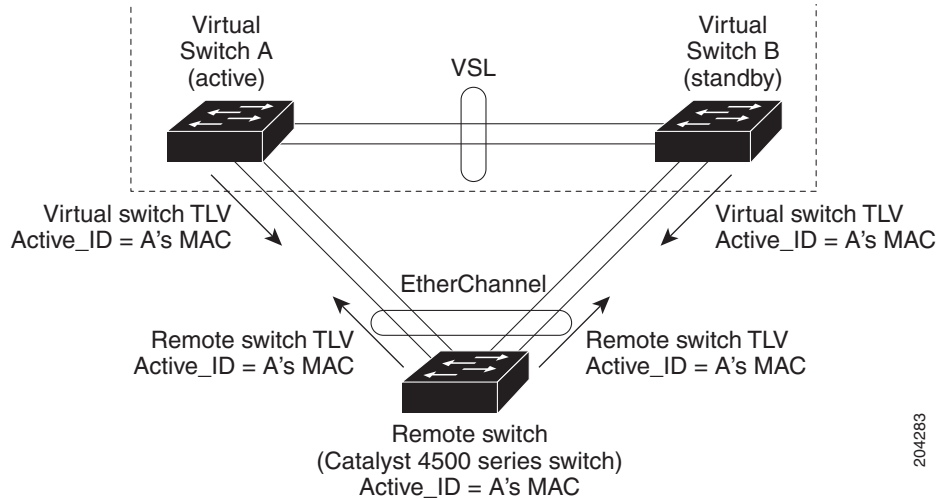
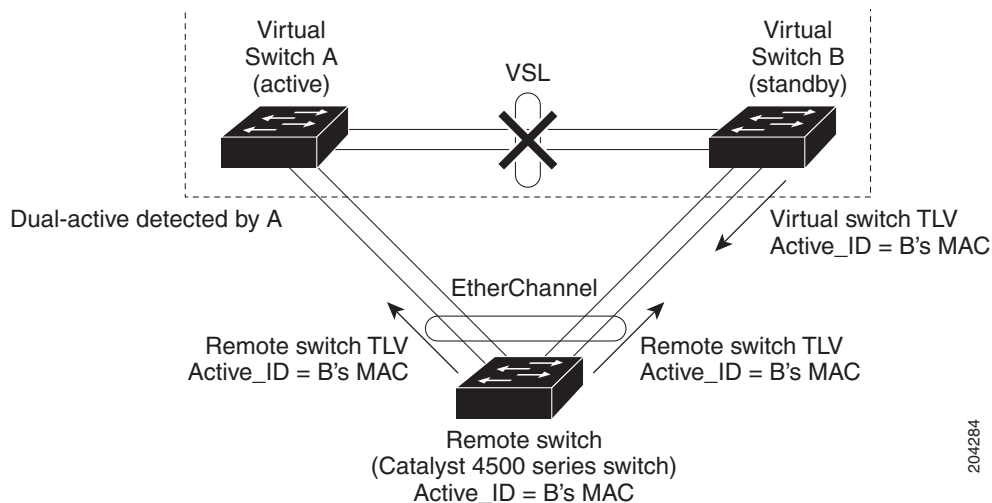
The Cisco Catalyst 6500 Series VSS 1440 allows for the combination of two Cisco Catalyst 6500 Series switches into a single, logical network entity from the network control-plane and management perspectives. Within the Cisco VSS, one chassis is designated as the active virtual switch, acting as the single management point of the entire system, and the other is designated as the standby virtual switch. These two chassis are bound together by a special link, called Virtual Switch Link (VSL), which carries the internal signaling and control information between them.

Dual-Active Scenarios

One of the failure scenarios in a VSS is called *dual-active*, which occurs when the VSL fails completely. Neither virtual switch knows of the other's status. From the perspective of the active virtual switch, the standby chassis is lost. The standby virtual switch also views the active chassis as failed and transitions to active state by using an SSO switchover. Two active virtual switches exist in the network with identical configurations, causing duplicate IP addresses and bridge identifiers. This scenario has adverse effects on the network topology and traffic if it persists.

Dual-Active Detection Using Enhanced PAgP

One method for detecting a dual-active scenario is based on enhanced PAgP (PAgP+). Specifically, the VSS sends regularly scheduled PAgP messages with Type-Length-Values (TLVs) containing the ID of the current active virtual switch ([Figure 27-1](#)). When the VSL fails completely, the standby virtual switch immediately sends asynchronous PAgP messages with TLVs containing its own ID on all port channels enabled for enhanced PAgP dual-active detection ([Figure 27-2](#)). The remote switch (the VSS client) connected to both VSS components by using EtherChannel links, compares every received active ID with its stored active ID. If they match, the remote switch sends TLVs containing its stored active ID back to the VSS in its regularly scheduled PAgP messages. If they do not match, the remote switch stores the new active ID and immediately transmits asynchronous PAgP messages with TLVs containing the new active ID. Upon receiving the new active ID from the remote switch, the original active virtual switch detects the dual-active scenario and takes appropriate actions.

Figure 27-1 Enhanced PAgP in VSS Normal Operation**Figure 27-2** Enhanced PAgP in VSS Dual-active Scenario

As a remote switch, the Catalyst 4500 series switch supports stateful VSS client. In particular, the ID of the current active virtual switch is synchronized from the active supervisor engine to the redundant supervisor engine of the Catalyst 4500 series switch. This ensures that dual-active detection is not disrupted even when the active supervisor engine switches over to the redundant supervisor engine.

Displaying EtherChannel Links to VSS

To display the dual-active detection capability of a configured PAgP port channel, enter the **show pagp port_channel_number dual-active** command.

The command provides the following information:

- A switch uses enhanced PAgP for dual-active detection.
You should always see Yes after PAgP dual-active detection enabled on a Catalyst 4500 switch.
- The configured PAgP EtherChannel is connected to a Catalyst 6500 switch VSS.
You see N/A below Partner Version if this EtherChannel is *not* connected to a VSS. Otherwise, you see the version of enhanced PAgP dual-active detection implemented in the VSS.
- This switch is capable of detecting dual-active scenarios in the connected VSS.
You see Yes below Dual-Active Detect Capable if and only if the configured EtherChannel is connected to a Catalyst 6500 series VSS that uses the same version of enhanced PAgP dual-active detection.



Note

You can also see the name of the neighboring switch (Partner Name) and the ports to which this EtherChannel is connected (Partner Port).

If a Catalyst 4500 switch is connected to a Catalyst 6500 series VSS with the same version of enhanced PAgP dual-active detection, the switch can detect a dual-active scenario:

```
Switch# show pagp 1 dual-active
```

```
PAgP dual-active detection enabled: Yes
```

```
PAgP dual-active version: 1.1
```

```
Channel group 1
```

Port	Dual-Active Detect Capable	Partner Name	Partner Port	Partner Version
Gi6/5	Yes	VSS	Gi1/8/1	1.1
Gi6/6	Yes	VSS	Gi2/8/1	1.1

If a Catalyst 4500 switch is not connected to a Catalyst 6500 series VSS, the switch cannot detect a dual-active scenario:

```
Switch# show pagp 1 dual-active
```

```
PAgP dual-active detection enabled: Yes
```

```
PAgP dual-active version: 1.1
```

```
Channel group 1
```

Port	Dual-Active Detect Capable	Partner Name	Partner Port	Partner Version
Gi6/5	No	Switch	Fa6/5	N/A
Gi6/6	No	Switch	Fa6/6	N/A

Understanding Link-State Tracking

Link-state tracking, also known as trunk failover, is a feature that binds the link state of multiple interfaces. For example, link-state tracking provides redundancy in the network when used with server NIC adapter teaming. When server network adapters are configured in a primary or secondary relationship known as teaming, if the link is lost on the primary interface, connectivity is transparently changed to the secondary interface.

Figure 27-3 on page 27-25 shows a network configured with link-state tracking. To enable link-state tracking, create a link-state group, and specify the interfaces that are assigned to the link-state group. An interface can be an aggregation of ports (an EtherChannel), a single physical port in access or trunk mode, or a routed port. In a link-state group, these interfaces are bundled together. The downstream interfaces are bound to the upstream interfaces. Interfaces connected to servers are referred to as downstream interfaces, and interfaces connected to distribution switches and network devices are referred to as upstream interfaces.

The configuration in Figure 27-3 ensures that the network traffic flow is balanced as follows:

- For links to switches and other network devices
 - Server 1 and server 2 use switch A for primary links and switch B for secondary links.
 - Server 3 and server 4 use switch B for primary links and switch A for secondary links.
- Link-state group 1 on switch A
 - Switch A provides primary links to server 1 and server 2 through link-state group 1. Port 1 is connected to server 1, and port 2 is connected to server 2. Port 1 and port 2 are the downstream interfaces in link-state group 1.
 - Port 5 and port 6 are connected to distribution switch 1 through link-state group 1. Port 5 and port 6 are the upstream interfaces in link-state group 1.
- Link-state group 2 on switch A
 - Switch A provides secondary links to server 3 and server 4 through link-state group 2. Port 3 is connected to server 3, and port 4 is connected to server 4. Port 3 and port 4 are the downstream interfaces in link-state group 2.
 - Port 7 and port 8 are connected to distribution switch 2 through link-state group 2. Port 7 and port 8 are the upstream interfaces in link-state group 2.
- Link-state group 2 on switch B
 - Switch B provides primary links to server 3 and server 4 through link-state group 2. Port 3 is connected to server 3, and port 4 is connected to server 4. Port 3 and port 4 are the downstream interfaces in link-state group 2.
 - Port 5 and port 6 are connected to distribution switch 2 through link-state group 2. Port 5 and port 6 are the upstream interfaces in link-state group 2.
- Link-state group 1 on switch B
 - Switch B provides secondary links to server 1 and server 2 through link-state group 1. Port 1 is connected to server 1, and port 2 is connected to server 2. Port 1 and port 2 are the downstream interfaces in link-state group 1.
 - Port 7 and port 8 are connected to distribution switch 1 through link-state group 1. Port 7 and port 8 are the upstream interfaces in link-state group 1.

In a link-state group, the upstream ports can become unavailable or lose connectivity because the distribution switch or router fails, the cables are disconnected, or the link is lost. These are the interactions between the downstream and upstream interfaces when link-state tracking is enabled:

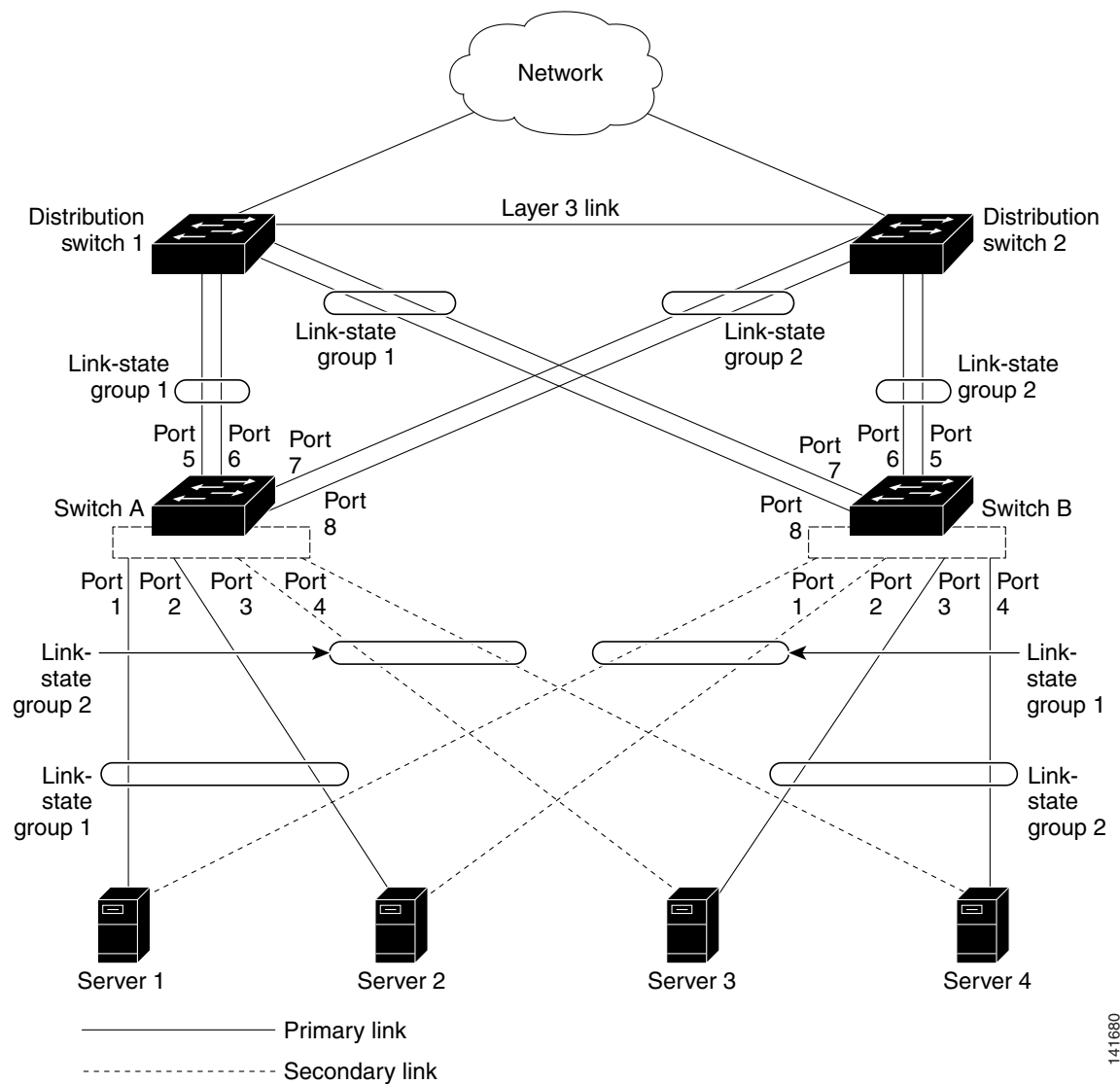
- If any of the upstream interfaces are in the link-up state, the downstream interfaces can change to or remain in the link-up state.
- If all of the upstream interfaces become unavailable, link-state tracking automatically puts the downstream interfaces in the error-disabled state. Connectivity to and from the servers is automatically changed from the primary server interface to the secondary server interface.

As an example of a connectivity change from link-state group 1 to link-state group 2 on switch A, see [Figure 27-3 on page 27-25](#). If the upstream link for port 6 is lost, the link states of downstream ports 1 and 2 do not change. However, if the link for upstream port 5 is also lost, the link state of the downstream ports changes to the link-down state. Connectivity to server 1 and server 2 is then changed from link-state group 1 to link-state group 2. The downstream ports 3 and 4 do not change state because they are in link-group 2.

- If the link-state group is configured, link-state tracking is disabled, and the upstream interfaces lose connectivity, the link states of the downstream interfaces remain unchanged. The server does not recognize that upstream connectivity has been lost and does not failover to the secondary interface.

You can recover a downstream interface link-down condition by removing the failed downstream port from the link-state group. To recover multiple downstream interfaces, disable the link-state group.

Figure 27-3 Typical Link-State Tracking Configuration



141680

Configuring Link-State Tracking

These sections describe how to configure link-state tracking ports:

- [Default Link-State Tracking Configuration, page 27-26](#)
- [Link-State Tracking Configuration Guidelines, page 27-26](#)
- [Configuring Link-State Tracking, page 27-26](#)
- [Displaying Link-State Tracking Status, page 27-27](#)

Default Link-State Tracking Configuration

No link-state groups are defined, and link-state tracking is not enabled for any group.

Link-State Tracking Configuration Guidelines

Follow these guidelines to avoid configuration problems:

- An interface that is defined as an upstream interface cannot also be defined as a downstream interface in the same or different link-state group. The reverse is also true.
- We recommend that you add the upstream interfaces to the link state group before adding the downstream interfaces. it is because when a downstream interface is added to a link state group without an upstream interface, the downstream interface is put in error-disabled state until an upstream interfaces is added to the group.
- An interface cannot be a member of more than one link-state group.
- You can configure up to 20 link-state groups per switch.
- If a SPAN destination port is configured as a downstream interface, it is error disabled when all upstream interfaces in its group are down. When an upstream interface is configured as a SPAN destination port, it is considered as a link down event on the interface.

Configuring Link-State Tracking

To configure a link-state group and to assign an interface to a group, perform this task beginning in privileged EXEC mode:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# link state track <i>number</i>	Creates a link-state group, and enables link-state tracking. The group number range is 1 to 20; the default is 1.
Step 3	Switch(config)# interface <i>interface-id</i>	Specifies a physical interface or range of interfaces to configure, and enters interface configuration mode. Valid interfaces include switch ports in access mode or trunk mode (IEEE 802.1q), routed ports, or multiple ports bundled into an EtherChannel interface (static or LACP), in trunk mode.

	Command	Purpose
Step 4	Switch(config-if)# link state group <i>[number]</i> { upstream downstream }	Specifies a link-state group, and configures the interface as either an upstream or downstream interface in the group. The group number range is from 1 to 20; the default is 1.
Step 5	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 6	Switch# show running-config	Verifies your entries.
Step 7	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to create a link-state group and configure the interfaces:

```
Switch# configure terminal
Switch(config)# link state track 1
Switch(config)# interface gigabitethernet3/1
Switch(config-if)# link state group 1 upstream
Switch(config-if)# interface gigabitethernet3/3
Switch(config-if)# link state group 1 upstream
Switch(config-if)# interface gigabitethernet3/5
Switch(config-if)# link state group 1 downstream
Switch(config-if)# interface gigabitethernet3/7
Switch(config-if)# link state group 1 downstream
Switch(config-if)# end
```

To disable a link-state group, use the **no link state track *number*** global configuration command.

Displaying Link-State Tracking Status

To display the link-state group information, enter the **show link state group** command. Enter this command without keywords to display information about all link-state groups.

Enter the group number to display information specific to the group. Enter the **detail** keyword to display detailed information about the group.

it is an example of output from the **show link state group 1** command:

```
Switch> show link state group 1
Link State Group: 1 Status: Enabled, Down
```

it is an example of output from the **show link state group detail** command:

```
Switch> show link state group detail
(Up):Interface up (Dwn):Interface Down (Dis):Interface disabled
Link State Group: 1 Status: Enabled, Down
Upstream Interfaces : Gi3/5(Dwn) Gi3/6(Dwn)
Downstream Interfaces : Gi3/1(Dis) Gi3/2(Dis) Gi3/3(Dis) Gi3/4(Dis)
Link State Group: 2 Status: Enabled, Down
Upstream Interfaces : Gi3/15(Dwn) Gi3/16(Dwn) Gi3/17(Dwn)
Downstream Interfaces : Gi3/11(Dis) Gi3/12(Dis) Gi3/13(Dis) Gi3/14(Dis)
(Up):Interface up (Dwn):Interface Down (Dis):Interface disabled
```




Configuring IGMP Snooping and Filtering, and MVR

This chapter describes how to configure Internet Group Management Protocol (IGMP) snooping on the switch, including an application of local IGMP snooping, Multicast VLAN Registration (MVR). It also includes procedures for controlling multicast group membership by using IGMP filtering.

This chapter consists of the following major sections:

- [About IGMP Snooping, page 28-1](#)
- [Configuring IGMP Snooping, page 28-5](#)
- [Displaying IGMP Snooping Information, page 28-14](#)
- [Understanding Multicast VLAN Registration, page 28-20](#)
- [Configuring MVR, page 28-23](#)
- [Displaying MVR Information, page 28-29](#)
- [Configuring IGMP Filtering, page 28-30](#)
- [Displaying IGMP Filtering Configuration, page 28-34](#)



Note

To support Cisco Group Management Protocol (CGMP) client devices, configure the switch as a CGMP server. For more information, see the Cisco IOS 15.0M configuration guides at this location: http://www.cisco.com/en/US/products/ps10591/products_installation_and_configuration_guides_list.html

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About IGMP Snooping

This section includes the following subsections:

- [Immediate-Leave Processing, page 28-3](#)
- [IGMP Configurable-Leave Timer, page 28-4](#)

- [IGMP Snooping Querier, page 28-4](#)
- [Explicit Host Tracking, page 28-4](#)

**Note**

Quality of service does not apply to IGMP packets.

IGMP snooping allows a switch to snoop or capture information from IGMP packets transmitted between hosts and a router. Based on this information, a switch adds or deletes multicast addresses from its address table, which enables (or disables) multicast traffic from flowing to individual host ports. IGMP snooping supports all versions of IGMP: IGMPv1, IGMPv2, and IGMPv3.

In contrast to IGMPv1 and IGMPv2, IGMPv3 snooping provides immediate-leave processing by default. It provides explicit host tracking (EHT) and allows network administrators to deploy SSM functionality on Layer 2 devices that support IGMPv3. See the [“Explicit Host Tracking” section on page 28-4](#). In subnets where IGMP is configured, IGMP snooping manages multicast traffic at Layer 2. You can configure interfaces to dynamically forward multicast traffic only to those interfaces that are interested in receiving it by using the **switchport** keyword.

IGMP snooping restricts traffic in MAC multicast groups 0100.5e00.0001 to 01-00-5e-ff-ff-ff. IGMP snooping does not restrict Layer 2 multicast packets generated by routing protocols.

**Note**

For more information on IP multicast and IGMP, refer to RFC 1112, RFC 2236, RFC 3376 (for IGMPv3).

IGMP (configured on a router) periodically sends out IGMP general queries. A host responds to these queries with IGMP membership reports for groups that it is interested in. When IGMP snooping is enabled, the switch creates one entry per-VLAN in the Layer 2 forwarding table for each Layer 2 multicast group from which it receives an IGMP join request. All hosts interested in this multicast traffic send IGMP membership reports and are added to the forwarding table entry.

Layer 2 multicast groups learned through IGMP snooping are dynamic. However, you can statically configure Layer 2 multicast groups using the **ip igmp snooping static** command. If you specify group membership statically, your setting supersedes any automatic manipulation by IGMP snooping. Multicast group membership lists can contain both user-defined and IGMP snooping settings.

Groups with IP addresses in the range 224.0.0.0 to 224.0.0.255, which map to the multicast MAC address range 0100.5E00.0001 to 0100.5E00.00FF, are reserved for routing control packets. These groups are flooded to all forwarding ports of the VLAN with the exception of 224.0.0.22, which is used for IGMPv3 membership reports.

**Note**

If a VLAN experiences a spanning-tree topology change, IP multicast traffic floods on all VLAN ports where PortFast is not enabled, as well as on ports with the **no igmp snooping tcn flood** command configured for a period of TCN query count.

For a Layer 2 IGMPv2 host interface to join an IP multicast group, a host sends an IGMP membership report for the IP multicast group. For a host to leave a multicast group, it can either ignore the periodic IGMP general queries or it can send an IGMP leave message. When the switch receives an IGMP leave message from a host, it sends out an IGMP group-specific query to determine whether any devices connected to that interface are interested in traffic for the specific multicast group. The switch then updates the table entry for that Layer 2 multicast group so that only those hosts interested in receiving multicast traffic for the group are listed.

In contrast, IGMPv3 hosts send IGMPv3 membership reports (with the **allow** group record mode) to join a specific multicast group. When IGMPv3 hosts send membership reports (with the **block** group record) to reject traffic from all sources in the previous source list, the last host on the port is removed by immediate-leave if EHT is enabled.

Immediate-Leave Processing

IGMP snooping immediate-leave processing allows the switch to remove an interface from the forwarding-table entry without first sending out IGMP group-specific queries to the interface. The VLAN interface is pruned from the multicast tree for the multicast group specified in the original IGMP leave message. Immediate-leave processing ensures optimal bandwidth management for all hosts on a switched network, even when multiple multicast groups are being used simultaneously.

When a switch with IGMP snooping enabled receives an IGMPv2 or IGMPv3 leave message, it sends an IGMP group-specific query from the interface where the leave message was received to determine when other hosts are attached to that interface that are interested in joining the MAC multicast group. If the switch does not receive an IGMP join message within the query response interval, the interface is removed from the port list of the (MAC-group, VLAN) entry in the Layer 2 forwarding table.

**Note**

By default all IGMP joins are forwarded to all multicast router ports.

With immediate-leave processing enabled on the VLAN, an interface can be removed immediately from the port list of the Layer 2 entry when the IGMP leave message is received, unless a multicast router was learned on the port.

**Note**

When using IGMPv2 snooping, use immediate-leave processing only on VLANs where just one host is connected to each interface. If immediate-leave processing is enabled on VLANs where multiple hosts are connected to an interface, some hosts might be dropped inadvertently. When using IGMPv3, immediate-leave processing is enabled by default, and due to explicit host tracking, the switch can detect when a port has single or multiple hosts maintained by the switch for IGMPv3 hosts. As a result, the switch can perform immediate-leave processing when it detects a single host behind a given port.

**Note**

IGMPv3 is interoperable with older versions of IGMP.

To display the IGMP version on a particular VLAN, use the **show ip igmp snooping querier vlan** command.

To display whether the switch supports IGMPv3 snooping, use the **show ip igmp snooping vlan** command.

To enable immediate-leave for IGMPv2, use the **ip igmp snooping immediate-leave** command.

**Note**

Immediate-leave processing is enabled by default for IGMPv3.

IGMP Configurable-Leave Timer

Immediate-leave processing cannot be used on VLANs where multiple hosts may be connected to a single interface. To reduce leave latency in such a scenario, IGMPv3 provides a configurable leave timer.

In Cisco IOS Release 12.2(25)SG and earlier, the IGMP snooping leave time was based on query response time. If membership reports were not received by the switch before the query response time of the query expired, a port was removed from the multicast group membership.

In Cisco IOS Release 12.2(31)SG and later, you can configure the length of time that the switch waits after sending a group-specific query to determine if hosts are still interested in a specific multicast group. The IGMP leave response time can be configured from 100 to 5000 milliseconds. The timer can be set either globally or per-VLAN. The VLAN configuration of the leave time overrides the global configuration.

For configuration steps, see the [“Configuring the IGMP Leave Timer” section on page 28-9](#).

IGMP Snooping Querier

IGMP Snooping Querier support was introduced in Cisco IOS Release 12.2(50)SG. This is a Layer 2 feature required to support IGMP snooping in a VLAN where PIM and IGMP are not configured because the multicast traffic does not require routing.

In a network where IP multicast routing is configured, the IP multicast router acts as the IGMP querier by sending general queries. If the IP-multicast traffic in a VLAN only needs to be Layer 2-switched, an IP-multicast router is not required. Without an IP-multicast router on the VLAN, you must configure another switch as the IGMP querier so that it can send queries.

When enabled, the IGMP snooping querier sends out periodic IGMPv2 queries that trigger IGMP report messages from the switch that requests IP multicast traffic. IGMP snooping listens to these IGMP reports to establish appropriate forwarding.

On switches that use IGMP to report interest in IP multicast traffic, configure at least one switch as the IGMP snooping querier in each supported VLAN.

You can configure a switch to generate IGMP queries on a VLAN regardless of whether IP multicast routing is enabled.

Explicit Host Tracking

Explicit host tracking (EHT) monitors group membership by tracking hosts that are sending IGMPv3 membership reports. This tracking enables a switch to detect host information associated with the groups of each port. EHT also enables the user to track the membership and various statistics.

EHT enables a switch to track membership on a per-port basis. Consequently, a switch is aware of the hosts residing on each port and can perform immediate-leave processing when there is only one host behind a port.

To determine whether EHT is enabled on a VLAN, use the **show ip igmp snoop vlan** command.

Configuring IGMP Snooping

**Note**

When configuring IGMP, configure the VLAN in the VLAN database mode. See [Chapter 17](#), “Configuring VLANs, VTP, and VMPS.”

IGMP snooping allows switches to examine IGMP packets and make forwarding decisions based on their content.

These sections describe how to configure IGMP snooping:

- [Default IGMP Snooping Configuration, page 28-5](#)
- [Enabling IGMP Snooping Globally, page 28-6](#)
- [Enabling IGMP Snooping on a VLAN, page 28-6](#)
- [Configuring Learning Methods, page 28-7](#)
- [Configuring a Static Connection to a Multicast Router, page 28-8](#)
- [Enabling IGMP Immediate-Leave Processing, page 28-8](#)
- [Configuring the IGMP Leave Timer, page 28-9](#)
- [Configuring IGMP Snooping Querier, page 28-10](#)
- [Configuring Explicit Host Tracking, page 28-11](#)
- [Configuring a Host Statically, page 28-11](#)
- [Suppressing Multicast Flooding, page 28-12](#)

Default IGMP Snooping Configuration

[Table 28-1](#) shows the IGMP snooping default configuration values.

Table 28-1 IGMP Snooping Default Configuration Values

Feature	Default Value
IGMP snooping	Enabled
Multicast routers	None configured
Explicit Host Tracking	Enabled for IGMPv3; Not available for IGMPv2
Immediate-leave processing	Enabled for IGMPv3; Disabled for IGMPv2
Report Suppression	Enabled
IGMP snooping learning method	PIM/DVMRP ¹

1. PIM/DVMRP = Protocol Independent Multicast/Distance Vector Multicast Routing Protocol

Enabling IGMP Snooping Globally

To enable IGMP snooping globally, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# [no] ip igmp snooping	Enables IGMP snooping. Use the no keyword to disable IGMP snooping.
Step 3	Switch(config)# end	Exits configuration mode.
Step 4	Switch# show ip igmp snooping include	Verifies the configuration.

This example shows how to enable IGMP snooping globally and verify the configuration:

```
Switch# configure terminal
Switch(config)# ip igmp snooping
Switch(config)# end
Switch# show ip igmp snooping
Global IGMP Snooping configuration:
-----
IGMP snooping           : Enabled
IGMPv3 snooping         : Enabled
Report suppression      : Enabled
TCN solicit query       : Disabled
TCN flood query count    : 2

Vlan 1:
-----
IGMP snooping           : Enabled
IGMPv2 immediate leave   : Disabled
Explicit host tracking    : Enabled
Multicast router learning mode : pim-dvmrp
CGMP interoperability mode : IGMP_ONLY

Vlan 2:
-----
IGMP snooping           : Enabled
IGMPv2 immediate leave   : Disabled
Explicit host tracking    : Enabled
Multicast router learning mode : pim-dvmrp
CGMP interoperability mode : IGMP_ONLY
```

Enabling IGMP Snooping on a VLAN

To enable IGMP snooping on a VLAN, perform this task:

	Command	Purpose
Step 1	Switch(config)# [no] ip igmp snooping vlan vlan_ID	Enables IGMP snooping. Use the no keyword to disable IGMP snooping.
Step 2	Switch(config)# end	Exits configuration mode.
Step 3	Switch# show ip igmp snooping vlan vlan_ID	Verifies the configuration.

This example shows how to enable IGMP snooping on VLAN 2 and verify the configuration:

```
Switch# configure terminal
Switch(config)# ip igmp snooping vlan 2
Switch(config)# end
Switch# show ip igmp snooping vlan 2
Global IGMP Snooping configuration:
-----
IGMP snooping           : Enabled
IGMPv3 snooping         : Enabled
Report suppression      : Enabled
TCN solicit query       : Disabled
TCN flood query count    : 2

Vlan 2:
-----
IGMP snooping           : Enabled
IGMPv2 immediate leave  : Disabled
Explicit host tracking   : Enabled
Multicast router learning mode : pim-dvmrp
CGMP interoperability mode : IGMP_ONLY
```

Configuring Learning Methods

The following sections describe IGMP snooping learning methods:

- [Configuring PIM/DVMRP Learning, page 28-7](#)
- [Configuring CGMP Learning, page 28-7](#)

Configuring PIM/DVMRP Learning

To configure IGMP snooping to learn from PIM/DVMRP packets, perform this task:

Command	Purpose
Switch(config)# ip igmp snooping vlan <i>vlan_ID</i> mrouter learn [cgmp pim-dvmrp]	Specifies the learning method for the VLAN.

This example shows how to configure IP IGMP snooping to learn from PIM/DVMRP packets:

```
Switch# configure terminal
Switch(config)# ip igmp snooping vlan 1 mrouter learn pim-dvmrp
Switch(config)# end
Switch#
```

Configuring CGMP Learning

To configure IGMP snooping to learn from CGMP self-join packets, perform this task:

Command	Purpose
Switch(config)# ip igmp snooping vlan <i>vlan_ID</i> mrouter learn [cgmp pim-dvmrp]	Specifies the learning method for the VLAN.

This example shows how to configure IP IGMP snooping to learn from CGMP self-join packets:

```
Switch# configure terminal
Switch(config)# ip igmp snooping vlan 1 mrouter learn cgmp
Switch(config)# end
Switch#
```

Configuring a Static Connection to a Multicast Router

To configure a static connection to a multicast router, enter the **ip igmp snooping vlan mrouter interface** command on the switch.

To configure a static connection to a multicast router, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# ip igmp snooping vlan <i>vlan_ID</i> mrouter interface <i>interface_num</i>	Specifies a static connection to a multicast router for the VLAN. Note The interface to the router must be in the VLAN where you are entering the command. The router and the line protocol must be up.
Step 3	Switch(config)# end	Exits configuration mode.
Step 4	Switch# show ip igmp snooping mrouter vlan <i>vlan_ID</i>	Verifies the configuration.

This example shows how to configure a static connection to a multicast router:

```
Switch# configure terminal
Switch(config)# ip igmp snooping vlan 200 mrouter interface fastethernet 2/10
Switch# show ip igmp snooping mrouter vlan 200
vlan  ports
-----+-----
 200   Fa2/10
Switch#
```

Enabling IGMP Immediate-Leave Processing

When you enable IGMP immediate-leave processing on a VLAN, a switch removes an interface from the multicast group when it detects an IGMPv2 leave message on that interface.



Note

For IGMPv3, immediate-leave processing is enabled by default with EHT.

To enable immediate-leave processing on an IGMPv2 interface, perform this task:

Command	Purpose
Switch(config)# ip igmp snooping vlan <i>vlan_ID</i> immediate-leave	Enables immediate-leave processing in the VLAN. Note This command applies only to IGMPv2 hosts.

This example shows how to enable IGMP immediate-leave processing on interface VLAN 200 and to verify the configuration:

```
Switch# configure terminal
Switch(config)# ip igmp snooping vlan 200 immediate-leave
Configuring immediate leave on vlan 200
Switch(config)# end
Switch# show ip igmp interface vlan 200 | include immediate leave
Immediate leave                : Disabled
Switch(config)#
```

Configuring the IGMP Leave Timer

Follows these guidelines when configuring the IGMP leave timer:

- You can configure the leave time globally or per-VLAN.
- Configuring the leave time on a VLAN overrides the global setting.
- The default leave time is 1000 milliseconds.
- The IGMP configurable leave time is only supported on hosts running IGMP Version 2.
- The actual leave latency in the network is usually the configured leave time. However, the leave time *might* vary around the configured time, depending on real-time CPU load conditions, network delays and the amount of traffic sent through the interface.

To enable the IGMP configurable-leave timer, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# ip igmp snooping last-member-query-interval time	Configures the IGMP leave timer globally. The range is 100 to 5000 milliseconds. The default is 1000 seconds. To globally reset the IGMP leave timer to the default setting, use the global configuration command no ip igmp snooping last-member-query-interval .
Step 3	Switch(config)# ip igmp snooping vlan vlan_ID last-member-query-interval time	(Optional) Configures the IGMP leave time on the VLAN interface. The range is 100 to 5000 milliseconds. To remove the configured IGMP leave-time setting from the specified VLAN, use the global configuration command no ip igmp snooping vlan vlan-id last-member-query-interval Note Configuring the leave time on a VLAN overrides the globally configured timer.
Step 4	Switch(config)# end	Returns to privileged EXEC mode.
Step 5	Switch# show ip igmp snooping	(Optional) Displays the configured IGMP leave time.
Step 6	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to enable the IGMP configurable-leave timer and to verify the configuration:

```
Switch# configure terminal
Switch(config)# ip igmp snooping last-member-query-interval 200
Switch(config)# ip igmp snooping vlan 10 last-member-query-interval 500
Switch(config)# end
```

```
Switch# show ip igmp snooping show ip igmp snooping
Global IGMP Snooping configuration:
-----

IGMP snooping           : Enabled
IGMPv3 snooping         : Enabled
Report suppression      : Enabled
TCN solicit query       : Disabled
TCN flood query count    : 2
Last Member Query Interval : 200

Vlan 1:
-----
IGMP snooping           : Enabled
IGMPv2 immediate leave  : Disabled
Explicit host tracking   : Enabled
Multicast router learning mode : pim-dvmrp
Last Member Query Interval : 200
CGMP interoperability mode : IGMP_ONLY

Vlan 10:
-----
IGMP snooping           : Enabled
IGMPv2 immediate leave  : Disabled
Explicit host tracking   : Enabled
Multicast router learning mode : pim-dvmrp
Last Member Query Interval : 500
CGMP interoperability mode : IGMP_ONLY

Switch#
```

Configuring IGMP Snooping Querier

The IGMP Snooping Querier feature can be enabled either globally or per-VLAN.



Note

The IGMP snooping querier is disabled by default.

To configure IGMP Snooping Querier, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# [no] ip igmp snooping [vlan vlan_id] querier	Enables IGMP Snooping Querier.
Step 3	Switch(config)# [no] ip igmp snooping [vlan vlan_id] querier address abcd	Configures the IGMP Snooping Querier source IP address.
Step 4	Switch(config)# [no] ip igmp snooping [vlan vlan_id] querier version [1 2]	Configures IGMP Snooping Querier IGMP version.
Step 5	Switch(config)# ip igmp snooping [vlan vlan_id] querier query-interval interval	Configures IGMP Snooping Querier query interval.
Step 6	Switch(config)# ip igmp snooping [vlan vlan_id] querier max-response-time value	Configures IGMP Snooping Querier query maximum response time.
Step 7	Switch(config)# ip igmp snooping [vlan vlan_id] querier timer expiry value	Configures IGMP Snooping Querier expiry time out.

	Command	Purpose
Step 8	Switch(config)# ip igmp snooping [vlan vlan_id] querier tcq query count value	Configures IGMP Snooping Querier tcq query count.
Step 9	Switch(config)# ip igmp snooping [vlan vlan_id] querier tcq query interval value	Configures IGMP Snooping Querier tcq query interval.
Step 10	Switch(config) # end	Returns to privileged EXEC mode.

For an example of how to display Snooping Querier information, refer to the [“Displaying IGMP Snooping Querier Information”](#) section on page 28-19.

Configuring Explicit Host Tracking

For IGMPv3, EHT is enabled by default and can be disabled on a per-VLAN basis.

To disable EHT processing on a VLAN, perform this task:

Command	Purpose
Switch(config)# [no] ip igmp snooping vlan vlan_ID explicit-tracking	Enables EHT on a VLAN. The no keyword disables EHT.

This example shows how to disable IGMP EHT on VLAN 200 and to verify the configuration:

```
Switch# configure terminal
Switch(config)# no ip igmp snooping vlan 200 explicit-tracking
Switch(config)# end
Switch# show ip igmp snooping vlan 200 | include Explicit host tracking
Explicit host tracking           : Disabled
```

Configuring a Host Statically

Hosts normally join multicast groups dynamically, but you can also configure a host statically on an interface.

To configure a host statically on an interface, perform this task:

Command	Purpose
Switch(config-if)# ip igmp snooping vlan vlan_ID static mac_address interface interface_num	Configures a host statically in the VLAN. Note This command cannot be configured to receive traffic for specific source IP addresses.

This example shows how to configure a host statically in VLAN 200 on interface Fast Ethernet 2/11:

```
Switch# configure terminal
Switch(config)# ip igmp snooping vlan 200 static 0100.5e02.0203 interface fastethernet
2/11
Configuring port FastEthernet2/11 on group 0100.5e02.0203 vlan 200
Switch(config)# end
```

Suppressing Multicast Flooding

An IGMP snooping-enabled switch floods multicast traffic to all ports in a VLAN when a spanning-tree topology change notification (TCN) is received. Multicast flooding suppression enables a switch to stop sending such traffic. To support flooding suppression, the following interface and global commands were introduced in Cisco IOS Release 12.1(11b)EW:

- **[no | default] ip igmp snooping tcn flood** (interface command)
- **[no | default] ip igmp snooping tcn flood query count [1 - 10]** (global command)
- **[no | default] ip igmp snooping tcn query solicit** (global command)

Prior to Cisco IOS Release 12.1(11b)EW, when a spanning tree topology change notification (TCN) was received by a switch, the multicast traffic was flooded to all the ports in a VLAN for a period of three IGMP query intervals. This was necessary for redundant configurations. In Cisco IOS Release 12.1(11b)EW, the default time period the switch waits before multicast flooding stops was changed to two IGMP query intervals.

This flooding behavior is undesirable if the switch that does the flooding has many ports that are subscribed to different groups. The traffic could exceed the capacity of the link between the switch and the end host, resulting in packet loss.

With the **no ip igmp snooping tcn flood** command, you can disable multicast flooding on a switch interface following a topology change. Only the multicast groups that have been joined by a port are sent to that port, even during a topology change.

With the **ip igmp snooping tcn flood query count** command, you can enable multicast flooding on a switch interface for a short period of time following a topology change by configuring an IGMP query threshold.

Typically, if a topology change occurs, the spanning tree root switch issues a global IGMP leave message (referred to as a “query solicitation”) with the group multicast address 0.0.0.0. When a switch receives this solicitation, it floods this solicitation on all ports in the VLAN where the spanning tree change occurred. When the upstream router receives this solicitation, it immediately issues an IGMP general query.

With the **ip igmp snooping tcn query solicit** command, you can now direct a non-spanning tree root switch to enter the same query solicitation.

The following sections provide additional details on the new commands and illustrate how you can use them.

IGMP Snooping Interface Configuration

A topology change in a VLAN may invalidate previously learned IGMP snooping information. A host that was on one port before the topology change may move to another port after the topology change. When the topology changes, the Catalyst 4500 series switch takes special actions to ensure that multicast traffic is delivered to all multicast receivers in that VLAN.

When the spanning tree protocol is running in a VLAN, a spanning tree topology change notification (TCN) is issued by the root switch in the VLAN. A Catalyst 4500 series switch that receives a TCN in a VLAN for which IGMP snooping has been enabled immediately enters into multicast flooding mode for a period of time until the topology restabilizes and the new locations of all multicast receivers are learned.

While in multicast flooding mode, IP multicast traffic is delivered to all ports in the VLAN, and not restricted to those ports on which multicast group members have been detected.

Starting with Cisco IOS Release 12.1(11b)EW, you can manually prevent IP multicast traffic from being flooded to a switch port by using the **no ip igmp snooping tcn flood** command on that port.

For trunk ports, the configuration applies to all VLANs.

By default, multicast flooding is enabled. Use the **no** keyword to disable flooding, and use **default** to restore the default behavior (flooding is enabled).

To disable multicast flooding on an interface, perform this task:

	Command	Purpose
Step 1	Switch(config)# interface { fastethernet gigabitethernet tengigabitethernet } <i>slot/port</i>	Selects the interface to configure.
Step 2	Switch(config-if)# no ip igmp snooping tcn flood	Disables multicast flooding on the interface when TCNs are received by the switch. To enable multicast flooding on the interface, enter this command: default ip igmp snooping tcn flood
Step 3	Switch(config)# end	Exits configuration mode.
Step 4	Switch# show running interface { fastethernet gigabitethernet tengigabitethernet } <i>slot/port</i>	Verifies the configuration.

This example shows how to disable multicast flooding on interface Fast Ethernet 2/11:

```
Switch(config)# interface fastethernet 2/11
Switch(config-if)# no ip igmp snooping tcn flood
Switch(config-if)# end
Switch#
```

IGMP Snooping Switch Configuration

By default, flooding mode persists until the switch receives two IGMP general queries. You can change this period of time by using the **ip igmp snooping tcn flood query count** *n* command, where *n* is a number between 1 and 10.

This command operates at the global configuration level.

The default number of queries is 2. The **no** and **default** keywords restore the default.

To establish an IGMP query threshold, perform this task:

	Command	Purpose
Step 1	Switch(config)# ip igmp snooping tcn flood query count <n>	Modifies the number of IGMP queries the switch waits for before it stops flooding multicast traffic. To return the switch to the default number of IGMP queries, enter this command: default ip igmp snooping tcn flood query count.
Step 2	Switch(config)# end	Exits configuration mode.

This example shows how to modify the switch to stop flooding multicast traffic after four queries:

```
Switch(config)# ip igmp snooping tcn flood query count 4
Switch(config)# end
Switch#
```

When a spanning tree root switch receives a topology change in an IGMP snooping-enabled VLAN, the switch issues a query solicitation that causes an Cisco IOS router to send out one or more general queries. The new command **ip igmp snooping tcn query solicit** causes the switch to send the query solicitation whenever it notices a topology change, even if that switch is not the spanning tree root.

This command operates at the global configuration level.

By default, query solicitation is disabled unless the switch is the spanning tree root. The **default** keyword restores the default behavior.

To direct a switch to send a query solicitation, perform this task:

	Command	Purpose
Step 1	Switch(config)# ip igmp snooping tcn query solicit	Configures the switch to send a query solicitation when a TCN is detected. To stop the switch from sending a query solicitation (if it is not a spanning tree root switch), enter the no ip igmp snooping tcn query solicit command.
Step 2	Switch(config)# end	Exits configuration mode.

This example shows how to configure the switch to send a query solicitation upon detecting a TCN:

```
Switch(config)# ip igmp snooping tcn query solicit
Switch(config)# end
Switch#
```

Displaying IGMP Snooping Information

The following sections show how to display IGMP snooping information:

- [Displaying Querier Information, page 28-15](#)
- [Displaying IGMP Host Membership Information, page 28-15](#)
- [Displaying Group Information, page 28-16](#)
- [Displaying Multicast Router Interfaces, page 28-17](#)

- [Displaying MAC Address Multicast Entries, page 28-18](#)
- [Displaying IGMP Snooping Information on a VLAN Interface, page 28-18](#)
- [Configuring IGMP Filtering, page 28-30](#)

Displaying Querier Information

To display querier information, perform this task:

Command	Purpose
Switch# show ip igmp snooping querier [vlan <i>vlan_ID</i>]	Displays multicast router interfaces.

This example shows how to display the IGMP snooping querier information for all VLANs on the switch:

```
Switch# show ip igmp snooping querier
Vlan      IP Address      IGMP Version      Port
-----
2         10.10.10.1      v2                 Router
3         172.20.50.22    v3                 Fa3/15
```

This example shows how to display the IGMP snooping querier information for VLAN 3:

```
Switch# show ip igmp snooping querier vlan 3
Vlan      IP Address      IGMP Version      Port
-----
3         172.20.50.22    v3                 Fa3/15
```

Displaying IGMP Host Membership Information



Note

By default, EHT maintains a maximum of 1000 entries in the EHT database. Once this limit is reached, no additional entries are created. To create additional entries, clear the database with the **clear ip igmp snooping membership vlan** command.

To display host membership information, perform this task:

Command	Purpose
Switch# show ip igmp snooping membership [interface <i>interface_num</i>] [vlan <i>vlan_ID</i>] [reporter <i>a.b.c.d</i>] [source <i>a.b.c.d</i> group <i>a.b.c.d</i>]	Displays EHT information. Note This command is valid only if EHT is enabled on the switch.

This example shows how to display host membership information for VLAN 20 and to delete the EHT database:

```
Switch# show ip igmp snooping membership vlan 20
#channels: 5
#hosts : 1
Source/Group Interface Reporter Uptime Last-Join Last-Leave
```

```

40.40.40.2/224.10.10.10 Gi4/1 20.20.20.20 00:23:37 00:06:50 00:20:30
40.40.40.3/224.10.10.10 Gi4/2 20.20.20.20 00:23:37 00:06:50 00:20:30
40.40.40.4/224.10.10.10Gi4/1 20.20.20.20 00:39:42 00:09:17 -

40.40.40.5/224.10.10.10Fa2/1 20.20.20.20 00:39:42 00:09:17 -
40.40.40.6/224.10.10.10 Fa2/1 20.20.20.20 00:09:47 00:09:17 -

```

```
Switch# clear ip igmp snooping membership vlan 20
```

This example shows how to display host membership for interface gi4/1:

```

Switch# show ip igmp snooping membership interface gi4/1
#channels: 5
#hosts : 1
Source/Group Interface Reporter Uptime Last-Join Last-Leave

40.40.40.2/224.10.10.10 Gi4/1 20.20.20.20 00:23:37 00:06:50 00:20:30
40.40.40.4/224.10.10.10Gi4/1 20.20.20.20 00:39:42 00:09:17 -

```

This example shows how to display host membership for VLAN 20 and group 224.10.10.10:

```

Switch# show ip igmp snooping membership vlan 20 source 40.40.40.2 group 224.10.10.10
#channels: 5
#hosts : 1
Source/Group Interface Reporter Uptime Last-Join Last-Leave

40.40.40.2/224.10.10.10 Gi4/1 20.20.20.20 00:23:37 00:06:50 00:20:30

```

Displaying Group Information

To display detailed IGMPv3 information associated with a group, perform one of the following tasks:

Command	Purpose
Switch# show ip igmp snooping groups [vlan <i>vlan_ID</i>]	<p>Displays groups, the type of reports that were received for the group (Host Type), and the list of ports on which reports were received.</p> <p>The report list includes neither the multicast router ports nor the complete forwarding port set for the group. It lists the ports on which the reports have been received.</p> <p>To display the complete forwarding port set for the group, display the CLI output for the MAC address that maps to this group by using the show mac-address-table multicast command.</p>
Switch# show ip igmp snooping groups [vlan <i>vlan_ID</i> <i>a.b.c.d</i>] [<i>summary/sources/hosts</i>]	<p>Displays information specific to a group address, providing details about the current state of the group with respect to sources and hosts.</p> <p>Note This command applies only to full IGMPv3 snooping support and can be used for IGMPv1, IGMPv2, or IGMPv3 groups.</p>
Switch# show ip igmp snooping groups [vlan <i>vlan_ID</i>] [<i>count</i>]	Displays the total number of group addresses learned by the system on a global or per-VLAN basis.

This example shows how to display the host types and ports of a group in VLAN 1:

```
Switch# show ip igmp snooping groups vlan 10 226.6.6.7
Vlan      Group      Version  Ports
-----
10        226.6.6.7    v3       Fa7/13, Fa7/14
Switch>
```

This example shows how to display the current state of a group with respect to a source IP address:

```
Switch# show ip igmp snooping groups vlan 10 226.6.6.7 sources
Source information for group 226.6.6.7:
Timers: Expired sources are deleted on next IGMP General Query

SourceIP      Expires    Uptime      Inc Hosts  Exc Hosts
-----
2.0.0.1       00:03:04   00:03:48    2          0
2.0.0.2       00:03:04   00:02:07    2          0
Switch>
```

This example shows how to display the current state of a group with respect to a host MAC address:

```
Switch# show ip igmp snooping groups vlan 10 226.6.6.7 hosts
IGMPv3 host information for group 226.6.6.7
Timers: Expired hosts are deleted on next IGMP General Query

Host (MAC/IP)  Filter mode  Expires    Uptime      # Sources
-----
175.1.0.29     INCLUDE     stopped    00:00:51     2
175.2.0.30     INCLUDE     stopped    00:04:14     2
```

This example shows how to display summary information for an IGMPv3 group:

```
Switch# show ip igmp snooping groups vlan 10 226.6.6.7 summary
Group Address (Vlan 10)      : 226.6.6.7
Host type                    : v3
Member Ports                 : Fa7/13, Fa7/14
Filter mode                  : INCLUDE
Expires                      : stopped
Sources                      : 2
Reporters (Include/Exclude)  : 2/0
```

This example shows how to display the total number of group addresses learned by the system globally:

```
Switch# show ip igmp snooping groups count
Total number of groups: 54
```

This example shows how to display the total number of group addresses learned on VLAN 5:

```
Switch# show ip igmp snooping groups vlan 5 count
Total number of groups: 30
```

Displaying Multicast Router Interfaces

When you enable IGMP snooping, the switch automatically learns to which interface the multicast routers are connected.

To display multicast router interfaces, perform this task:

Command	Purpose
Switch# show ip igmp snooping mrouter vlan <i>vlan_ID</i>	Displays multicast router interfaces.

This example shows how to display the multicast router interfaces in VLAN 1:

```
Switch# show ip igmp snooping mrouter vlan 1
vlan                ports
-----+-----
  1                Gi1/1,Gi2/1,Fa3/48,Router
Switch#
```

Displaying MAC Address Multicast Entries

To display MAC address multicast entries for a VLAN, perform this task:

Command	Purpose
Switch# show mac-address-table multicast vlan <i>vlan_ID</i> [<i>count</i>]	Displays MAC address multicast entries for a VLAN.

This example shows how to display MAC address multicast entries for VLAN 1:

```
Switch# show mac-address-table multicast vlan 1
Multicast Entries
vlan    mac address      type    ports
-----+-----
  1     0100.5e01.0101    igmp    Switch,Gi6/1
  1     0100.5e01.0102    igmp    Switch,Gi6/1
  1     0100.5e01.0103    igmp    Switch,Gi6/1
  1     0100.5e01.0104    igmp    Switch,Gi6/1
  1     0100.5e01.0105    igmp    Switch,Gi6/1
  1     0100.5e01.0106    igmp    Switch,Gi6/1
Switch#
```

This example shows how to display a total count of MAC address entries for VLAN 1:

```
Switch# show mac-address-table multicast vlan 1 count
Multicast MAC Entries for vlan 1:    4
Switch#
```

Displaying IGMP Snooping Information on a VLAN Interface

To display IGMP snooping information on a VLAN, perform this task:

Command	Purpose
Switch# show ip igmp snooping vlan <i>vlan_ID</i>	Displays IGMP snooping information on a VLAN interface.

This example shows how to display IGMP snooping information on VLAN 5:

```
Switch# show ip igmp snooping vlan 5
Global IGMP Snooping configuration:
-----
IGMP snooping                :Enabled
IGMPv3 snooping support      :Full
Report suppression           :Enabled
TCN solicit query             :Disabled
TCN flood query count         :2

Vlan 5:
-----
IGMP snooping                :Enabled
Immediate leave               :Disabled
Explicit Host Tracking        :Disabled
Multicast router learning mode :pim-dvmrp
CGMP interoperability mode     :IGMP_ONLY
```

Displaying IGMP Snooping Querier Information

To display IGMP Snooping Querier information, perform this task:

Command	Purpose
Switch# show ip igmp snooping querier [vlan <i>vlan_ID</i>] [detail]	Displays the IGMP Snooping Querier state.

This example shows how to display Snooping Querier information:

```
switch# show ip igmp snooping querier vlan 2 detail
IP address          : 1.2.3.4
IGMP version         : v2
Port                : Router/Switch
Max response time    : 12s

Global IGMP switch querier status
-----
admin state          : Enabled
admin version         : 2
source IP address     : 1.2.3.4
query-interval (sec)  : 130
max-response-time (sec) : 10
querier-timeout (sec) : 100
tcn query count       : 2
tcn query interval (sec) : 10
```

```

Vlan 2:  IGMP switch querier status
-----
admin state                : Enabled
admin version              : 2
source IP address          : 1.2.3.4
query-interval (sec)       : 55
max-response-time (sec)    : 12
querier-timeout (sec)      : 70
tcn query count            : 10
tcn query interval (sec)   : 8
operational state          : Querier
operational version        : 2
tcn query pending count    : 0

```

Understanding Multicast VLAN Registration

When a network involves multi-VLAN's, subscribers to a multicast group may exist in more than one VLAN (i.e., the broadcast of multiple television channels over a service provider network). The multicast router must replicate the multicast data transmission to the same group in the every subscriber VLANs. The number of multicast stream replication is directly proportional to the subscriber VLANs. This results in using more than the required bandwidth.

Multicast VLAN Registration (MVR) overcomes this inefficiency by conserving network bandwidth. MVR allows a subscriber on a port to subscribe and unsubscribe to a multicast stream on the network-wide single "multicast VLAN," while subscribers remain in separate VLANs. It also isolates the streams from the subscriber VLANs for bandwidth and security reasons.



Note

Only Layer 2 ports participate in MVR.



Note

You need to configure subscriber ports as *MVR receiver ports* and router or data-source ports as *MVR source ports*.



Note

Only one MVR multicast VLAN per switch is supported.

MVR assumes that subscriber ports subscribe and unsubscribe (join and leave) these multicast streams by sending out IGMP join and leave messages. These messages can originate from an IGMP v2 compatible host with an Ethernet connection. Although MVR and IGMP snooping use the same underlying mechanism, the two features operate independently. You can enable or disable one without affecting the behavior of the other. However, if IGMP snooping and MVR are both enabled, MVR reacts only to join and leave messages from multicast groups configured under MVR. Join and leave messages from all other multicast groups are managed by IGMP snooping.

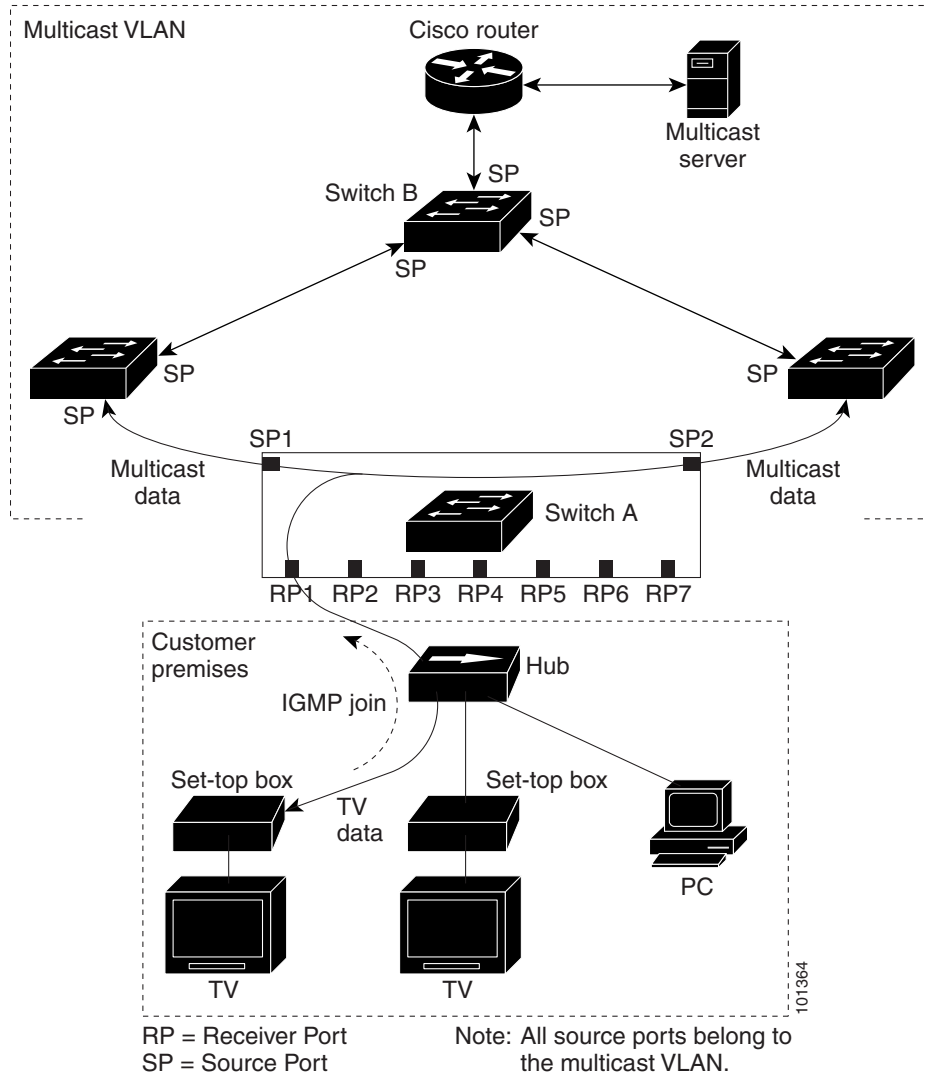
You can set the switch to operate MVR in compatible or dynamic mode:

- In *compatible* mode, a multicast router learned or configured is not required for MVR traffic to egress MVR source ports. All the MVR traffic is forwarded to the source ports. The IGMP reports that are received by the receiver ports are not forwarded to the mrouter or source ports.

- In *dynamic* mode, the interface on which the multicast router is learned or configured will receive MVR traffic. The receiver ports from where the MVR hosts have explicitly joined either by IGMP reports or by MVR static configuration will receive the MVR data traffic. The IGMP reports are forwarded to all the multicast VLAN (mVLAN) mrouter ports.

Using MVR in a Multicast Television Application

In a multicast television application, a PC or a television with a set-top box can receive the multicast stream. Multiple set-top boxes or PCs can be connected to one subscriber port, which is a switch port configured as an MVR receiver port. [Figure 28-1](#) is an example configuration. DHCP assigns an IP address to the set-top box or the PC. When a subscriber selects a channel, the set-top box or PC sends an IGMP report to Switch A to join the appropriate multicast. If the IGMP report matches one of the configured IP multicast group addresses, the switch CPU modifies the hardware address table to include this receiver port and VLAN as a forwarding destination of the specified multicast stream when it is received from the multicast VLAN. Uplink ports that send and receive multicast data to and from the multicast VLAN are called MVR *source* ports.

Figure 28-1 Multicast VLAN Registration Example

When a subscriber changes channels or turns off the television, the set-top box sends an IGMP leave message for the multicast stream. The switch CPU sends a MAC-based general query through the receiver port VLAN. If there is another set-top box in the VLAN still subscribing to this group, that set-top box must respond within the maximum response time specified in the query. If the CPU does not receive a response, it eliminates the receiver port as a forwarding destination for this group.

Without Immediate Leave, when the switch receives an IGMP leave message from a subscriber on a receiver port, it sends out an IGMP query on that port and waits for IGMP group membership reports. If no reports are received in a configured time period, the receiver port is removed from multicast group membership. With Immediate Leave, an IGMP query is not sent from the receiver port on which the IGMP leave was received. Once the leave message is received, the receiver port is removed from multicast group membership, which speeds up leave latency. Enable Immediate Leave feature only on receiver ports to which a single receiver device is connected.

Because MVR multicast traffic is sent only on mVLANs, duplicating television-channel multicast traffic for subscribers on different VLANs is unnecessary. The IGMP leave and join messages are in the VLAN to which the subscriber port is assigned. The access layer switch (Switch A) modifies the forwarding behavior to allow traffic forwarding from the multicast VLAN to the subscriber port in a different VLAN. This is done by selectively allowing traffic to cross between the two VLANs.

IGMP reports are sent to the same IP multicast group address as the multicast data. The Switch A CPU must capture all IGMP join and leave messages from the receiver ports and forward them to the multicast VLAN of the source (uplink) port, based on the MVR mode.

Configuring MVR

These sections include basic MVR configuration information:

- [Default MVR Configuration, page 28-23](#)
- [MVR Configuration Guidelines and Limitations, page 28-23](#)
- [Configuring MVR Global Parameters, page 28-24](#)
- [Configuring MVR on Access Ports, page 28-26](#)
- [Configuring MVR on a Trunk Port, page 28-27](#)
- [Displaying MVR Information, page 28-29](#)

Default MVR Configuration

**Note**

Enabling the **MVR** command will set all the MVR default parameters.

[Table 28-2](#) shows the default MVR configuration.

Table 28-2 **Default MVR Configuration**

Feature	Default Setting
MVR	Disabled globally
Multicast addresses	None configured
Query response time	5 (tenths of a second)
Multicast VLAN	VLAN 1
Mode	Compatible
Interface (per port) default	Neither a receiver nor a source port
Immediate Leave	Disabled on all ports

MVR Configuration Guidelines and Limitations

Follow these guidelines when configuring MVR:

- Ports can be configured as either a *source port* or a *receiver port*.
 - Ports connected to subscribers are configured as receiver ports.

- Router ports or ports that are connected to another MVR switch are configured as source ports.
- Compatible mode

A source port configuration is required for those ports that must receive MVR traffic, even when there is no JOIN request from that port. All the MVR traffic received on the mVLAN on any port is forwarded to all source and receiver ports. (The receiver port should have been joined either by IGMP report or through static configurations).
- Dynamic mode

Source port configuration is not required, unless there is a port connected to another Layer 2 switch that runs MVR on the same mVLAN as this switch. Configure such ports as *source* ports. All MVR traffic received on the mVLAN on any ports is forwarded to the receiver or source ports that are joined either by IGMP report or through static configurations.
- Only one MVR VLAN can be configured.
- Although receiver ports that are connected to subscribers can be on different VLANs, they should not belong to the mVLAN.
- mRouter ports should not be configured as receiver ports.
- Both trunk and access ports can be configured as either source or receiver ports.
- The maximum number of MVR groups is fixed at 1500.
- MVR cannot coexist with a PVLAN; do not configure MVR on a PVLAN.
- The IGMPSN group MAC address can alias with an MVR group's MAC address.

For example, 225.1.1.1 and 226.1.1.1 are IP addresses whose MAC addresses match to the same multicast MAC address (0100.5e01.0101). If 225.1.1.1 is configured as an MVR group then 225.1.1.1 is handled by MVR and 226.1.1.1 is handled by IGMPSN.

If the 226.1.1.1 host is present on the MVR trunk receiver, IGMPSN might not handle the forwarding for 226.1.1.1. Instead, the switch treats 226.1.1.1 as an MVR group and MVR handles forwarding on the mVLAN. You should not connect the hosts *interested* in MVR aliased groups on the MVR trunk receiver port. (By *interested* we mean that a host sends a JOIN request for a multicast group in order to receive the traffic or stream for that group.) This limitation applies only to MVR trunk receiver ports.
- MVR and multicast-routing cannot co-exist on the same switch. If you try to enable MVR while multicast routing or a multicast routing protocol are enabled, your operation is cancelled and you receive an error message. If you enable multicast routing or a multicast routing protocol while MVR is enabled, MVR is disabled and you receive a warning message.
- MVR can coexist with IGMP snooping on a switch.
- MVR is not supported with IPv6 multicast groups.
- MVR supports only IGMPv2 messages; MVR group reports derived from other versions are dropped.

Configuring MVR Global Parameters

If you select the default settings, you do not need to set the optional MVR parameters. If you want to change the default parameters (except for the MVR VLAN), you must first enable MVR.

To configure MVR parameters, perform these steps:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# mvr	Enables MVR on the switch.
Step 3	Switch(config)# mvr group <i>ip-address [count]</i>	Configures an IP multicast address on the switch or uses the count parameter to configure a contiguous series of MVR group addresses (maximum of 1500 groups).
Step 4	Switch(config)# mvr querytime <i>value</i>	(optional) Defines the maximum wait time for IGMP report memberships on a receiver port before removing the port from multicast group membership.
Step 5	Switch(config)# mvr vlan <i>vlan-id</i>	Specifies the VLAN in which multicast data is received; all source ports must belong to this VLAN. The VLAN ID range is 1 to 1001 and 1006 to 4094. The default is VLAN 1.
Step 6	Switch(config)# mvr mode { dynamic compatible }	Specifies the MVR mode of operation.
Step 7	Switch(config)# end	Returns to privileged EXEC mode.
Step 8	Switch# show mvr OR show mvr members	Verifies the configuration.
Step 9	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To return a switch to the default settings, use the **no mvr [mode | group ip-address | querytime | vlan]** global configuration commands.

The following example shows how to enable and verify MVR:

```
Switch(config)# mvr
Switch(config)# mvr vlan 100
Switch(config)# mvr group 225.1.1.1
Switch(config)# mvr querytime 10
Switch(config)# mvr mode dynamic
Switch(config)# end
Switch# show mvr
MVR Running: TRUE
MVR multicast VLAN: 100
MVR Max Multicast Groups: 1500
MVR Current multicast groups: 1
MVR Global query response time: 10 (tenths of sec)
MVR Mode: dynamic
Switch# show mac address-table
Multicast Entries
  vlan      mac address      type      ports
-----+-----+-----+-----
100        0100.5e01.0101      igmp Fa2/1

Switch# show platform hardware mac-address-table address 0100.5e01.0101
Flags are:
-----
D - Drop
ND - Do not drop
Index  Mac Address      Vlan  Type      SinglePort/RetIndex/AdjIndex
-----
40048  0100.5E01.0101    100   Ret       104444

Switch# show platform hardware ret chain index 104444
RetIndex 104444
```

```
RetWordIndex: 522220 Link: 1048575(0xFFFFF) FieldsCnt: 1
SuppressRxVlanBridging: true
Vlan: 100 BridgeOnly: N Fa2/1(8)
```

**Note**

Fa2/1 is an mrouter port.

Configuring MVR on Access Ports

To configure the access port, perform these steps:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# mvr	Enables MVR on the switch.
Step 3	Switch(config)# interface interface-id	Enters interface configuration mode, and enter the type and number of the Layer 2 port to configure.
Step 4	Switch(config-if)# switch mode access	Change the interface to access mode.
Step 5	Switch(config-if)# switch access vlan value	Assign the VLAN to the port.
Step 6	Switch(config-if)# mvr type {source receiver}	Configures an MVR port as source or receiver: source —Subscribers cannot be directly connected to source ports. All source ports on a switch belong to the single multicast VLAN. receiver —Configure a port as a receiver port if it is a subscriber port and should only receive multicast data. It does not receive data unless it becomes a member of the multicast group, either statically or through IGMP leave and join messages. Receiver ports cannot belong to the multicast VLAN. The mrouter port should not be configured as the receiver port. Non-MVR port is the default configuration.
Step 7	Switch(config-if)# mvr vlan vlan-id group [ip-address]	(Optional) Statically configures a port to receive multicast traffic sent to the multicast VLAN and the IP multicast address. Receiver ports can also dynamically join multicast groups by using IGMP join and leave messages.
Step 8	Switch(config-if)# mvr immediate	(Optional) Enables the Immediate-Leave feature of MVR on the port.
Step 9	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 10	Switch# show mvr [interface members]	Verifies the configuration.
Step 11	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To return the interface to its default settings, use the **no mvr [type | immediate | vlan vlan-id | group]** interface configuration commands.

This example shows how to configure MVR "source and receiver" access ports:

```
Switch(config)# int fastEthernet 2/2
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 200
Switch(config-if)# mvr type receiver
```

```
Switch(config-if)# exit
Switch(config)# interface fastEthernet 2/3
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 100
Switch(config-if)# mvr type source
```

To verify the configuration, enter the **show mvr** command:

```
Switch# show mvr interface
Port          Type      Mode      VLAN      Status      Immediate Leave
-----
Fa2/2         RECEIVER Access     200        ACTIVE/UP    DISABLED
Fa2/3         SOURCE    Access     100        ACTIVE/UP    DISABLED
```

Dynamic Mode:

```
Switch# show mvr members
MVR Group IP      Status      Members      VLAN      Membership
```

Compatible Mode:

```
Switch# show mvr members
MVR Group IP      Status      Members      VLAN      Membership
-----
225.1.1.1         ACTIVE/UP    Fa2/3        100        Static
```

Configuring MVR on a Trunk Port

To configure MVR on a trunk port, perform these steps:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# mvr	Enables MVR on the switch.
Step 3	Switch(config)# interface <i>interface-id</i>	Enters interface configuration mode, and enters the type and number of the Layer 2 port to configure.
Step 4	Switch(config-if)# switchport mode trunk	Change the interface to trunk mode
Step 5	Switch(config-if)# mvr type receiver	Specifies that the trunk port is an MVR receiver port.
Step 6	Switch(config-if)# mvr vlan <i>mvr-vlan-id</i> receiver vlan <i>receiver-vlan-id</i>	Enables this trunk port to distribute MVR traffic arriving from the MVR VLAN to the VLAN on the trunk identified by the receiver VLAN. Note This command is not accepted unless you first enter the mvr type receiver command.
Step 7	Switch(config-if)# mvr vlan <i>vlan-id</i> group <i>ip-address</i> receiver <i>vlan-id</i>	(Optional) Configures the trunk port to be a static member of the group on the receiver VLAN.
Step 8	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 9	Switch# show mvr [interface members]	Verifies the configuration.
Step 10	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to configure MVR “receiver” VLANs on trunk ports:

```
Switch(config)# interface fastEthernet 2/1
Switch(config-if)# switchport mode trunk
Switch(config-if)# mvr type source
```

```
Switch(config)# interface fastEthernet 2/4
Switch(config-if)# switchport mode trunk
Switch(config-if)# mvr type receiver
Switch(config-if)# mvr vlan 100 receiver vlan 300
```

```
Switch# show mvr interface
```

Port	Type	Mode	VLAN	Status	Immediate Leave
Fa2/1	SOURCE	Trunk	100	ACTIVE/UP	DISABLED
Fa2/2	RECEIVER	Access	200	ACTIVE/UP	DISABLED
Fa2/3	SOURCE	Access	100	ACTIVE/UP	DISABLED
Fa2/4	RECEIVER	Trunk	300	ACTIVE/UP	DISABLED

Compatible Mode

```
Switch# show mvr members
```

MVR Group IP	Status	Members	VLAN	Membership
225.1.1.1	ACTIVE/UP	Fa2/1	100	Static
225.1.1.1	ACTIVE/UP	Fa2/3	100	Static

Dynamic Mode

```
Switch# show mvr members
```

MVR Group IP	Status	Members	VLAN	Membership
--------------	--------	---------	------	------------

Displaying MVR Information

You can display MVR information for the switch or a specified interface. Use the following commands in privileged EXEC mode:

Table 28-3 *Commands for Displaying MVR Information*

show mvr	<p>Displays MVR status:</p> <ul style="list-style-type: none"> • whether MVR is enabled or disabled • the multicast VLAN • the maximum (1500) and current (0 to 1500) number of multicast groups • the query response time • the MVR mode
show mvr interface [<i>interface-id</i>] [members [<i>vlan</i> <i>vlan-id</i>]]	<p>Displays all MVR interfaces and their MVR configurations. Interface specific MVR information can be obtained as well.</p> <p>Type—Receiver or Source</p> <p>Status—One of these:</p> <ul style="list-style-type: none"> – ACTIVE/INACTIVE means the port is part/not part of VLAN. – UP/DOWN means that the port is forwarding/non-forwarding. <p>Immediate Leave—Enabled or Disabled</p> <p>If the members keyword is entered, it displays all multicast group members on this port. If a VLAN is identified, it displays all multicast group members on the VLAN. The VLAN ID range is 1 to 1001 and 1006 to 4094.</p>
show mvr members [<i>ip-address</i>]	This displays all receiver and source ports that are members of all MVR IP multicast groups or the specified MVR IP multicast group address.

The following examples show how to display MVR information for either the switch or an interface on the switch:

```
Switch# show mvr
MVR Running: TRUE
MVR multicast vlan: 10022
MVR Max Multicast Groups: 1500
MVR Current multicast groups: 1
MVR Global query response time: 10 (tenths of sec)
MVR Mode: dynamic
```

```
Switch# show mvr interface
```

Port	Type	Mode	VLAN	Status	Immediate Leave
Fa2/1	SOURCE	Trunk	100	ACTIVE/UP	DISABLED
Fa2/2	RECEIVER	Access	200	ACTIVE/UP	DISABLED
Fa2/3	SOURCE	Access	100	ACTIVE/UP	DISABLED
Fa2/4	RECEIVER	Trunk	300	ACTIVE/UP	DISABLED

```
Switch# show mvr interface
interface
```

```
Switch# show mvr interface fastEthernet 2/2
```

Port	Type	Mode	VLAN	Status	Immediate Leave
Fa2/2	RECEIVER	Access	200	ACTIVE/UP	DISABLED

```
Switch# show mvr interface fastEthernet 2/2 members
MVR Group IP          VLAN          Membership  Status
-----
225.1.1.1             vlan 200      DYNAMIC     ACTIVE/UP

Switch# show mvr interface fastEthernet 2/2 members vlan 200
MVR Group IP          VLAN          Membership  Status
-----
225.1.1.1             vlan 200      DYNAMIC     ACTIVE/UP

Switch# show mvr members
MVR Group IP          Status          Members          VLAN          Membership
-----
225.1.1.1             ACTIVE/UP       Fa2/2            200           Dynamic

Switch# show mvr members 225.1.1.1
MVR Group IP          Status          Members          VLAN          Membership
-----
225.1.1.1             ACTIVE/UP       Fa2/2            200           Dynamic
```

Configuring IGMP Filtering

This section includes the following subsections:

- [Default IGMP Filtering Configuration, page 28-30](#)
- [Configuring IGMP Profiles, page 28-31](#)
- [Applying IGMP Profiles, page 28-32](#)
- [Setting the Maximum Number of IGMP Groups, page 28-33](#)



Note

The IGMP filtering feature works for IGMPv1 and IGMPv2 only.

In some environments (like metropolitan or multiple-dwelling unit (MDU) installations), an administrator might want to control the multicast groups to which a user on a switch port can belong. This allows the administrator to control the distribution of multicast services, such as IP/TV, based on some type of subscription or service plan.

With IGMP filtering, an administrator can apply this type of control. With this feature, you can filter multicast joins on a per-port basis by configuring IP multicast profiles and associating them with individual switch ports. An IGMP profile can contain one or more multicast groups and specifies whether access to the group is permitted or denied. If an IGMP profile denying access to a multicast group is applied to a switch port, the IGMP join report requesting the stream of IP multicast traffic is dropped, and the port is not allowed to receive IP multicast traffic from that group. If the filtering action permits access to the multicast group, the IGMP report from the port is forwarded for normal processing.

IGMP filtering controls only IGMP membership join reports and has no relationship to the function that directs the forwarding of IP multicast traffic.

You can also set the maximum number of IGMP groups that a Layer 2 interface can join with the **ip igmp max-groups** *n* command.

Default IGMP Filtering Configuration

[Table 28-4](#) shows the default IGMP filtering configuration.

Table 28-4 **Default IGMP Filtering Settings**

Feature	Default Setting
IGMP filters	No filtering
IGMP maximum number of IGMP groups	No limit
IGMP profiles	None defined

Configuring IGMP Profiles

To configure an IGMP profile and to enter IGMP profile configuration mode, use the **ip igmp profile** global configuration command. From the IGMP profile configuration mode, you can specify the parameters of the IGMP profile to be used for filtering IGMP join requests from a port. When you are in IGMP profile configuration mode, you can apply these keywords:

- **deny**—Specifies that matching addresses are denied (the default condition).
- **exit**—Exits from igmp-profile configuration mode.
- **no**—Negates a command or sets its defaults.
- **permit**—Specifies that matching addresses are permitted.
- **range**—Specifies a range of IP addresses for the profile. You can enter a single IP address or a range with starting and ending addresses.

By default, no IGMP profiles are configured. When a profile is configured with neither the **permit** nor the **deny** keyword, the default is to deny access to the range of IP addresses.

To create an IGMP profile for a port, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# ip igmp profile <i>profile number</i>	Enters IGMP profile configuration mode and assigns a number to the profile you are configuring. The range is from 1 to 4,294,967,295.
Step 3	Switch(config-igmp-profile)# permit deny	(Optional) Sets the action to permit or deny access to the IP multicast address. If no action is configured, the default for the profile is to deny access.
Step 4	Switch(config-igmp-profile)# range <i>ip multicast address</i>	Enters the IP multicast address or range of IP multicast addresses to which access is being controlled. If entering a range, enter the low IP multicast address, a space, and the high IP multicast address. Use the range command multiple times to enter multiple addresses or ranges of addresses.
Step 5	Switch(config-igmp-profile)# end	Returns to privileged EXEC mode.
Step 6	Switch# show ip igmp profile <i>profile-number</i>	Verifies the profile configuration.
Step 7	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To delete a profile, use the **no ip igmp profile** *profile-number* global configuration command.

To delete an IP multicast address or range of IP multicast addresses, use the **no range ip multicast address** IGMP profile configuration command.

This example shows how to create IGMP profile 4 (allowing access to the single IP multicast address) and how to verify the configuration. If the action were to deny (the default), it does not appear in the output of the **show ip igmp profile** command.

```
Switch# configure terminal
Switch(config)# ip igmp profile 4
Switch(config-igmp-profile)# permit
Switch(config-igmp-profile)# range 229.9.9.0
Switch(config-igmp-profile)# end
Switch# show ip igmp profile 4
IGMP Profile 4
    permit
    range 229.9.9.0 229.9.9.0
```

Applying IGMP Profiles

To control access as defined in an IGMP profile, use the **ip igmp filter** interface configuration command to apply the profile to the appropriate interfaces. You can apply a profile to multiple interfaces, but each interface can only have one profile applied to it.



Note

You can apply IGMP profiles to Layer 2 ports only. You cannot apply IGMP profiles to routed ports (or SVIs) or to ports that belong to an EtherChannel port group.

To apply an IGMP profile to a switch port, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Enters interface configuration mode, and enters the physical interface to configure, for example, fastethernet2/3 . The interface must be a Layer 2 port that does not belong to an EtherChannel port group.
Step 3	Switch(config-if)# ip igmp filter <i>profile number</i>	Applies the specified IGMP profile to the interface. The profile number can be from 1 to 4,294,967,295.
Step 4	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 5	Switch# show running configuration interface <i>interface-id</i>	Verifies the configuration.
Step 6	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To remove a profile from an interface, use the **no ip igmp filter** command.

This example shows how to apply IGMP profile 4 to an interface and to verify the configuration:

```
Switch# configure terminal
Switch(config)# interface fastethernet2/12
Switch(config-if)# ip igmp filter 4
Switch(config-if)# end
```



```

Switch# show running-config interface fastethernet2/12
Building configuration...

Current configuration : 123 bytes
!
interface FastEthernet2/12
 no ip address
 shutdown
 snmp trap link-status
 ip igmp max-groups 25
 ip igmp filter 4
end

```

Setting the Maximum Number of IGMP Groups

You can set the maximum number of IGMP groups that a Layer 2 interface can join by using the **ip igmp max-groups** interface configuration command. Use the **no** form of this command to set the maximum back to the default, which is no limit.



Note

This restriction can be applied to Layer 2 ports only. You cannot set a maximum number of IGMP groups on routed ports (or SVIs) or on ports that belong to an EtherChannel port group.

To apply an IGMP profile on a switch port, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Enters interface configuration mode, and enter the physical interface to configure, for example, gigabitethernet1/1 . The interface must be a Layer 2 port that does not belong to an EtherChannel group.
Step 3	Switch(config-if)# ip igmp max-groups <i>number</i>	Sets the maximum number of IGMP groups that the interface can join. The range is from 0 to 4,294,967,294. By default, no maximum is set. To remove the maximum group limitation and return to the default of no maximum, use the no ip igmp max-groups command.
Step 4	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 5	Switch# show running-configuration interface <i>interface-id</i>	Verifies the configuration.
Step 6	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to limit the number of IGMP groups that an interface can join to 25:

```

Switch# configure terminal
Switch(config)# interface fastethernet2/12
Switch(config-if)# ip igmp max-groups 25
Switch(config-if)# end
Switch# show running-config interface fastethernet2/12
Building configuration...

Current configuration : 123 bytes
!

```

```
interface FastEthernet2/12
no ip address
shutdown
snmp trap link-status
ip igmp max-groups 25
ip igmp filter 4
end
```

Displaying IGMP Filtering Configuration

You can display IGMP profile and maximum group configuration for all interfaces on the switch or for a specified interface.

To display IGMP profiles, perform this task:

Command	Purpose
Switch# show ip igmp profile [<i>profile number</i>]	Displays the specified IGMP profile or all IGMP profiles defined on the switch.

To display interface configuration, perform this task:

Command	Purpose
Switch# show running-configuration [interface interface-id]	Displays the configuration of the specified interface or all interfaces on the switch, including (if configured) the maximum number of IGMP groups to which an interface can belong and the IGMP profile applied to the interface.

This is an example of the **show ip igmp profile** privileged EXEC command when no profile number is entered. All profiles defined on the switch are displayed.

```
Switch# show ip igmp profile
IGMP Profile 3
  range 230.9.9.0 230.9.9.0
IGMP Profile 4
  permit
  range 229.9.9.0 229.255.255.255
```

This is an example of the **show running-config** privileged EXEC command when an interface is specified with IGMP maximum groups configured and IGMP profile 4 has been applied to the interface:

```
Switch# show running-config interface fastethernet2/12
Building configuration...
Current configuration : 123 bytes
!
interface FastEthernet2/12
no ip address
shutdown
snmp trap link-status
ip igmp max-groups 25
ip igmp filter 4
end
```



EIGRP Stub Routing

This feature module describes the EIGRP Stub Routing feature and includes the following sections:

- [Feature Overview](#)
- [Benefits](#)
- [Restrictions](#)
- [Related Features and Technologies](#)
- [Supported Platforms](#)
- [Supported Standards, MIBs, and RFCs](#)
- [Configuration Tasks](#)
- [Monitoring and Maintaining EIGRP Stub Routing](#)
- [Configuration Examples](#)

Feature Overview

The Enhanced Interior Gateway Routing Protocol (EIGRP) Stub Routing feature improves network stability, reduces resource utilization, and simplifies stub router configuration.

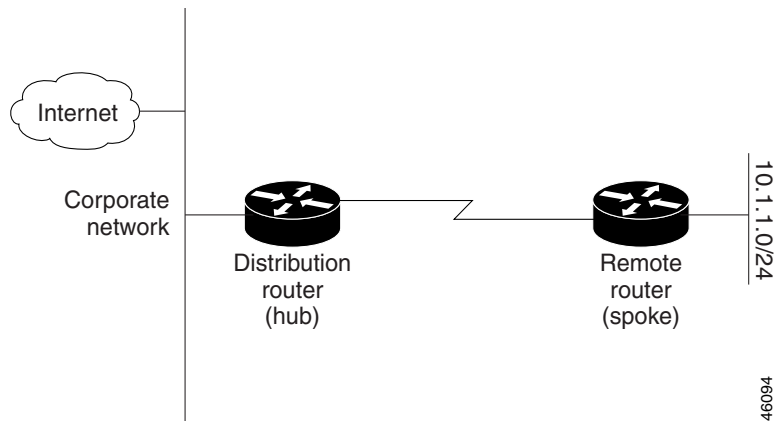
Stub routing is commonly used in a hub and spoke network topology. In a hub and spoke network, one or more end (stub) networks are connected to a remote router (the spoke) that is connected to one or more distribution routers (the hub). The remote router is adjacent only to one or more distribution routers. The only route for IP traffic to follow into the remote router is through a distribution router. This type of configuration is commonly used in WAN topologies where the distribution router is directly connected to a WAN. The distribution router can be connected to many more remote routers. Often, the distribution router will be connected to 100 or more remote routers. In a hub and spoke topology, the remote router must forward all nonlocal traffic to a distribution router, so it becomes unnecessary for the remote router to hold a complete routing table. Generally, the distribution router need not send anything more than a default route to the remote router.

When using the EIGRP Stub Routing feature, you need to configure the distribution and remote routers to use EIGRP, and to configure only the remote router as a stub. Only specified routes are propagated from the remote (stub) router. The router responds to queries for summaries, connected routes, redistributed static routes, external routes, and internal routes with the message “inaccessible.” A router that is configured as a stub will send a special peer information packet to all neighboring routers to report its status as a stub router.

Any neighbor that receives a packet informing it of the stub status will not query the stub router for any routes, and a router that has a stub peer will not query that peer. The stub router will depend on the distribution router to send the proper updates to all peers.

Figure 1 shows a simple spoke and hub configuration.

Figure 1 Simple Spoke and Hub Network



The stub feature by itself does not prevent routes from being advertised to the remote router. In the example in Figure 1, the remote router can access the corporate network and the Internet through the distribution router only. Having a full route table on the remote router, in this example, would have no functional purpose because the path to the corporate network and the Internet would always be through the distribution router. The larger route table would only increase the amount of memory required by the remote router.

Bandwidth and memory can further be conserved by summarizing and filtering routes on the distribution router. The remote router need not receive routes that have been learned from other networks because the remote router must send all nonlocal traffic, regardless of destination, to the distribution router. If a true stub network is desired, the distribution router should be configured to send only a default route to the remote router. The EIGRP Stub Routing feature does not automatically enable summarization on the distribution router. In most cases, the network administrator will need to configure summarization on the distribution routers.



Note

When configuring the distribution router to send only a default route to the remote router, you must use the **ip classless** command on the remote router. By default, the **ip classless** command is enabled in all Cisco IOS images that support the EIGRP Stub Routing feature.

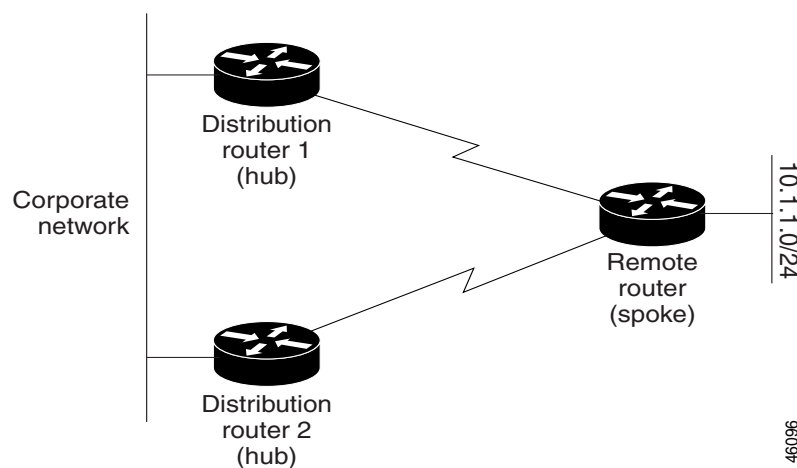
Without the stub feature, even after the routes that are sent from the distribution router to the remote router have been filtered or summarized, a problem might occur. If a route is lost somewhere in the corporate network, EIGRP could send a query to the distribution router, which in turn will send a query to the remote router even if routes are being summarized. If there is a problem communicating over the WAN link between the distribution router and the remote router, an EIGRP stuck in active (SIA) condition could occur and cause instability elsewhere in the network. The EIGRP Stub Routing feature allows a network administrator to prevent queries from being sent to the remote router.

Dual-Homed Remote

In addition to a simple hub and spoke network where a remote router is connected to a single distribution router, the remote router can be dual-homed to two or more distribution routers. This configuration adds redundancy and introduces unique issues, and the stub feature helps to address some of these issues.

A dual-homed remote will have two or more distribution (hub) routers. However, the principles of stub routing are the same as they are with a hub and spoke topology. [Figure 2](#) shows a common dual-homed remote topology with one remote router, but 100 or more routers could be connected on the same interfaces on distribution router 1 and distribution router 2. The remote router will use the best route to reach its destination. If distribution router 1 experiences a failure, the remote router can still use distribution router 2 to reach the corporate network.

Figure 2 Simple Dual-Homed Remote Topology



[Figure 2](#) shows a simple dual-homed remote with one remote router and two distribution routers. Both distribution routers maintain routes to the corporate network and stub network (10.1.1.0/24).

Dual-homed routing can introduce instability into an EIGRP network. In [Figure 3](#), distribution router 1 is directly connected to network 10.3.1.0/24. If summarization or filtering is applied on distribution router 1, the router will advertise network 10.3.1.0/24 to all of its directly connected EIGRP neighbors (distribution router 2 and the remote router).

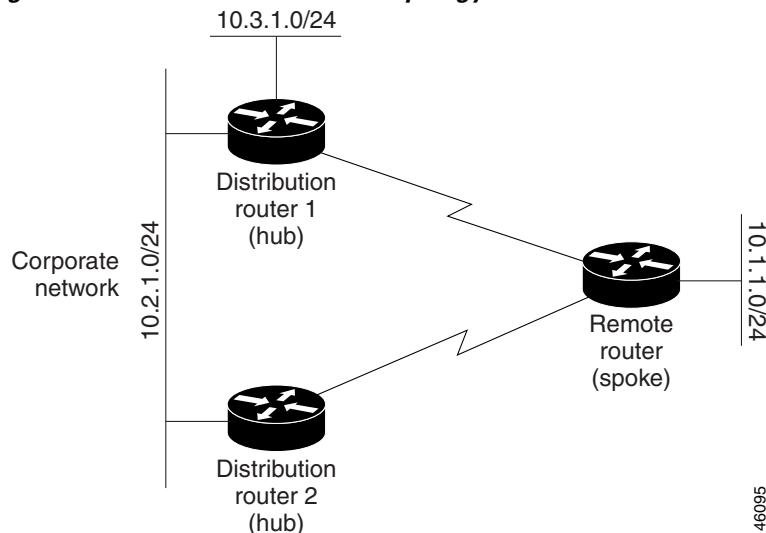
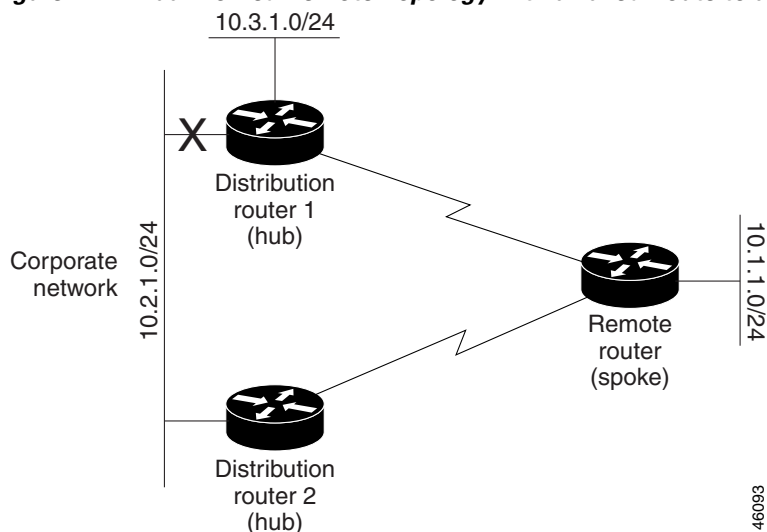
Figure 3 *Dual-Homed Remote Topology Where Distribution Router 1 Is Connected to 2 Networks*

Figure 3 shows a simple dual-homed remote topology where distribution router 1 is connected to both network 10.3.1.0/24 and network 10.2.1.0/24.

If the 10.2.1.0/24 link between distribution router 1 and distribution router 2 has failed, the lowest cost path to network 10.3.1.0/24 from distribution router 2 is through the remote router (see Figure 4). This route is not desirable because the traffic that was previously traveling across the corporate network 10.2.1.0/24 would now be sent across a much lower bandwidth connection. The overutilization of the lower bandwidth WAN connection can cause a number of problems that might affect the entire corporate network. The use of the lower bandwidth route that passes through the remote router might cause WAN EIGRP distribution routers to be dropped. Serial lines on distribution and remote routers could also be dropped, and EIGRP SIA errors on the distribution and core routers could occur.

Figure 4 *Dual-Homed Remote Topology with a Failed Route to a Distribution Router*

It is not desirable for traffic from distribution router 2 to travel through any remote router in order to reach network 10.3.1.0/24. If the links are sized to handle the load, it would be acceptable to use one of the backup routes. However, most networks of this type have remote routers located at remote offices with relatively slow links. This problem can be prevented if proper summarization is configured on the distribution router and remote router.

It is typically undesirable for traffic from a distribution router to use a remote router as a transit path. A typical connection from a distribution router to a remote router would have much less bandwidth than a connection at the network core. Attempting to use a remote router with a limited bandwidth connection as a transit path would generally produce excessive congestion to the remote router. The EIGRP Stub Routing feature can prevent this problem by preventing the remote router from advertising core routes back to distribution routers. Routes learned by the remote router from distribution router 1 will not be advertised to distribution router 2. Since the remote router will not advertise core routes to distribution router 2, the distribution router will not use the remote router as a transit for traffic destined for the network core.

Benefits

Greater Network Stability

In the event of network instability, the EIGRP Stub Routing feature prevents EIGRP queries from being sent over limited bandwidth links to nontransit routers. Instead, distribution routers to which the stub router is connected answers the query on behalf of the stub router. This feature greatly reduces the chance of further network instability due to congested or problematic WAN links.

Simplified Stub Router Configuration

The EIGRP Stub Routing feature simplifies the configuration and maintenance of hub and spoke networks. When stub routing is enabled in dual-homed remote configurations, it is no longer necessary to configure filtering on remote routers to prevent those remote routers from appearing as transit paths to the hub routers.

Restrictions

Supports Only Stub Routers

This feature should only be used on stub routers. A stub router is defined as a router connected to the network core or distribution layer through which core transit traffic should not flow. A stub router should not have any EIGRP neighbors other than distribution routers. Ignoring this restriction will cause undesirable behavior.

Multi-Access Interfaces

Multi-access interfaces, such as ATM, Ethernet, Frame Relay, ISDN PRI, and X.25, are supported by the EIGRP Stub Routing feature only when all routers on that interface, except the hub, are configured as stub routers.

Related Features and Technologies

The EIGRP Stub Routing feature is an extension of the Enhanced Interior Gateway Routing Protocol (EIGRP). For more information about configuring EIGRP and configuring route summarization and filtering, refer to the “Configuring EIGRP” chapter of the *Cisco IOS Release 12.0 Network Protocols Configuration Guide, Part 1* and *Cisco IOS Release 12.0 Network Protocols Command Reference, Part 1*.

-

Supported Platforms

This feature is supported on all platforms that support EIGRP in Cisco IOS Release 12.0(15)S, including the following platforms:

- Cisco 7200 series
- Cisco 7500 series
- Cisco 12000 series

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that are supported by specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions at <http://www.cisco.com/register>.

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

<http://www.cisco.com/go/fn>

Availability of Cisco IOS Software Images

Platform support for particular Cisco IOS software releases is dependent on the availability of the software images for those platforms. Software images for some platforms may be deferred, delayed, or changed without prior notice. For updated information about platform support and availability of software images for each Cisco IOS software release, refer to the online release notes or, if supported, Cisco Feature Navigator.

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

No new or modified MIBs are supported by this feature.

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:

<http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml>

RFCs

No new or modified MIBs are supported by this feature.

Configuration Tasks

See the following sections for configuration tasks for the EIGRP Stub Routing feature. Each task in the list is identified as either optional or required:

- [Configuring EIGRP Stub Routing](#)(required)
- [Verifying EIGRP Stub Routing](#)(optional)

Configuring EIGRP Stub Routing

To configure a remote or spoke router for EIGRP stub routing, use the following commands beginning in router configuration mode:

	Command	Purpose
Step 1	router(config)# router eigrp <i>as-number</i>	Configures a remote or distribution router to run an EIGRP process.
Step 2	router(config-router)# network <i>network-number</i>	Specifies the network address of the EIGRP distribution router.
Step 3	router(config-router)# eigrp stub [receive-only connected static summary]	Configures a remote router as an EIGRP stub router.

Verifying EIGRP Stub Routing

To verify that a remote router has been configured as a stub router with the EIGRP Stub Routing feature, use the **show ip eigrp neighbor detail** command from the distribution router in privileged EXEC mode. The last line of the output will show the stub status of the remote or spoke router. The following example output is from the **show ip eigrp neighbor detail** command:

```
router# show ip eigrp neighbor detail
IP-EIGRP neighbors for process 1
H   Address                  Interface    Hold Uptime    SRTT    RTO  Q  Seq Type
                               (sec)          (ms)          Cnt Num
0   10.1.1.2                  Se3/1       11 00:00:59    1   4500  0   7
Version 12.1/1.2, Retrans: 2, Retries: 0
Stub Peer Advertising ( CONNECTED SUMMARY ) Routes
```

Monitoring and Maintaining EIGRP Stub Routing

To enable EIGRP stub packet debugging, use the following command in privileged EXEC mode:

Command	Purpose
Router# debug eigrp packet stub	Displays debug information about the stub status of peer routers.

Configuration Examples

A router that is configured as a stub with the **eigrp stub** command shares connected and summary routing information with all neighbor routers by default. Four optional keywords can be used with the **eigrp stub** command to modify this behavior:

- **receive-only**
- **connected**
- **static**
- **summary**

This section provides configuration examples for all forms of the **eigrp stub** command. The **eigrp stub** command can be modified with several options, and these options can be used in any combination except for the **receive-only** keyword. The **receive-only** keyword will restrict the router from sharing any of its routes with any other router in that EIGRP autonomous system, and the **receive-only** keyword will not permit any other option to be specified because it prevents any type of route from being sent. The three other optional keywords (**connected**, **static**, and **summary**) can be used in any combination but cannot be used with the **receive-only** keyword. If any of these three keywords is used individually with the **eigrp stub** command, connected and summary routes will not be sent automatically.

The **connected** keyword will permit the EIGRP Stub Routing feature to send connected routes. If the connected routes are not covered by a network statement, it may be necessary to redistribute connected routes with the **redistribute connected** command under the EIGRP process. This option is enabled by default.

The **static** keyword will permit the EIGRP Stub Routing feature to send static routes. Without the configuration of this option, EIGRP will not send any static routes, including internal static routes that normally would be automatically redistributed. It will still be necessary to redistribute static routes with the **redistribute static** command.

The **summary** keyword will permit the EIGRP Stub Routing feature to send summary routes. Summary routes can be created manually with the **summary address** command or automatically at a major network border router with the **auto-summary** command enabled. This option is enabled by default.

In the following example, the **eigrp stub** command is used to configure the router as a stub that advertises connected and summary routes:

```
router eigrp 1
network 10.0.0.0
eigrp stub
```

In the following example, the **eigrp stub** command is issued with the **connected** and **static** keywords to configure the router as a stub that advertises connected and static routes (sending summary routes will not be permitted):

```
router eigrp 1
network 10.0.0.0
eigrp stub connected static
```

In the following example, the **eigrp stub** command is issued with the **receive-only** keyword to configure the router as a receive-only neighbor (Connected, summary, and static routes will not be sent):

```
router eigrp 1
network 10.0.0.0 eigrp
eigrp stub receive-only
```




Configuring IPv6 Multicast Listener Discovery Snooping

Use Multicast Listener Discovery (MLD) snooping to enable efficient distribution of IP version 6 (IPv6) multicast data to clients and routers in a switched network on the Catalyst 4500 series switch.

This chapter includes these sections:

- [About MLD Snooping, page 30-1](#)
- [Configuring IPv6 MLD Snooping, page 30-5](#)
- [Displaying MLD Snooping Information, page 30-10](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About MLD Snooping

In IP version 4 (IPv4), Layer 2 switches can use Internet Group Management Protocol (IGMP) snooping to limit the flooding of multicast traffic by dynamically configuring Layer 2 interfaces so that multicast traffic is forwarded to only those interfaces associated with IP multicast devices. In IPv6, MLD snooping performs a similar function. With MLD snooping, IPv6 multicast data is selectively forwarded to a list of ports that want to receive the data, instead of being flooded to all ports in a VLAN. This list is constructed by snooping IPv6 multicast control packets.

MLD is a protocol used by IPv6 multicast routers to discover the presence of multicast listeners (nodes that want to receive IPv6 multicast packets) on its directly attached links and to discover which multicast packets are of interest to neighboring nodes. MLD is derived from IGMP; MLD version 1 (MLDv1) is equivalent to IGMPv2 and MLD version 2 (MLDv2) is equivalent to IGMPv3. MLD is a subprotocol of Internet Control Message Protocol version 6 (ICMPv6), and MLD messages are a subset of ICMPv6 messages, identified in IPv6 packets by a preceding Next Header value of 58.

The switch supports two versions of MLD snooping:

- MLDv1 snooping detects MLDv1 control packets and sets up traffic bridging based on IPv6 destination multicast addresses.
- MLDv2 basic snooping (MBSS) uses MLDv2 control packets to set up traffic forwarding based on IPv6 destination multicast addresses.

The switch can snoop on both MLDv1 and MLDv2 protocol packets and bridge IPv6 multicast data based on destination IPv6 multicast addresses.



Note

The switch does not support MLDv2 enhanced snooping (MESS), which sets up IPv6 source and destination multicast address-based forwarding.

MLD snooping can be enabled or disabled globally or per-VLAN. When MLD snooping is enabled, a per-VLAN IPv6 multicast MAC address table is constructed in software and a per-VLAN IPv6 multicast address table is constructed in software and hardware. The switch then performs IPv6 multicast-address based bridging in hardware.

These sections describe some parameters of IPv6 MLD snooping:

- [MLD Messages, page 30-2](#)
- [MLD Queries, page 30-3](#)
- [Multicast Client Aging, page 30-3](#)
- [Multicast Router Discovery, page 30-3](#)
- [MLD Reports, page 30-4](#)
- [MLD Done Messages and Immediate-Leave, page 30-4](#)
- [Topology Change Notification Processing, page 30-4](#)

MLD Messages

MLDv1 supports three types of messages:

- Listener Queries are the equivalent of IGMPv2 queries and are either General Queries or Multicast-Address-Specific Queries (MASQs).
- Multicast Listener Reports are the equivalent of IGMPv2 reports.
- Multicast Listener Done messages are the equivalent of IGMPv2 leave messages.

MLDv2 supports MLDv2 queries and reports, as well as MLDv1 Report and Done messages.

Message timers and state transitions resulting from messages being sent or received are the same as those of IGMPv2 messages. MLD messages that do not have valid link-local IPv6 source addresses are ignored by MLD routers and switches.

MLD Queries

The switch sends out MLD queries, constructs an IPv6 multicast address database, and generates MLD group-specific and MLD group-and-source-specific queries in response to MLD Done messages. The switch also supports report suppression, report proxying, Immediate-Leave functionality, and static IPv6 multicast MAC-address configuration.

When MLD snooping is disabled, all MLD queries are flooded in the ingress VLAN.

When MLD snooping is enabled, received MLD queries are flooded in the ingress VLAN, and a copy of the query is sent to the CPU for processing. From the received query, MLD snooping builds the IPv6 multicast address database. It detects multicast router ports, maintains timers, sets report response time, learns the querier IP source address for the VLAN, learns the querier port in the VLAN, and maintains multicast-address aging.

When a group exists in the MLD snooping database, the switch responds to a group-specific query by sending an MLDv1 report. When the group is unknown, the group-specific query is flooded to the ingress VLAN.

When a host wants to leave a multicast group, it can send out an MLD Done message (equivalent to IGMP Leave message). When the switch receives an MLDv1 Done message, if Immediate-Leave is not enabled, the switch sends an MASQ to the port from which the message was received to determine if other devices connected to the port should remain in the multicast group.

Multicast Client Aging

You can configure port membership removal from addresses based on the number of queries. A port is removed from membership to an address only when there are no reports to the address on the port for the configured number of queries. The default number is 2.

Multicast Router Discovery

MLD snooping performs multicast router discovery with these characteristics:

- Ports configured by a user never age out.
- Dynamic port learning results from MLDv1 snooping queries and IPv6 PIMv2 packets.
- If multiple routers exist on the same Layer 2 interface, MLD snooping tracks a single multicast router on the port (the router that most recently sent a router control packet).
- Dynamic multicast router port aging is based on a default timer of 5 minutes; the multicast router is deleted from the router port list if no control packet is received on the port for 5 minutes.
- IPv6 multicast router discovery only takes place when MLD snooping is enabled on the switch.
- Received IPv6 multicast router control packets are always flooded to the ingress VLAN, whether MLD snooping is enabled on the switch.
- After the discovery of the first IPv6 multicast router port, unknown IPv6 multicast data is forwarded only to the discovered router ports (before that time, all IPv6 multicast data is flooded to the ingress VLAN).

MLD Reports

The processing of MLDv1 join messages is essentially the same as with IGMPv2. When no IPv6 multicast routers are detected in a VLAN, reports are not processed or forwarded from the switch. When IPv6 multicast routers are detected and an MLDv1 report is received, an IPv6 multicast group address and an IPv6 multicast MAC address are entered in the VLAN MLD database. All IPv6 multicast traffic to the group within the VLAN is then forwarded using this address. When MLD snooping is disabled, reports are flooded in the ingress VLAN.

When MLD snooping is enabled, MLD report suppression, called listener message suppression, is automatically enabled. With report suppression, the switch forwards the first MLDv1 report received by a group to IPv6 multicast routers; subsequent reports for the group are not sent to the routers. When MLD snooping is disabled, report suppression is disabled, and all MLDv1 reports are flooded to the ingress VLAN.

The switch also supports MLDv1 proxy reporting. When an MLDv1 MASQ is received, the switch responds with MLDv1 reports for the address on which the query arrived if the group exists in the switch on another port and if the port on which the query arrived is not the last member port for the address.

MLD Done Messages and Immediate-Leave

When the Immediate-Leave feature is enabled and a host sends an MLDv1 Done message (equivalent to an IGMP leave message), the port on which the Done message was received is immediately deleted from the group. You enable Immediate-Leave on VLANs and (as with IGMP snooping), you should only use the feature on VLANs where a single host is connected to the port. If the port was the last member of a group, the group is also deleted, and the leave information is forwarded to the detected IPv6 multicast routers.

When Immediate Leave is not enabled in a VLAN (the case when multiple clients for a group exist on the same port) and a Done message is received on a port, an MASQ is generated on that port. The user can control when a port membership is removed for an existing address in terms of the number of MASQs. A port is removed from membership to an address when there are no MLDv1 reports to the address on the port for the configured number of queries.

The number of MASQs generated is configured by using the **ipv6 mld snooping last-listener-query count** global configuration command. The default number is 2.

The MASQ is sent to the IPv6 multicast address for which the Done message was sent. If no reports are sent to the IPv6 multicast address specified in the MASQ during the switch maximum response time, the port on which the MASQ was sent is deleted from the IPv6 multicast address database. The maximum response time is the time configured by using the **ipv6 mld snooping last-listener-query-interval** global configuration command. If the deleted port is the last member of the multicast address, the multicast address is also deleted, and the switch sends the address leave information to all detected multicast routers.

Topology Change Notification Processing

When topology change notification (TCN) solicitation is enabled by using the **ipv6 mld snooping tcn query solicit** global configuration command, MLDv1 snooping sets the VLAN to flood all IPv6 multicast traffic with a configured number of MLDv1 queries before it begins sending multicast data only to selected ports. You set this value by using the **ipv6 mld snooping tcn flood query count** global

configuration command. The default is to send two queries. The switch also generates MLDv1 global Done messages with valid link-local IPv6 source addresses when the switch becomes the STP root in the VLAN or when it is configured by the user. This process is similar to that in IGMP snooping.

Configuring IPv6 MLD Snooping

These sections describe how to configure IPv6 MLD snooping:

- [Default MLD Snooping Configuration, page 30-5](#)
- [MLD Snooping Configuration Guidelines, page 30-6](#)
- [Enabling or Disabling MLD Snooping, page 30-6](#)
- [Configuring a Static Multicast Group, page 30-7](#)
- [Configuring a Multicast Router Port, page 30-7](#)
- [Enabling MLD Immediate Leave, page 30-8](#)
- [Configuring MLD Snooping Queries, page 30-9](#)
- [Disabling MLD Listener Message Suppression, page 30-10](#)

Default MLD Snooping Configuration

[Table 30-1](#) shows the default MLD snooping configuration.

Table 30-1 *Default MLD Snooping Configuration*

Feature	Default Setting
MLD snooping (Global)	Disabled.
MLD snooping (per VLAN)	Enabled. MLD snooping must be globally enabled for VLAN MLD snooping to take place.
IPv6 Multicast addresses	None configured.
IPv6 Multicast router ports	None configured.
MLD snooping Immediate Leave	Disabled.
MLD snooping robustness variable	Global: 2; per-VLAN: 0. Note The VLAN value overrides the global setting. When the VLAN value is 0, the VLAN uses the global count.
Last listener query count	Global: 2; per-VLAN: 0. Note The VLAN value overrides the global setting. When the VLAN value is 0, the VLAN uses the global count.
Last listener query interval	Global: 1000 (1 second); VLAN: 0. Note The VLAN value overrides the global setting. When the VLAN value is 0, the VLAN uses the global interval.
TCN query solicit	Disabled.
TCN query count	2.
MLD listener suppression	Disabled.

MLD Snooping Configuration Guidelines

When configuring MLD snooping, consider these guidelines:

- You can configure MLD snooping characteristics at any time, but you must globally enable MLD snooping by using the **ipv6 mld snooping** global configuration command for the configuration to take effect.
- MLD snooping and IGMP snooping act independently of each other. You can enable both features at the same time on the switch. The total number of IPv4 and IPv6 multicast groups entries that can coexist on the Catalyst 4500 series switch is limited to 16384.
- The supervisor engine with 512 MB of memory supports about 11000 MLD Snooping multicast groups. A supervisor engine with 1 GB memory supports 16384 MLD Snooping multicast groups.

Enabling or Disabling MLD Snooping

By default, IPv6 MLD snooping is globally disabled on the switch and enabled on all VLANs. When MLD snooping is globally disabled, it is also disabled on all VLANs. When you globally enable MLD snooping, the VLAN configuration overrides the global configuration. MLD snooping is enabled only on VLAN interfaces in the default state (enabled).

You can enable and disable MLD snooping on a per-VLAN basis, but if you globally disable MLD snooping, it is disabled in all VLANs. If global snooping is enabled, you can enable or disable VLAN snooping.

To globally enable MLD snooping on the switch, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# ipv6 mld snooping	Globally enables MLD snooping on the switch.
Step 3	Switch(config)# end	Returns to privileged EXEC mode.
Step 4	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To globally disable MLD snooping on the switch, use the **no ipv6 mld snooping** global configuration command.

To enable MLD snooping on a VLAN, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# ipv6 mld snooping	Globally enables MLD snooping on the switch.
Step 3	Switch(config)# ipv6 mld snooping vlan <i>vlan-id</i>	Enables MLD snooping on the VLAN. The VLAN ID range is 1 to 1001 and 1006 to 4094. Note MLD snooping must be globally enabled for VLAN snooping to be enabled.
Step 4	Switch(config)# end	Returns to privileged EXEC mode.
Step 5	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To disable MLD snooping on a VLAN interface, use the **no ipv6 mld snooping vlan *vlan-id*** global configuration command for the specified VLAN number.

Configuring a Static Multicast Group

Hosts or Layer 2 ports normally join multicast groups dynamically, but you can also statically configure an IPv6 multicast address and member ports for a VLAN.

To add a Layer 2 port as a member of a multicast group, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode
Step 2	Switch(config)# ipv6 mld snooping vlan <i>vlan-id</i> static <i>ipv6_multicast_address</i> interface <i>interface-id</i>	Statically configures a multicast group with a Layer 2 port as a member of a multicast group: <ul style="list-style-type: none"> <i>vlan-id</i> is the multicast group VLAN ID. The VLAN ID range is 1 to 1001 and 1006 to 4094. <i>ipv6_multicast_address</i> is the 128-bit group IPv6 address. The address must be in the form specified in RFC 2373. <i>interface-id</i> is the member port. It can be a physical interface or a port channel (1 to 64).
Step 3	Switch(config)# end	Returns to privileged EXEC mode.
Step 4	Switch# show mac-address-table multicast mld-snooping	Verifies the static member port and the IPv6 address.
Step 5	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To remove a Layer 2 port from the multicast group, use the **no ipv6 mld snooping vlan *vlan-id* static *mac-address* interface *interface-id*** global configuration command. If all member ports are removed from a group, the group is deleted.

This example shows how to statically configure an IPv6 MAC address:

```
Switch# configure terminal
Switch(config)# ipv6 mld snooping vlan 2 static 3333.0000.0003 interface
gigabitethernet1/1
Switch(config)# end
```

Configuring a Multicast Router Port

Although MLD snooping learns about router ports through MLD queries and PIMv6 queries, you can also use the command-line interface (CLI) to add a multicast router port to a VLAN. To add a multicast router port (add a static connection to a multicast router), use the **ipv6 mld snooping vlan *mrouter*** global configuration command on the switch.



Note

Static connections to multicast routers are supported only on switch ports.

To add a multicast router port to a VLAN, follow these steps:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# ipv6 mld snooping vlan <i>vlan-id mrouter interface interface-id</i>	Specifies the multicast router VLAN ID, and specify the interface to the multicast router. <ul style="list-style-type: none"> • The VLAN ID range is 1 to 1001 and 1006 to 4094. • The interface can be a physical interface or a port channel. The port-channel range is 1 to 64.
Step 3	Switch(config)# end	Returns to privileged EXEC mode.
Step 4	Switch# show ipv6 mld snooping mrouter <i>[vlan vlan-id]</i>	Verifies that IPv6 MLD snooping is enabled on the VLAN interface.
Step 5	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To remove a multicast router port from the VLAN, use the **no ipv6 mld snooping vlan *vlan-id* mrouter interface *interface-id*** global configuration command.

This example shows how to add a multicast router port to VLAN 200:

```
Switch# configure terminal
Switch(config)# ipv6 mld snooping vlan 200 mrouter interface gigabitethernet0/2
Switch(config)# exit
```

Enabling MLD Immediate Leave

When you enable MLDv1 Immediate Leave, the switch immediately removes a port from a multicast group when it detects an MLD Done message on that port. You should only use the Immediate Leave feature when there is a single receiver present on every port in the VLAN. When multiple clients exist for a multicast group on the same port, do not enable Immediate-Leave in a VLAN.

To enable MLDv1 Immediate Leave, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# ipv6 mld snooping vlan <i>vlan-id</i> immediate-leave	Enables MLD Immediate Leave on the VLAN interface.
Step 3	Switch(config)# end	Returns to privileged EXEC mode.
Step 4	Switch# show ipv6 mld snooping vlan <i>vlan-id</i>	Verifies that Immediate Leave is enabled on the VLAN interface.
Step 5	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To disable MLD Immediate Leave on a VLAN, use the **no ipv6 mld snooping vlan *vlan-id* immediate-leave** global configuration command.

This example shows how to enable MLD Immediate Leave on VLAN 130:

```
Switch# configure terminal
Switch(config)# ipv6 mld snooping vlan 130 immediate-leave
Switch(config)# exit
```

Configuring MLD Snooping Queries

When Immediate Leave is not enabled and a port receives an MLD Done message, the switch generates MASQs on the port and sends them to the IPv6 multicast address for which the Done message was sent. You can optionally configure the number of MASQs that are sent and the length of time the switch waits for a response before deleting the port from the multicast group.

To configure MLD snooping query characteristics for the switch or for a VLAN, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# ipv6 mld snooping robustness-variable <i>value</i>	(Optional) Sets the number of queries that are sent before the switch deletes a listener (port) that does not respond to a general query. The range is 1 to 3; the default is 2.
Step 3	Switch(config)# ipv6 mld snooping vlan <i>vlan-id</i> robustness-variable <i>value</i>	(Optional) Sets the robustness variable on a VLAN basis, which determines the number of general queries that MLD snooping sends before aging out a multicast address when there is no MLD report response. The range is 1 to 3; the default is 0. When set to 0, the number used is the global robustness variable value.
Step 4	Switch(config)# ipv6 mld snooping last-listener-query-count <i>count</i>	(Optional) Sets the number of MASQs that the switch sends before aging out an MLD client. The range is 1 to 7; the default is 2. The queries are sent 1 second apart.
Step 5	Switch(config)# ipv6 mld snooping vlan <i>vlan-id</i> last-listener-query-count <i>count</i>	(Optional) Sets the last-listener query count on a VLAN basis. This value overrides the value configured globally. The range is 1 to 7; the default is 0. When set to 0, the global count value is used. Queries are sent 1 second apart.
Step 6	Switch(config)# ipv6 mld snooping last-listener-query-interval <i>interval</i>	(Optional) Sets the maximum response time that the switch waits after sending out a MASQ before deleting a port from the multicast group. The range is 100 to 32,768 thousands of a second. The default is 1000 (1 second).
Step 7	Switch(config)# ipv6 mld snooping vlan <i>vlan-id</i> last-listener-query-interval <i>interval</i>	(Optional) Sets the last-listener query interval on a VLAN basis. This value overrides the value configured globally. The range is 0 to 32,768 thousands of a second. The default is 0. When set to 0, the global last-listener query interval is used.
Step 8	Switch(config)# ipv6 mld snooping tcn query solicit	(Optional) Enables topology change notification (TCN) solicitation, which means that VLANs flood all IPv6 multicast traffic for the configured number of queries before sending multicast data to only those ports requesting to receive it. The default is for TCN to be disabled.
Step 9	Switch(config)# ipv6 mld snooping tcn flood query count <i>count</i>	(Optional) When TCN is enabled, specifies the number of TCN queries to be sent. The range is from 1 to 10; the default is 2.
Step 10	Switch(config)# end	Returns to privileged EXEC mode.
Step 11	Switch# show ipv6 mld snooping querier [<i>vlan vlan-id</i>]	(Optional) Verifies that the MLD snooping querier information for the switch or for the VLAN.
Step 12	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to set the MLD snooping global robustness variable to 3:

```
Switch# configure terminal
Switch(config)# ipv6 mld snooping robustness-variable 3
Switch(config)# exit
```

This example shows how to set the MLD snooping last-listener query count for a VLAN to 3:

```
Switch# configure terminal
Switch(config)# ipv6 mld snooping vlan 200 last-listener-query-count 3
Switch(config)# exit
```

This example shows how to set the MLD snooping last-listener query interval (maximum response time) to 2000 (2 seconds):

```
Switch# configure terminal
Switch(config)# ipv6 mld snooping last-listener-query-interval 2000
Switch(config)# exit
```

Disabling MLD Listener Message Suppression

MLD snooping listener message suppression is enabled by default. When it is enabled, the switch forwards only one MLD report per multicast router query. When message suppression is disabled, multiple MLD reports could be forwarded to the multicast routers.

To disable MLD listener message suppression, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# no ipv6 mld snooping listener-message-suppression	Disables MLD message suppression.
Step 3	Switch(config)# end	Returns to privileged EXEC mode.
Step 4	Switch# show ipv6 mld snooping	Verifies that IPv6 MLD snooping report suppression is disabled.
Step 5	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To reenabling MLD message suppression, use the **ipv6 mld snooping listener-message-suppression** global configuration command.

Displaying MLD Snooping Information

You can display MLD snooping information for dynamically learned and statically configured router ports and VLAN interfaces. You can also display MAC address multicast entries for a VLAN configured for MLD snooping.

To display MLD snooping information, use one or more of the privileged EXEC commands in [Table 30-2](#).

Table 30-2 **Commands for Displaying MLD Snooping Information**

Command	Purpose
<code>show ipv6 mld snooping [vlan <i>vlan-id</i>]</code>	Displays the MLD snooping configuration information for all VLANs on the switch or for a specified VLAN. (Optional) Enter vlan <i>vlan-id</i> to display information for a single VLAN. The VLAN ID range is 1 to 1001 and 1006 to 4094.
<code>show ipv6 mld snooping mrouter [vlan <i>vlan-id</i>]</code>	Displays information on dynamically learned and manually configured multicast router interfaces. When you enable MLD snooping, the switch automatically learns the interface to which a multicast router is connected. These are dynamically learned interfaces. (Optional) Enter vlan <i>vlan-id</i> to display information for a single VLAN. The VLAN ID range is 1 to 1001 and 1006 to 4094.
<code>show ipv6 mld snooping querier [vlan <i>vlan-id</i>]</code>	Displays information about the IPv6 address and incoming port for the most-recently received MLD query messages in the VLAN. (Optional) Enter vlan <i>vlan-id</i> to display information for a single VLAN. The VLAN ID range is 1 to 1001 and 1006 to 4094.

■ Displaying MLD Snooping Information



Configuring 802.1Q Tunneling, VLAN Mapping, and Layer 2 Protocol Tunneling

Virtual private networks (VPNs) provide enterprise-scale connectivity on a shared infrastructure, often Ethernet-based, with the same security, prioritization, reliability, and manageability requirements of private networks. Tunneling is a feature designed for service providers who carry traffic of multiple customers across their networks and who are required to maintain the VLAN and Layer 2 protocol configurations of each customer without impacting the traffic of other customers. This chapter describes how to configure 802.1Q, Layer 2 protocol tunneling, and VLAN mapping (or VLAN ID translation) on a Catalyst 4500 series switch.

This chapter contains these sections:

- [About 802.1Q Tunneling, page 31-1](#)
- [Configuring 802.1Q Tunneling, page 31-3](#)
- [About VLAN Mapping, page 31-6](#)
- [Configuring VLAN Mapping, page 31-9](#)
- [About Layer 2 Protocol Tunneling, page 31-13](#)
- [Configuring Layer 2 Protocol Tunneling, page 31-15](#)
- [Monitoring and Maintaining Tunneling Status, page 31-23](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About 802.1Q Tunneling

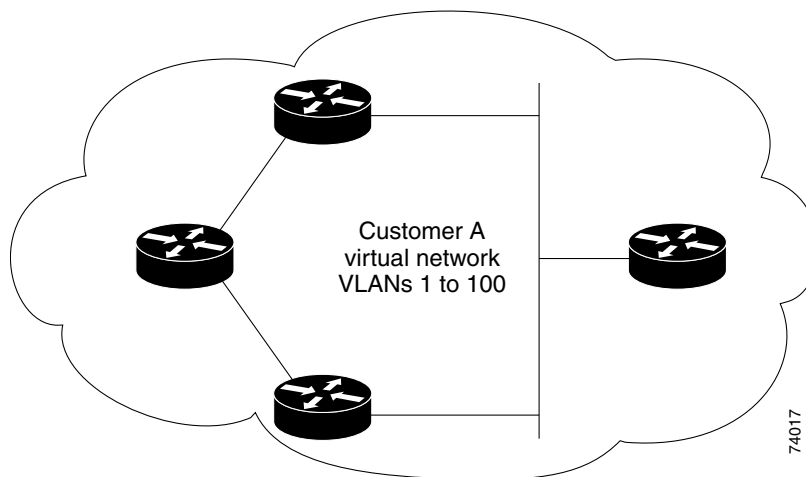
The VLAN ranges required by different customers in the same service provider network might overlap, and customer traffic through the infrastructure might be mixed. Assigning a unique range of VLAN IDs to each customer restricts customer configurations and could easily exceed the VLAN limit (4096) of the 802.1Q specification.

802.1Q tunneling enables service providers to use a single VLAN to support customers who have multiple VLANs, while preserving customer VLAN IDs and keeping traffic in different customer VLANs segregated.

A port configured to support 802.1Q tunneling is called a tunnel port. When you configure tunneling, you assign a tunnel port to a VLAN ID that is dedicated to tunneling. Each customer requires a separate service provider VLAN ID, but that service provider VLAN ID supports VLANs of all the customers.

Customer traffic tagged in the normal way with appropriate VLAN IDs comes from an 802.1Q trunk port on the customer device and into a tunnel port on the service provider edge switch. The link between the customer device and the edge switch is asymmetric because one end is configured as an 802.1Q trunk port, and the other end is configured as a tunnel port. You assign the tunnel port interface to an access VLAN ID that is unique to each customer (Figure 31-1).

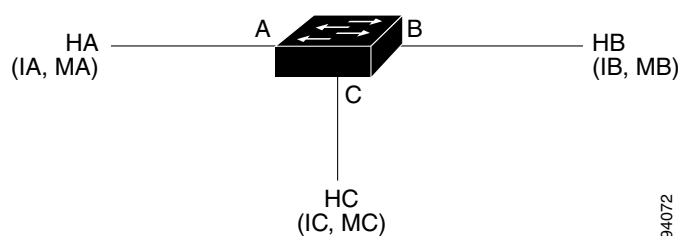
Figure 31-1 802.1Q Tunnel Ports in a Service Provider Network



Packets coming from the customer trunk port into the tunnel port on the service provider edge switch are normally 802.1Q-tagged with the appropriate VLAN ID. When the tagged packets exit the trunk port into the service provider network, they are encapsulated with another layer of an 802.1Q tag (called the *metro tag*) that contains the VLAN ID that is unique to the customer. The original customer 802.1Q tag is preserved in the encapsulated packet. Packets entering the service provider network are double-tagged, with the metro tag containing the customer's access VLAN ID, and the inner VLAN ID being that of the incoming traffic.

When the double-tagged packet enters another trunk port in a service provider core switch, the metro tag is stripped as the switch processes the packet. When the packet exits another trunk port on the same core switch, the same metro tag is again added to the packet. Figure 31-2 shows the tag structures of the Ethernet packets starting with the original, or normal, frame.

Figure 31-2 Original (Normal), 802.1Q, and Double-Tagged Ethernet Packet Formats



When the packet enters the trunk port of the service provider egress switch, the metro tag is again stripped as the switch processes the packet. However, the metro tag is not added when the packet is sent out the tunnel port on the edge switch into the customer network. The packet is sent as a normal 802.1Q-tagged frame to preserve the original VLAN numbers in the customer network.

All packets entering the service provider network through a tunnel port on an edge switch are treated as untagged packets, whether they are untagged or already tagged with 802.1Q headers. The packets are encapsulated with the metro tag VLAN ID (set to the access VLAN of the tunnel port) when they are sent through the service provider network on an 802.1Q trunk port. The priority field on the metro tag is set to the interface class of service (CoS) priority configured on the tunnel port. (The default is zero if none is configured.)

In [Figure 31-1](#), Customer A was assigned VLAN 30, and Customer B was assigned VLAN 40. Packets entering the edge-switch tunnel ports with 802.1Q tags are double-tagged when they enter the service provider network, with the metro tag containing VLAN ID 30 or 40, appropriately, and the inner tag containing the original customer VLAN number, for example, VLAN 100. Even if Customers A and B both have VLAN 100 in their networks, the traffic remains segregated within the service provider network because the metro tag is different. Each customer controls its own VLAN numbering space, which is independent of the VLAN numbering space used by other customers and the VLAN numbering space used by the service provider network.

Configuring 802.1Q Tunneling

These sections describe 802.1Q tunneling configuration:

- [802.1Q Tunneling Configuration Guidelines, page 31-3](#)
- [802.1Q Tunneling and Other Features, page 31-5](#)
- [Configuring an 802.1Q Tunneling Port, page 31-5](#)



Note

By default, 802.1Q tunneling is disabled because the default switch port mode is dynamic auto. Tagging of 802.1Q native VLAN packets on all 802.1Q trunk ports is also disabled.

802.1Q Tunneling Configuration Guidelines

When you configure 802.1Q tunneling, you should always use asymmetrical links for traffic going through a tunnel and should dedicate one VLAN for each tunnel. You should also be aware of configuration requirements for native VLANs and maximum transmission units (MTUs). For more information about MTUs, see the [“System MTU” section on page 31-5](#).

Native VLANs

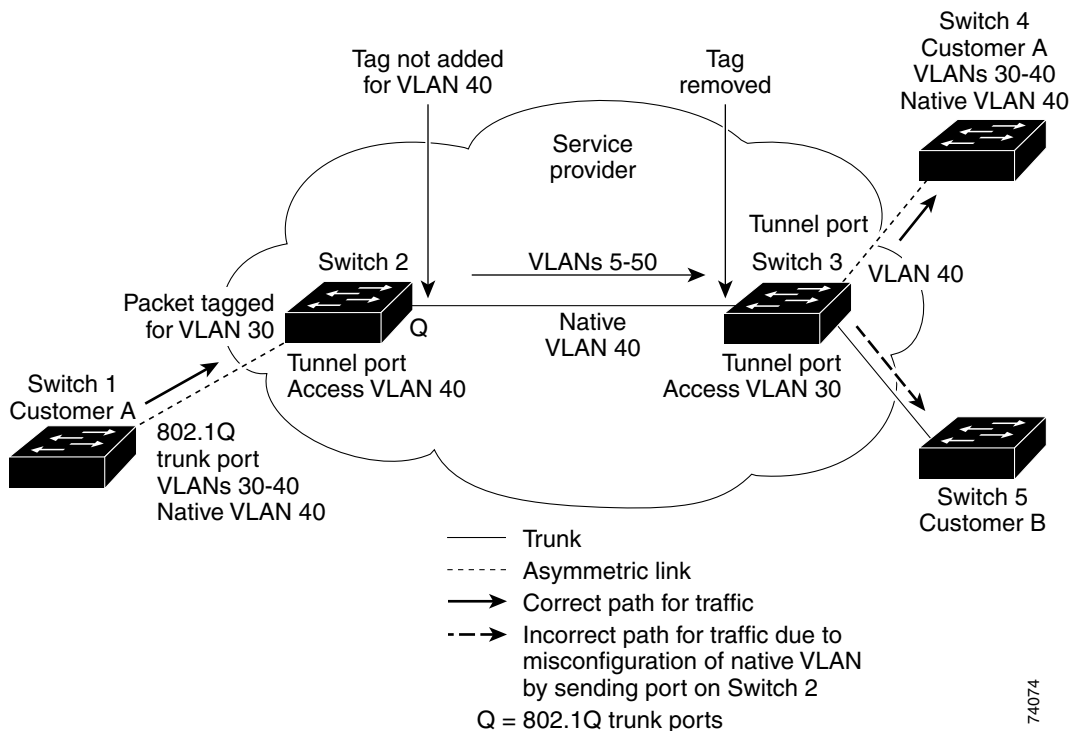
When configuring 802.1Q tunneling on an edge switch, you must use 802.1Q trunk ports for sending packets into the service provider network. However, packets going through the core of the service provider network can be carried through 802.1Q trunks, ISL trunks, or nontrunking links. When 802.1Q trunks are used in these core switches, the native VLANs of the 802.1Q trunks must not match any native VLAN of the nontrunking (tunneling) port on the same switch because traffic on the native VLAN is not tagged on the 802.1Q sending trunk port ([Figure 31-3](#)).

VLAN 40 is configured as the native VLAN for the 802.1Q trunk port from Customer A at the ingress edge switch in the service provider network (Switch 2). Switch 1 of Customer A sends a tagged packet on VLAN 30 to the ingress tunnel port of Switch 2 in the service provider network, which belongs to access VLAN 40. Because the access VLAN of the tunnel port (VLAN 40) is the same as the native VLAN of the edge-switch trunk port (VLAN 40), the metro tag is not added to tagged packets received from the tunnel port. The packet carries only the VLAN 30 tag through the service provider network to the trunk port of the egress-edge switch (Switch 3) and is misdirected through the egress switch tunnel port to Customer B.

These are some ways to solve this problem:

- Use ISL trunks between core switches in the service provider network. Although customer interfaces connected to edge switches must be 802.1Q trunks, we recommend using ISL trunks for connecting switches in the core layer.
- Use the **switchport trunk native vlan tag** per-port command and the **vlan dot1q tag native** global configuration command to configure the edge switch so that all packets going out an 802.1Q trunk, including the native VLAN, are tagged. If the switch is configured to tag native VLAN packets on all 802.1Q trunks, the switch ensures that all packets exiting the trunk are tagged and prevents the reception of untagged packets on the trunk port.
- Ensure that the native VLAN ID on the edge-switch trunk port is not within the customer VLAN range. For example, if the trunk port carries traffic of VLANs 100 to 200, assign the native VLAN a number outside that range.

Figure 31-3 Potential Problem with 802.1Q Tunneling and Native VLANs



System MTU

The default system MTU for traffic on the switch is 1500 bytes. You can configure the switch to support larger frames by using the **system mtu** global configuration command. Because the 802.1Q tunneling feature increases the frame size by 4 bytes when the metro tag is added, you must configure all switches in the service provider network to be able to process larger frames by increasing the switch system MTU size to at least 1504 bytes. The maximum allowable system MTU for Catalyst 4500 Gigabit Ethernet switches is 9198 bytes; the maximum system MTU for Fast Ethernet switches is 1552 bytes.

802.1Q Tunneling and Other Features

Although 802.1Q tunneling works well for Layer 2 packet switching, incompatibilities exist between some Layer 2 features and Layer 3 switching:

- A tunnel port cannot be a routed port.
- IP routing is not supported on a VLAN that includes 802.1Q ports. Packets received from a tunnel port are forwarded based only on Layer 2 information. If routing is enabled on a switch virtual interface (SVI) that includes tunnel ports, untagged IP packets received from the tunnel port are recognized and routed by the switch. Customers can access the Internet through the native VLAN. If this access is not needed, you should not configure SVIs on VLANs that include tunnel ports.
- Tunnel ports do not support IP access control lists (ACLs).
- Layer 3 quality of service (QoS) ACLs and other QoS features related to Layer 3 information are not supported on tunnel ports. MAC-based QoS is supported on tunnel ports.
- EtherChannel port groups are compatible with tunnel ports as long as the 802.1Q configuration is consistent within an EtherChannel port group.
- Port Aggregation Protocol (PAgP), Link Aggregation Control Protocol (LACP), and UniDirectional Link Detection (UDLD) are supported on 802.1Q tunnel ports.
- Dynamic Trunking Protocol (DTP) is not compatible with 802.1Q tunneling because you must manually configure asymmetric links with tunnel ports and trunk ports.
- Loopback detection is supported on 802.1Q tunnel ports.
- When a port is configured as an 802.1Q tunnel port, spanning-tree bridge protocol data unit (BPDU) filtering is automatically enabled on the interface. Cisco Discovery Protocol (CDP) is automatically disabled on the interface.

Configuring an 802.1Q Tunneling Port

To configure a port as an 802.1Q tunnel port, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Enters interface configuration mode and the interface to be configured as a tunnel port. This should be the edge port in the service provider network that connects to the customer switch. Valid interfaces include physical interfaces and port-channel logical interfaces (port channels 1 to 64).
Step 3	Switch(config-if)# switchport access vlan <i>vlan-id</i>	Specifies the default VLAN, which is used if the interface stops trunking. This VLAN ID is specific to the particular customer.

	Command	Purpose
Step 4	Switch(config-if)# switchport mode dot1q-tunnel	Sets the interface as an 802.1Q tunnel port.
Step 5	Switch(config-if)# exit	Returns to global configuration mode.
Step 6	Switch(config)# vlan dot1q tag native	(Optional) Sets the switch to enable tagging of native VLAN packets on all 802.1Q trunk ports. When not set, and a customer VLAN ID is the same as the native VLAN, the trunk port does not apply a metro tag, and packets could be sent to the wrong destination.
Step 7	Switch(config)# end	Returns to privileged EXEC mode.
Step 8	Switch# show dot1q-tunnel	Displays the tunnel ports on the switch.
Step 9	Switch# show vlan dot1q tag native	Displays 802.1Q native-VLAN tagging status.
Step 10	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Use the **no vlan dot1q tag native** global command and the **no switchport mode dot1q-tunnel** interface configuration command to return the port to the default state of dynamic auto. Use the **no vlan dot1q tag native** global configuration command to disable tagging of native VLAN packets.

This example shows how to configure an interface as a tunnel port, enable tagging of native VLAN packets, and verify the configuration. In this configuration, the VLAN ID for the customer connected to Gigabit Ethernet interface 2/7 is VLAN 22.

```
Switch(config)# interface gigabitethernet2/7
Switch(config-if)# switchport access vlan 22
% Access VLAN does not exist. Creating vlan 22
Switch(config-if)# switchport mode dot1q-tunnel
Switch(config-if)# exit
Switch(config)# vlan dot1q tag native
Switch(config)# end
Switch# show dot1q-tunnel interface gigabitethernet2/7
Port
----
LAN Port(s)
----
Gi2/7
Switch# show vlan dot1q tag native
dot1q native vlan tagging is enabled globally
```

About VLAN Mapping



Note

VLAN mapping is supported only on Supervisor Engine 6-E and later engines.

In a typical deployment of VLAN mapping, you want the service provider to provide a transparent switching infrastructure that treats customers' switches at the remote location as a part of the local site. This allows customers to use the same VLAN ID space and run Layer 2 control protocols seamlessly across the provider network. In such scenarios, we recommend that service providers do not impose their VLAN IDs on their customers.

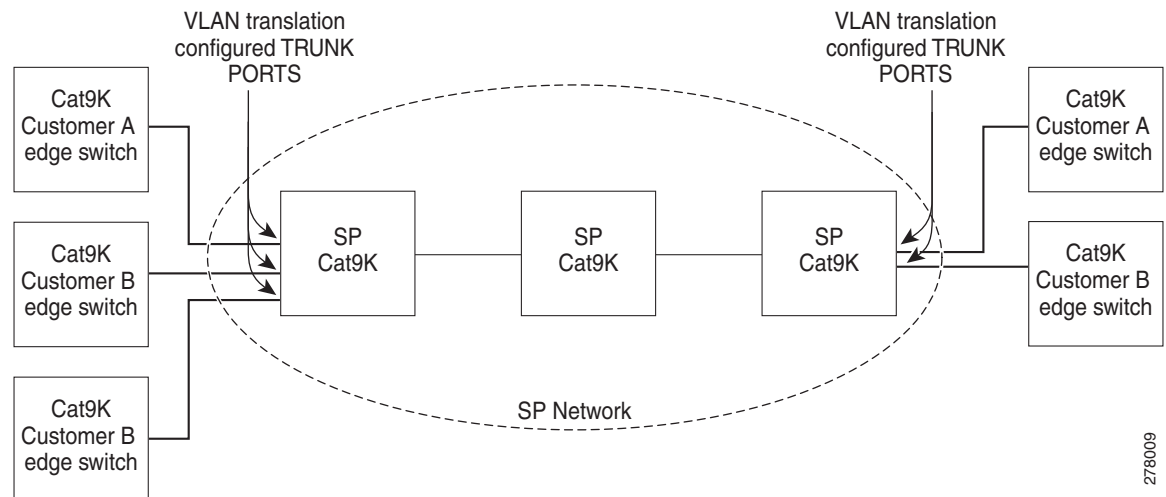
One way to establish translated VLAN IDs (S-VLANs) is to map customer VLANs to service-provider VLANs (called VLAN ID translation) on trunk ports connected to a customer network. Packets entering the port are mapped to a service provider VLAN (S-VLAN) based on the port number and the packet's original customer VLAN-ID (C-VLAN).

Service providers's internal assignments might conflict with a customer's VLAN. To isolate customer traffic, a service provider could decide to map a specific VLAN into another one while the traffic is in its cloud.

Deployment Example

In [Figure 31-4](#), the service provider provides Layer 2 VPN service to two different customers, A and B. The service provider separates the data and control traffic between the two customers and from the providers' own control traffic. The service provider network must also be transparent to the customer edge devices.

Figure 31-4 Layer 2 VPN Deployment



All forwarding operations on the Catalyst 4500 series switch are performed using S-VLAN and not C-VLAN information because the VLAN ID is mapped to the S-VLAN on ingress.



Note

When you configure features on a port configured for VLAN mapping, you always use the S-VLAN rather than the customer VLAN-ID (C-VLAN).

On an interface configured for VLAN mapping, the specified C-VLAN packets are mapped to the specified S-VLAN when they enter the port. Symmetrical mapping to the customer C-VLAN occurs when packets exit the port.

The switch supports these types of VLAN mapping on UNI trunk ports:

- One-to-one VLAN mapping occurs at the ingress and egress of the port and maps the customer C-VLAN ID in the 802.1Q tag to the service-provider S-VLAN ID. You can also specify that packets with all other Vlan Ids are dropped. See the [“One-to-One Mapping”](#) section on page 31-10.

- Traditional 802.1Q tunneling (QinQ) performs all-to-one bundling of C-VLAN IDs to a single S-VLAN ID for the port. The S-VLAN is added to the incoming unmodified C-VLAN. You can configure the UNI as an 802.1Q tunnel port for traditional QinQ, or you can configure selective QinQ on trunk ports for a more flexible implementation. Mapping takes place at ingress and egress of the port. All packets on the port are bundled into the specified S-VLAN. See the [“Traditional Q-in-Q on a Trunk Port”](#) section on page 31-11.
- Selective QinQ maps the specified customer VLANs entering the UNI to the specified S-VLAN ID. The S-VLAN is added to the incoming unmodified C-VLAN. You can also specify whether traffic carrying all other customer VLAN IDs should be dropped or not. See the [“Selective Q-in-Q on a Trunk Port”](#) section on page 31-12.

**Note**

Untagged packets enter the switch on the trunk native VLAN and are not mapped.

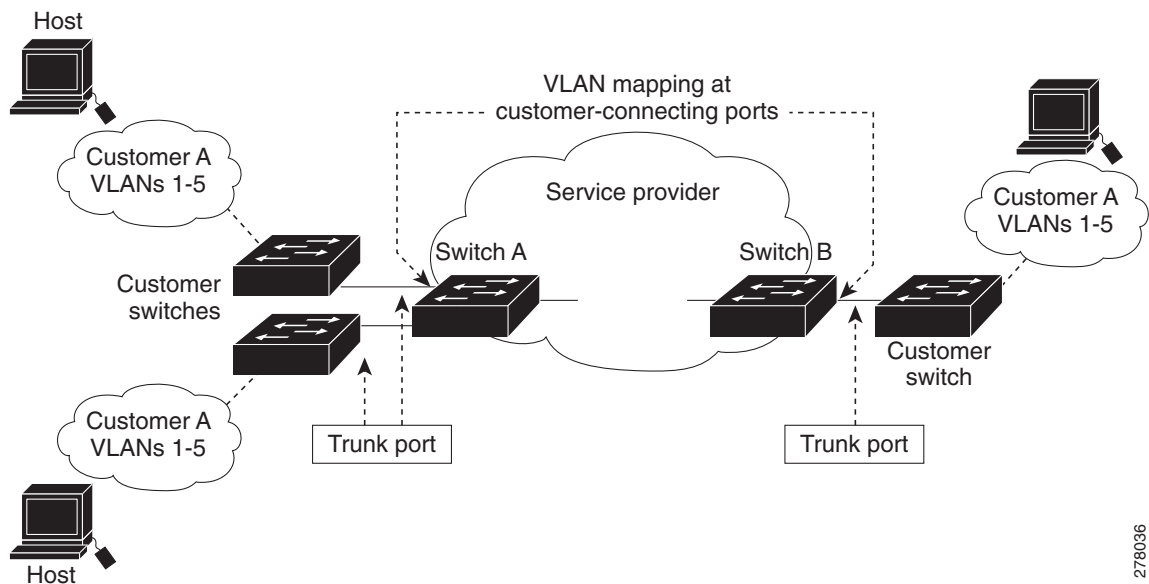
For quality of service (QoS), the switch supports flexible mapping between C-CoS or C-DSCP and S-CoS, and maps the inner CoS to the outer CoS for traffic with traditional QinQ or selective QinQ VLAN mapping.

Mapping Customer VLANs to Service-Provider VLANs

Figure 31-5 shows a topology where a customer uses the same VLANs in multiple sites on different sides of a service-provider network. You map the customer VLAN IDs to service-provider VLAN IDs for packet travel across the service-provider backbone. The customer VLAN IDs are retrieved at the other side of the service-provider backbone for use in the other customer site. Configure the same set of VLAN mappings at a customer-connected port on each side of the service-provider network.

The examples following the configuration steps illustrate how to use one-to-one mapping, traditional QinQ, or selective QinQ to map customer VLANs 1 to 5 to service-provider VLANs.

Figure 31-5 Mapping Customer VLANs



278036

Configuring VLAN Mapping

- [Default VLAN Mapping Configuration, page 31-9](#)
- [VLAN Mapping Configuration Guidelines, page 31-9](#)
- [Configuring VLAN Mapping, page 31-10](#)

Default VLAN Mapping Configuration

By default, no VLAN mapping is configured.

VLAN Mapping Configuration Guidelines

Guidelines include the following:

- Traditional QinQ uses 802.1Q tunnel ports; you configure one-to-one VLAN mapping and selective QinQ on 802.1Q trunk ports.
- To avoid mixing customer traffic, when you configure traditional Q-in-Q on a trunk port, you should configure the service provider S-VLAN ID as an allowed VLAN on the trunk port.
- When you configure VLAN mapping on an EtherChannel, the mapping applies to all ports in the port channel.
- You cannot configure encapsulation replicate on a SPAN destination port if the source port is configured as a tunnel port or has a 1-to-2 mapping configured. Encapsulation replicate is supported with 1-to-1 VLAN mapping.
- When configuring IEEE 802.1Q tunneling on an edge switch, you must use IEEE 802.1Q trunk ports for sending packets into the service-provider network. However, packets going through the core of the service-provider network can be carried through IEEE 802.1Q trunks, ISL trunks, or nontrunking links. When IEEE 802.1Q trunks are used in these core switches, the native VLANs of the IEEE 802.1Q trunks must not match any native VLAN of the nontrunking (tunneling) port on the same switch. It is because traffic on the native VLAN is not tagged on the IEEE 802.1Q sending trunk port.
- Ensure that the native VLAN for an IEEE 802.1Q trunk is the same on both ends of the trunk link. If the native VLAN on one end of the trunk is different from the native VLAN on the other end, spanning-tree loops might result.
- Layer 2 protocol tunneling must be configured for CDP, VTP, LLDP, or your switch detects the SP switches, which is not desirable:

```
interface GigabitEthernet1/23
switchport mode trunk
switchport vlan mapping 1 dot1q-tunnel 311
switchport vlan mapping 31 dot1q-tunnel 311
l2protocol-tunnel cdp
l2protocol-tunnel ll dp
l2protocol-tunnel vtp
```

- To process control traffic consistently, either enable Layer 2 protocol tunneling (recommended) or insert a BPDU filter for spanning tree, as follows:

```
Current configuration : 153 bytes
!
interface FastEthernet9/1
```

```

switchport trunk native vlan 40
switchport mode trunk
switchport vlan mapping 10 20
spanning-tree bpdufilter enable
end

```

- If you need to merge CVLAN and SVLAN spanning tree topology, you do not need to configure **spanning-tree bpdufilter enable**.
- To ensure consistent operation, do not use a native VLAN for translation.

Configuring VLAN Mapping

The following procedures show how to configure each type of VLAN mapping on trunk ports. To verify your configuration, enter either the **show interfaces interface-id vlan mapping** or the **show vlan mapping** privileged EXEC command. See the “[Monitoring and Maintaining Tunneling Status](#)” section on page 31-23 for the syntax of these commands. For more information about all commands in this section, see the *Catalyst 4500 Series Switch Software Command Reference* for this release.

The following VLAN mapping types are discussed:

- [One-to-One Mapping, page 31-10](#)
- [Traditional Q-in-Q on a Trunk Port, page 31-11](#)
- [Selective Q-in-Q on a Trunk Port, page 31-12](#)

One-to-One Mapping

To configure one-to-one VLAN mapping to map a customer VLAN ID to a service-provider VLAN ID, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface interface-id	Enters interface configuration mode for the interface connected to the service-provider network. You can enter a physical interface or an EtherChannel port channel.
Step 3	Switch(config-if)# switchport mode trunk	Configures the interface as a trunk port.
Step 4	Switch(config-if)# switchport vlan mapping vlan-id translated-id	<p>Enters the VLAN IDs to be mapped:</p> <ul style="list-style-type: none"> • <i>vlan-id</i>—the customer VLAN ID (C-VLAN) entering the switch from the customer network. The range is from 1 to 4094. • <i>translated-id</i>—the assigned service-provider VLAN ID (S-VLAN). The range is from 1 to 4094. <p>Note Packets with unconfigured <code>vlan_ids</code> are dropped.</p>
Step 5	Switch# end	Returns to privileged EXEC mode.
Step 6	Switch# show vlan mapping	Verifies the configuration.
Step 7	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Use the **no switchport vlan mapping *vlan-id translated-id*** command to remove the VLAN mapping information. Entering the **no switchport vlan mapping all** command deletes all mapping configurations.

This example shows how to map VLAN IDs 1 to 5 in the customer network to VLANs 101 to 105 in the service-provider network (Figure 31-5). You configure the same VLAN mapping commands for a port in Switch A and Switch B; the traffic on all other VLAN IDs is dropped.

```
Switch(config)# interface gigabitEthernet0/1
Switch(config-if)# switchport vlan mapping 1 101
Switch(config-if)# switchport vlan mapping 2 102
Switch(config-if)# switchport vlan mapping 3 103
Switch(config-if)# switchport vlan mapping 4 104
Switch(config-if)# switchport vlan mapping 5 105
Switch(config-if)# exit
```

In the previous example, at the ingress of the service-provider network, VLAN IDs 1 to 5 in the customer network are mapped to VLANs 101 to 105, in the service provider network. At the egress of the service provider network, VLANs 101 to 105 in the service provider network are mapped to VLAN IDs 1 to 5, in the customer network.

**Note**

Packets with unconfigured `vlan_ids` are dropped.

Traditional Q-in-Q on a Trunk Port

To configure VLAN mapping for traditional Q-in-Q on a trunk port or tunneling by default, perform the following task:

**Note**

Configuring tunneling by default bundles all packets on the port into the configured S-VLAN.

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config) # interface <i>interface-id</i>	Enters interface configuration mode for the interface connected to the service provider network. You can enter a physical interface or an EtherChannel port channel.
Step 3	Switch(config-if)# switchport mode trunk	Configures the interface as a trunk port.
Step 4	Switch(config-if)# switchport trunk allowed vlan <i>vlan-id</i>	Configures the outer VLAN of the service provider network (S-VLAN) to to be allowed on the interface. This should be the same outer VLAN ID entered in the next step.
Step 5	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 6	Switch# show interfaces <i>interface-id</i> vlan mapping	Verifies the configuration.
Step 7	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Entering the **no switchport vlan mapping all** command deletes all mapping configurations.

Selective Q-in-Q on a Trunk Port

To configure VLAN mapping for selective Q-in-Q on a trunk port, perform this task:



Note

You cannot configure one-to-one mapping and selective Q-in-Q on the same interface.

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Enters interface configuration mode for the interface connected to the service provider network. You can enter a physical interface or an EtherChannel port channel.
Step 3	Switch(config-if)# switchport mode trunk	Configure the interface as a trunk port.
Step 4	Switch(config-if)# switchport vlan mapping <i>vlan-id</i> dot1q-tunnel <i>outer vlan-id</i>	Enters the VLAN IDs to be mapped: <ul style="list-style-type: none"> <i>vlan-id</i>—The customer VLAN ID (C-VLAN) entering the switch from the customer network. The range is from 1 to 4094. You can enter a string of VLAN-IDs. <i>outer-vlan-id</i>—The outer VLAN ID (S-VLAN) of the service provider network. The range is from 1 to 4094.
Step 5	Switch(config-if)# switchport vlan mapping default drop	Specifies that all packets on the port are dropped if they do not match the mapped VLANs (Step 4). By default, packets that do not match, are dropped. If the packets (that do not match) should not be dropped, enter the no version of the command.
Step 6	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 7	Switch# show interfaces <i>interface-id</i> vlan mapping	Verifies the configuration.
Step 8	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Use the **no switchport vlan mapping** *vlan-id* **dot1q-tunnel** *outer vlan-id* command to remove the VLAN mapping configuration. Entering the **no switchport vlan mapping all** command deletes all mapping configurations.

This example shows how to configure selective QinQ mapping on the port so that traffic with a C-VLAN ID of 1 to 5 enters the switch with an S-VLAN ID of 100. The traffic of any other VLAN ID is dropped.

```
Switch(config)# interface gigabitEthernet0/1
Switch(config-if)# switchport vlan mapping 1-5 dot1q-tunnel 100
Switch(config-if)# switchport vlan mapping default drop
Switch(config-if)# exit
```

This example shows how to configure selective QinQ mapping on the port so that traffic with a C-VLAN ID of 1 to 5 enters the switch with an S-VLAN ID of 100. The traffic of any other VLAN ID is allowed.

```
Switch(config)# interface gigabitEthernet0/1
Switch(config-if)# switchport vlan mapping 1-5 dot1q-tunnel 100
Switch(config-if)# no switchport vlan mapping default drop
Switch(config-if)# exit
```

About Layer 2 Protocol Tunneling

**Note**

IPsec VPN is supported for control plane traffic protection on the management port, and must be used for management purposes only.

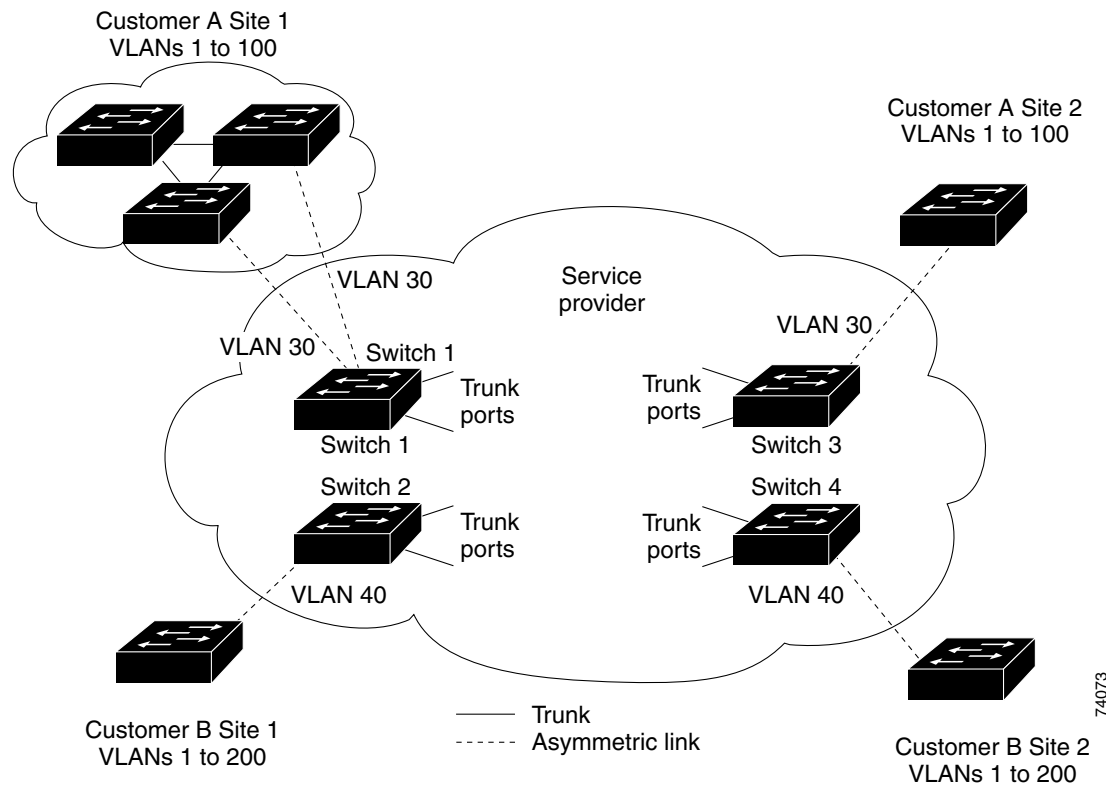
Customers at different sites connected across a service provider network need to use various Layer 2 protocols to scale their topologies to include all remote and local sites. STP must run properly, and every VLAN should build a proper spanning tree that includes the local site and all remote sites across the service provider network. Cisco Discovery Protocol (CDP) must discover neighboring Cisco devices from local and remote sites. VLAN Trunking Protocol (VTP) must provide consistent VLAN configuration throughout all sites in the customer network.

When protocol tunneling is enabled, edge switches on the inbound side of the service provider network encapsulate Layer 2 protocol packets with a special MAC address and send them across the service provider network. Core switches in the network do not process these packets but forward them as normal packets. Layer 2 protocol data units (PDUs) for CDP, STP, or VTP cross the service provider network and are delivered to customer switches on the outbound side of the service provider network. Identical packets are received by all customer ports on the same VLANs with these results:

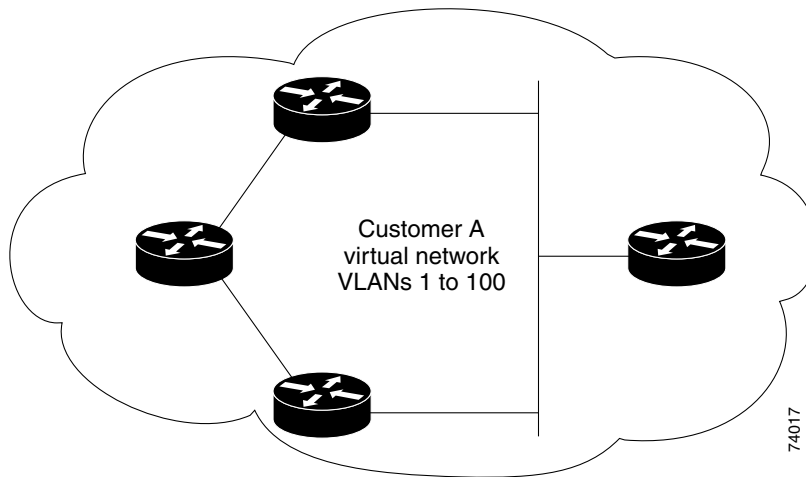
- Users on each of a customer's sites can properly run STP, and every VLAN can build a correct spanning tree, based on parameters from all sites and not just from the local site.
- CDP discovers and shows information about the other Cisco devices connected through the service provider network.
- VTP provides consistent VLAN configuration throughout the customer network, propagating to all switches through the service provider.

Layer 2 protocol tunneling can be enabled on trunk, access and tunnel ports. If protocol tunneling is not enabled, remote switches at the receiving end of the service provider network do not receive the PDUs and cannot properly run STP, CDP, and VTP. When protocol tunneling is enabled, Layer 2 protocols within each customer's network are totally separate from those running within the service provider network.

As an example, Customer A in [Figure 31-6](#) has four switches in the same VLAN that are connected through the service provider network. If the network does not tunnel PDUs, switches on the far ends of the network cannot properly run STP, CDP, and VTP. For example, STP for a VLAN on a switch in Customer A's Site 1 builds a spanning tree on the switches at that site without considering convergence parameters based on Customer A's switch in Site 2. [Figure 31-6](#) shows one possible spanning tree topology.

Figure 31-6 Layer 2 Protocol Tunneling

74073

Figure 31-7 Layer 2 Network Topology without Proper Convergence

74017

Configuring Layer 2 Protocol Tunneling

You can enable Layer 2 protocol tunneling (by protocol) on access ports, tunnel ports, or trunk ports that are connected to the customer in the edge switches of the service provider network. The service provider edge switches connected to the customer switch perform the tunneling process. Edge-switch tunnel ports or normal trunk ports can be connected to customer 802.1Q trunk ports. Edge-switch access ports are connected to customer access ports.

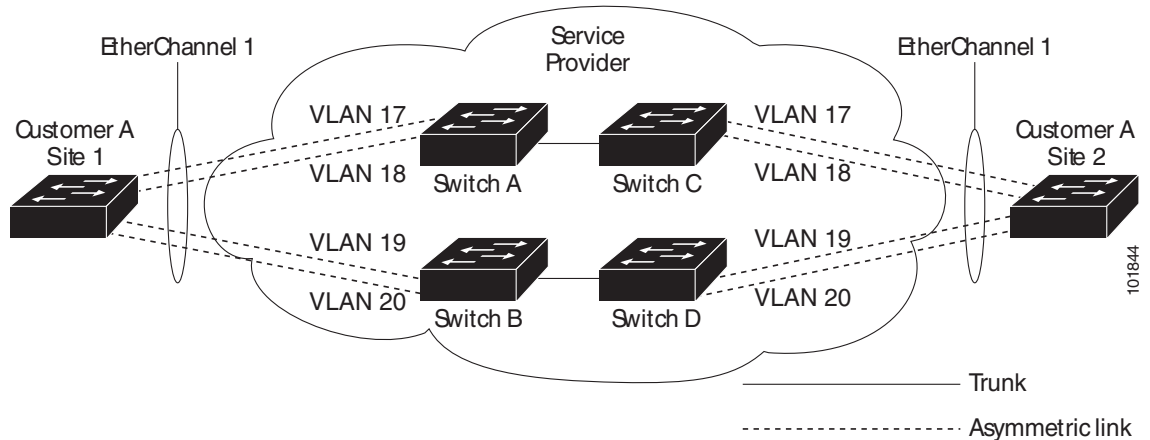
When the Layer 2 PDUs that entered the service provider inbound edge switch through the tunnel port or the access port exit through its the trunk port into the service provider network, the switch overwrites the customer PDU-destination MAC address with a well-known Cisco proprietary multicast address (01-00-0c-cd-cd-d0). If 802.1Q tunneling is enabled, packets are also double-tagged; the outer tag is the customer metro tag, and the inner tag is the customer's VLAN tag. The core switches ignore the inner tags and forward the packet to all trunk ports in the same metro VLAN. The edge switches on the outbound side restore the proper Layer 2 protocol and MAC address information and forward the packets to all tunnel or access ports in the same metro VLAN. This section includes these topics. The Layer 2 PDUs remain intact and are delivered across the service provider network to the other side of the customer network.

Figure 31-6 shows Customer A and Customer B in access VLANs 30 and 40. Asymmetric links connect the Customers in Site 1 to edge switches in the service provider network. The Layer 2 PDUs (for example, BPDUs) coming into Switch 2 from Customer B in Site 1 are forwarded to the infrastructure as double-tagged packets with the well-known MAC address as the destination MAC address. These double-tagged packets have the metro VLAN tag of 40, as well as an inner VLAN tag (for example, VLAN 100). When the double-tagged packets enter Switch 4, the metro VLAN tag 40 is removed. The well-known MAC address is replaced with the respective Layer 2 protocol MAC address, and the packet is sent to Customer B on Site 2 as a single-tagged frame in VLAN 100.

You can also enable Layer 2 protocol tunneling on access ports on the edge switch connected to access ports on the customer switch. The encapsulation and de-encapsulation process is the same as described in the previous paragraph, except that the packets are not double-tagged in the service provider network. The single tag is the customer-specific access VLAN tag.

In an SP (service-provider) network, you can use Layer 2 protocol tunneling to enhance the creation of EtherChannels by emulating a point-to-point network topology. When you enable protocol tunneling (PAgP or LACP) on the SP switch, remote customer switches receive the PDUs and can negotiate the automatic creation of EtherChannels.

For example, in the following figure (Layer 2 Protocol Tunneling for EtherChannels), Customer A has two switches in the same VLAN that are connected through the SP network. When the network tunnels PDUs, switches on the far ends of the network can negotiate the automatic creation of EtherChannels without needing dedicated lines.

Figure 31-8 Layer 2 Protocol Tunneling for EtherChannels

This section contains the following subsections:

- [Default Layer 2 Protocol Tunneling Configuration, page 31-16](#)
- [Layer 2 Protocol Tunneling Configuration Guidelines, page 31-16](#)
- [Configuring Layer 2 Tunneling, page 31-17](#)
- [Configuring Layer 2 Tunneling for EtherChannels, page 31-19](#)

Default Layer 2 Protocol Tunneling Configuration

[Table 31-1](#) shows the default configuration for Layer 2 protocol tunneling.

Table 31-1 Default Layer 2 Ethernet Interface VLAN Configuration

Feature	Default Setting
Layer 2 protocol tunneling	Disabled.
Shutdown threshold	None set.
Drop threshold	None set.
CoS value	If a CoS value is configured on the interface for data packets, that value is the default used for Layer 2 PDUs. If none is configured, the default is 5.

Layer 2 Protocol Tunneling Configuration Guidelines

These are some configuration guidelines and operating characteristics of Layer 2 protocol tunneling:

- The switch supports tunneling of CDP, STP, including multiple STP (MSTP), and VTP. Protocol tunneling is disabled by default but can be enabled for the individual protocols on 802.1Q tunnel ports, access ports or trunk ports.
- Dynamic Trunking Protocol (DTP) is not compatible with Layer 2 protocol tunneling because you must manually configure asymmetric links with tunnel ports and trunk ports.

- EtherChannel port groups are compatible with tunnel ports when the 802.1Q configuration is consistent within an EtherChannel port group.
- If an encapsulated PDU (with the proprietary destination MAC address) is received on a port with Layer 2 tunneling enabled, the port is shut down to prevent loops.
- The port also shuts down when a configured shutdown threshold for the protocol is reached. You can manually reenab the port (by entering a **shutdown** and a **no shutdown** command sequence). If errdisable recovery is enabled, the operation is retried after a specified time interval.
- Only decapsulated PDUs are forwarded to the customer network. The spanning-tree instance running on the service provider network does not forward BPDUs to Layer 2 protocol tunneling ports. CDP packets are not forwarded from Layer 2 protocol tunneling ports.
- When protocol tunneling is enabled on an interface, you can set a per-protocol, per-port, shutdown threshold for the PDUs generated by the customer network. If the limit is exceeded, the port shuts down. You can also limit the BPDU rate by using QoS ACLs and policy maps on a Layer 2 protocol tunneling port.
- When protocol tunneling is enabled on an interface, you can set a per-protocol, per-port, drop threshold for the PDUs generated by the customer network. If the limit is exceeded, the port drops PDUs until the rate at which it receives them is below the drop threshold.
- Because tunneled PDUs (especially STP BPDUs) must be delivered to all remote sites so that the customer virtual network operates properly, you can give PDUs higher priority within the service provider network than data packets received from the same tunnel port. By default, the PDUs use the same CoS value as data packets.

**Note**

If Layer 2 protocol tunneling is not configured on a system, Layer 2 protocol tunneling packets are handled as data packets and this situation does not apply.

Configuring Layer 2 Tunneling

To configure a port for Layer 2 protocol tunneling, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Enters interface configuration mode, and enter the interface to be configured as a tunnel port. This should be the edge port in the service provider network that connects to the customer switch. Valid interfaces can be physical interfaces and port-channel logical interfaces (port channels 1 to 64).
Step 3	Switch(config-if)# switchport mode access or Switch(config-if)# switchport mode dot1q-tunnel or Switch(config-if)# switchport mode trunk	Configures the interface as an access port, an 802.1Q tunnel port or a trunk port.
Step 4	Switch(config-if)# l2protocol-tunnel [cdp point-to-point stp stp vtp]	Enables protocol tunneling for the desired protocol. If no keyword is entered, tunneling is enabled for all three Layer 2 protocols.

	Command	Purpose
Step 5	Switch(config-if)# l2protocol-tunnel shutdown-threshold [cdp point-to-point stp vtp] <i>value</i>	(Optional) Configures the threshold for packets-per-second accepted for encapsulation. The interface is disabled if the configured threshold is exceeded. If no protocol option is specified, the threshold applies to each of the tunneled Layer 2 protocol types. The range is 1 to 4096. The default is to have no threshold configured. Note If you also set a drop threshold on this interface, the shutdown-threshold value must be greater than or equal to the drop-threshold value.
Step 6	Switch(config-if)# l2protocol-tunnel drop-threshold [cdp point-to-point stp vtp] <i>value</i>	(Optional) Configures the threshold for packets-per-second accepted for encapsulation. The interface drops packets if the configured threshold is exceeded. If no protocol option is specified, the threshold applies to each of the tunneled Layer 2 protocol types. The range is 1 to 4096. The default is to have no threshold configured. Note If you also set a shutdown threshold on this interface, the drop-threshold value must be less than or equal to the shutdown-threshold value.
Step 7	Switch(config-if)# exit	Returns to global configuration mode.
Step 8	Switch(config)# errdisable recovery cause l2ptguard	(Optional) Configures the recovery method from a Layer 2 maximum-rate error so that the interface is reenabled and can try again. Errdisable recovery is disabled by default; when enabled, the default time interval is 300 seconds.
Step 9	Switch(config)# l2protocol-tunnel cos <i>value</i>	(Optional) Configures the CoS value for all tunneled Layer 2 PDUs. The range is 0 to 7; the default is the default CoS value for the interface. If none is configured, the default is 5.
Step 10	Switch(config)# end	Returns to privileged EXEC mode.
Step 11	Switch# show l2protocol	Displays the Layer 2 tunnel ports on the switch, including the protocols configured, the thresholds, and the counters.
Step 12	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Use the **no l2protocol-tunnel [cdp | stp | vtp]** interface configuration command to disable protocol tunneling for one of the Layer 2 protocols or for all three. Use the **no l2protocol-tunnel shutdown-threshold [cdp | stp | vtp]** and the **no l2protocol-tunnel drop-threshold [cdp | stp | vtp]** commands to return the shutdown and drop thresholds to the default settings.

This example shows how to configure Layer 2 protocol tunneling on an 802.1Q tunnel port for CDP, STP, VTP, and LLDP and how to verify the configuration:

```
Switch(config)# interface FastEthernet 1/1/11
Switch(config-if)# switchport mode dot1q-tunnel
Switch(config-if)# l2protocol-tunnel cdp
Switch(config-if)# l2protocol-tunnel stp
Switch(config-if)# l2protocol-tunnel vtp
Switch(config-if)# l2protocol-tunnel llbp
Switch(config-if)# l2protocol-tunnel shutdown-threshold 1500
Switch(config-if)# l2protocol-tunnel drop-threshold 1000
Switch(config-if)# exit
Switch(config)# l2protocol-tunnel cos 7
Switch(config)# end
Switch# show l2protocol
COS for Encapsulated Packets: 7
```

Drop Threshold for Encapsulated Packets: 0


Port	Protocol	Shutdown Threshold	Drop Threshold	Encaps Counter	Decaps Counter	Drop Counter
Gi1/1/11	cdp	1500	1000	2288	2282	0
	lldp	1500	1000	0	0	0
	stp	1500	1000	116	13	0
	vtp	1500	1000	3	67	0
---	---	----	----	----	----	----
---	---	----	----	----	----	----
---	---	----	----	----	----	----

Configuring Layer 2 Tunneling for EtherChannels

To configure Layer 2 point-to-point tunneling to facilitate the automatic creation of EtherChannels, you need to configure both the SP edge switch and the customer switch. This section includes the following tasks:

- [Configuring the SP Edge Switch, page 31-19](#)
- [Configuring the Customer Switch, page 31-21](#)

Configuring the SP Edge Switch

	Command	Purpose
Step 1	Switch# configure terminal	Enters the global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Enters interface configuration mode. Enter the interface to be configured as a tunnel port. This should be the edge port in the service provider network that connects to the customer switch.
Step 3	Switch(config-if)# switchport mode dot1q-tunnel	Configures the interface as an 802.1Q tunnel port
Step 4	Switch(config-if)# l2protocol-tunnel point-to-point [pagp lacp udld]	(Optional) Enables point-to-point protocol tunneling for the desired protocol. If no keyword is entered, tunneling is enabled for all three protocols.
		 Caution To avoid a network failure, make sure that the network is a point-to-point topology before you enable tunneling for PAgP, LACP, or UDLD packets.
Step 5	Switch(config-if)# l2protocol-tunnel shutdown-threshold [point-to-point [pagp lacp udld]] <i>value</i>	(Optional) Configures the threshold for packets-per-second accepted for encapsulation. The interface is disabled if the configured threshold is exceeded. If no protocol option is specified, the threshold applies to each of the tunneled Layer 2 protocol types. The range is 1 to 4096. The default is to have no threshold configured.
		Note If you also set a drop threshold on this interface, the shutdown-threshold value must be greater than or equal to the drop-threshold value.

	Command	Purpose
Step 6	Switch(config-if)# l2protocol-tunnel drop-threshold [point-to-point [pagp lacp udld]] value	(Optional) Configures the threshold for packets-per-second accepted for encapsulation. The interface drops packets if the configured threshold is exceeded. If no protocol option is specified, the threshold applies to each of the tunneled Layer 2 protocol types. The range is 1 to 4096. The default is to have no threshold configured. Note If you also set a shutdown threshold on this interface, the drop-threshold value must be less than or equal to the shutdown-threshold value.
Step 7	Switch(config-if)# no cdp enable	Disables CDP on the interface.
Step 8	Switch(config-if)# spanning-tree bpdufilter	Enables BPDU filtering on the interface.
Step 9	Switch(config)# exit	Returns to global configuration mode.
Step 10	Switch(config)# errdisable recovery cause l2ptguard	(Optional) Configures the recovery mechanism from a Layer 2 maximum-rate error so that the interface is re-enabled and can try again. Errdisable recovery is disabled by default; when enabled, the default time interval is 300 seconds.
Step 11	Switch(config)# l2protocol-tunnel cos value	(Optional) Configures the CoS value for all tunneled Layer 2 PDUs. The range is 0 to 7; the default is the default CoS value for the interface. If none is configured, the default is 5.
Step 12	Switch(config)# end	Returns to privileged EXEC mode.
Step 13	Switch# show l2protocol	Displays the Layer 2 tunnel ports on the switch, including the protocols configured, the thresholds, and the counters.
Step 14	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Use the **no l2protocol-tunnel [point-to-point [pagp | lacp | udld]]** interface configuration command to disable point-to-point protocol tunneling for one of the Layer 2 protocols or for all three. Use the **no l2protocol-tunnel shutdown-threshold [point-to-point [pagp | lacp | udld]]** and the **no l2protocol-tunnel drop-threshold [point-to-point [pagp | lacp | udld]]** commands to return the shutdown and drop thresholds to their default settings.

This example shows how to configure the SP edge switch:

```
Switch(config)# interface gigabitEthernet 1/1/11
Switch(config-if)# switchport mode dot1q-tunnel
Switch(config-if)# switchport access vlan 10
Switch(config-if)# l2protocol-tunnel point-to-point
Switch(config-if)# l2protocol-tunnel shutdown-threshold point-to-point 3000
Switch(config-if)# l2protocol-tunnel drop-threshold point-to-point 2500
Switch(config-if)# exit
Switch# show l2protocol-tunnel
COS for Encapsulated Packets: 5
Drop Threshold for Encapsulated Packets: 0
```

Port	Protocol	Shutdown Threshold	Drop Threshold	Encaps Counter	Decaps Counter	Drop Counter
Gi1/1/11	---	----	----	----	----	----
	---	----	----	----	----	----
	---	----	----	----	----	----
	---	----	----	----	----	----

pagp	3000	2500	0	0	0
lacp	3000	2500	0	0	0
udld	3000	2500	0	0	0

Configuring the Customer Switch

	Command	Purpose
Step 1	Switch# configure terminal	Enters the global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Enters interface configuration mode. Enter the interface to be configured as a tunnel port.
Step 3	Switch(config-if)# switchport mode trunk	Enables trunking on the interface.
Step 4	Switch(config-if)# udld enable	Enables UDLD in normal mode on the interface
Step 5	Switch(config-if)# channel-group <i>channel-group-number</i> mode desirable	Assigns the interface to a channel group, and specifies desirable for the PAgP mode.
Step 6	Switch(config-if)# exit	Returns to global configuration mode.
Step 7	Switch(config)# interface port-channel <i>port-channel number</i>	Enters port-channel interface mode.
Step 8	Switch(config-if)# shutdown	Shuts down the interface.
Step 9	Switch(config-if)# no shutdown	Enables the interface.
Step 10	Switch(config-if)# end	Returns to global configuration mode.
Step 11	Switch# show l2protocol	Displays the Layer 2 tunnel ports on the switch, including the protocols configured, the thresholds, and the counters.
Step 12	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Use the **no switchport mode trunk**, the **no udld enable**, and the **no channel group channel-group-number mode desirable** interface configuration commands to restore default interface settings.

This example shows you how to configure a customer switch:

```
Switch(config)# interface gigabitEthernet 2/14
Switch(config-if)# switchport mode trunk
Switch(config-if)# udld port
Switch(config-if)# channel-group 10 mode desirable
Switch(config-if)# exit
```

This example shows how to configure the SP edge switch 1 and edge switch 2. VLANs 17, 18, 19, and 20 are the access VLANs, Gigabit Ethernet interfaces 1/1/11 and 1/1/12 are point-to-point tunnel ports with PAgP and UDLD enabled, the drop threshold is 1000, and Fast Ethernet interface 1/1/13 is a trunk port.

SP edge switch 1 configuration:

```
Switch(config)# interface gigabitEthernet 1/1/11
Switch(config-if)# switchport access vlan 17
Switch(config-if)# switchport mode dot1q-tunnel
Switch(config-if)# l2protocol-tunnel point-to-point pagp
```

```

Switch(config-if)# l2protocol-tunnel point-to-point udld
Switch(config-if)# l2protocol-tunnel drop-threshold point-to-point pagp 1000
Switch(config-if)# exit
Switch(config)# interface gigabitEthernet 1/1/12
Switch(config-if)# switchport access vlan 18
Switch(config-if)# switchport mode dot1q-tunnel
Switch(config-if)# l2protocol-tunnel point-to-point pagp
Switch(config-if)# l2protocol-tunnel point-to-point udld
Switch(config-if)# l2protocol-tunnel drop-threshold point-to-point pagp 1000
Switch(config-if)# exit
Switch(config)# interface gigabitEthernet 1/1/13
Switch(config-if)# switchport mode trunk

```

SP edge switch 2 configuration:

```

Switch(config)# interface gigabitEthernet 1/1/11
Switch(config-if)# switchport access vlan 19
Switch(config-if)# switchport mode dot1q-tunnel
Switch(config-if)# l2protocol-tunnel point-to-point pagp
Switch(config-if)# l2protocol-tunnel point-to-point udld
Switch(config-if)# l2protocol-tunnel drop-threshold point-to-point pagp 1000
Switch(config-if)# exit
Switch(config)# interface gigabitEthernet 1/1/12
Switch(config-if)# switchport access vlan 20
Switch(config-if)# switchport mode dot1q-tunnel
Switch(config-if)# l2protocol-tunnel point-to-point pagp
Switch(config-if)# l2protocol-tunnel point-to-point udld
Switch(config-if)# l2protocol-tunnel drop-threshold point-to-point pagp 1000
Switch(config-if)# exit
Switch(config)# interface gigabitEthernet 1/1/13
Switch(config-if)# switchport mode trunk

```

This example shows how to configure the customer switch at Site 1. Gigabit Ethernet interfaces 1/1, 1/2, 1/3, and 1/4 are set for 802.1Q trunking, UDLD is enabled, EtherChannel group 1 is enabled, and the port channel is shut down and then enabled to activate EtherChannel configuration. See also [Figure 31-8](#).

```

Switch(config)# interface GigabitEthernet1/1
Switch(config-if)# switchport mode trunk
Switch(config-if)# udld enable
Switch(config-if)# channel-group 1 mode desirable
Switch(config-if)# exit
Switch(config)# interface GigabitEthernet1/2
Switch(config-if)# switchport mode trunk
Switch(config-if)# udld enable
Switch(config-if)# channel-group 1 mode desirable
Switch(config-if)# exit
Switch(config)# interface GigabitEthernet1/3
Switch(config-if)# switchport mode trunk
Switch(config-if)# udld enable
Switch(config-if)# channel-group 1 mode desirable
Switch(config-if)# exit
Switch(config)# interface GigabitEthernet1/4
Switch(config-if)# switchport mode trunk
Switch(config-if)# udld enable
Switch(config-if)# channel-group 1 mode desirable
Switch(config-if)# exit
Switch(config)# interface port-channel 1
Switch(config-if)# shutdown
Switch(config-if)# no shutdown
Switch(config-if)# exit

```

Monitoring and Maintaining Tunneling Status

Table 31-2 shows the commands for monitoring and maintaining 802.1Q and Layer 2 protocol tunneling.

Table 31-2 Commands for Monitoring and Maintaining Tunneling

Command	Purpose
Switch# clear l2protocol-tunnel counters	Clears the protocol counters on Layer 2 protocol tunneling ports.
Switch# show dot1q-tunnel	Displays 802.1Q tunnel ports on the switch.
Switch# show dot1q-tunnel interface interface-id	Verifies if a specific interface is a tunnel port.
Switch# show l2protocol-tunnel	Displays information about Layer 2 protocol tunneling ports.
Switch# show errdisable recovery	Verifies if the recovery timer from a Layer 2 protocol-tunnel error disable state is enabled.
Switch# show l2protocol-tunnel interface interface-id	Displays information about a specific Layer 2 protocol tunneling port.
Switch# show l2protocol-tunnel summary	Displays only Layer 2 protocol summary information.
Switch# show vlan dot1q native	Displays the status of native VLAN tagging on the switch.



Note

With Cisco IOS Release 12.2(20)EW, the BPDU filtering configuration for both dot1q and Layer 2 protocol tunneling is no longer visible in the running configuration as spanning-tree bpdupfilter enable. The configuration is visible in the output of the **show spanning tree int detail** command.

```
Switch# show spann int f6/1 detail
Port 321 (FastEthernet6/1) of VLAN0001 is listening
  Port path cost 19, Port priority 128, Port Identifier 128.321.
  Designated root has priority 32768, address 0008.e341.4600
  Designated bridge has priority 32768, address 0008.e341.4600
  Designated port id is 128.321, designated path cost 0
  Timers: message age 0, forward delay 2, hold 0
  Number of transitions to forwarding state: 0
  Link type is point-to-point by default
  ** Bpdu filter is enabled internally **
  BPDU: sent 0, received 0
```




Configuring Cisco Discovery Protocol

This chapter describes how to configure Cisco Discovery Protocol and Cisco Discovery Protocol Bypass on the Catalyst 4006 switch with Supervisor Engine III. It also provides guidelines, procedures, and configuration examples.

This chapter includes the following major sections:

- [About Cisco Discovery Protocol, page 32-2](#)
- [Configuring Cisco Discovery Protocol, page 32-2](#)
- [About Cisco Discovery Protocol Bypass, page 32-5](#)
- [Configuring Cisco Discovery Protocol Bypass, page 32-5](#)



Note

For complete syntax and usage information for the Cisco IOS commands used in this chapter, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide*, Release 12.4:

http://www.cisco.com/en/US/docs/ios/fundamentals/configuration/guide/12_4/cf_12_4_book.html

and the *Cisco IOS Configuration Fundamentals Command Reference*, Release 12.2:

http://www.cisco.com/en/US/docs/ios/12_2/configfun/command/reference/ffun_r.html



Note

For complete syntax and usage information for the switch commands used in this chapter, see the *Cisco Catalyst 4500 Series Switch Command Reference* and related publications at this location:

<http://www.cisco.com/en/US/products/hw/switches/ps4324/index.html>

If a command is not in the *Catalyst 4500 Series Switch Command Reference*, you can locate it in the Cisco IOS library. See the *Command Reference for the Catalyst 4006 Switch with Supervisor Engine III* and related publications at this location:

<http://www.cisco.com/en/US/products/ps6350/index.html>

About Cisco Discovery Protocol

Cisco Discovery Protocol is a protocol that runs over Layer 2 (the data link layer) on all Cisco routers, bridges, access servers, and switches. Cisco Discovery Protocol allows network management applications to discover Cisco devices that are neighbors of already known devices, in particular, neighbors running lower-layer, transparent protocols. With Cisco Discovery Protocol, network management applications can learn the device type and the SNMP agent address of neighboring devices. Cisco Discovery Protocol enables applications to send SNMP queries to neighboring devices.

Cisco Discovery Protocol runs on all LAN and WAN media that support Subnetwork Access Protocol (SNAP).

Each Cisco Discovery Protocol-configured device sends periodic messages to a multi-cast address. Each device advertises at least one address at which it can receive SNMP messages. The advertisements also contain the time-to-live, or holdtime information, which indicates the length of time a receiving device should hold Cisco Discovery Protocol information before discarding it.

Configuring Cisco Discovery Protocol

The following sections describe how to configure Cisco Discovery Protocol:

- [Enabling Cisco Discovery Protocol Globally, page 32-2](#)
- [Displaying the Cisco Discovery Protocol Global Configuration, page 32-2](#)
- [Enabling Cisco Discovery Protocol on an Interface, page 32-3](#)
- [Displaying the Cisco Discovery Protocol Interface Configuration, page 32-3](#)
- [Monitoring and Maintaining Cisco Discovery Protocol, page 32-4](#)

Enabling Cisco Discovery Protocol Globally

To enable Cisco Discovery Protocol globally, use this command:

Command	Purpose
Switch(config)# [no] cdp run	Enables Cisco Discovery Protocol globally. Use the no keyword to disable Cisco Discovery Protocol globally.

This example shows how to enable Cisco Discovery Protocol globally:

```
Switch(config)# cdp run
```

Displaying the Cisco Discovery Protocol Global Configuration

To display the Cisco Discovery Protocol configuration, use this command:

Command	Purpose
Switch# show cdp	Displays global Cisco Discovery Protocol information.

This example shows how to display the Cisco Discovery Protocol configuration:

```
Switch# show cdp
Global Cisco Discovery Protocol information:
    Sending CDP packets every 120 seconds
    Sending a holdtime value of 180 seconds
    Sending CDPv2 advertisements is enabled
Switch#
```

For additional Cisco Discovery Protocol **show** commands, see the [“Monitoring and Maintaining Cisco Discovery Protocol”](#) section on page 32-4.

Enabling Cisco Discovery Protocol on an Interface

To enable Cisco Discovery Protocol on an interface, use this command:

Command	Purpose
Switch(config-if)# [no] cdp enable	Enables Cisco Discovery Protocol on an interface. Use the no keyword to disable Cisco Discovery Protocol on an interface.

This example shows how to enable Cisco Discovery Protocol on Fast Ethernet interface 5/1:

```
Switch(config)# interface fastethernet 5/1
Switch(config-if)# cdp enable
```

This example shows how to disable Cisco Discovery Protocol on Fast Ethernet interface 5/1:

```
Switch(config)# interface fastethernet 5/1
Switch(config-if)# no cdp enable
```

Displaying the Cisco Discovery Protocol Interface Configuration

To display the Cisco Discovery Protocol configuration for an interface, use this command:

Command	Purpose
Switch# show cdp interface <i>[type/number]</i>	Displays information about interfaces where Cisco Discovery Protocol is enabled.

This example shows how to display the Cisco Discovery Protocol configuration of Fast Ethernet interface 5/1:

```
Switch# show cdp interface fastethernet 5/1
FastEthernet5/1 is up, line protocol is up
    Encapsulation ARPA
```

```

Sending CDP packets every 120 seconds
Holdtime is 180 seconds
Switch#

```

Monitoring and Maintaining Cisco Discovery Protocol

To monitor and maintain Cisco Discovery Protocol on your device, enter one or more of the following commands:

Command	Purpose
Switch# clear cdp counters	Resets the traffic counters to zero.
Switch# clear cdp table	Deletes the Cisco Discovery Protocol table of information about neighbors.
Switch# show cdp	Displays global information such as frequency of transmissions and the holdtime for packets being transmitted.
Switch# show cdp entry <i>entry_name</i> [protocol version]	Displays information about a specific neighbor. The display can be limited to protocol or version information.
Switch# show cdp interface [<i>type/number</i>]	Displays information about interfaces on which Cisco Discovery Protocol is enabled.
Switch# show cdp neighbors [<i>type/number</i>] [detail]	Displays information about neighboring equipment. The display can be limited to neighbors on a specific interface and expanded to provide more detailed information.
Switch# show cdp traffic	Displays Cisco Discovery Protocol counters, including the number of packets sent and received and checksum errors.
Switch# show debugging	Displays information about the types of debugging that are enabled for your switch.

This example shows how to clear the Cisco Discovery Protocol counter configuration on your switch:

```
Switch# clear cdp counters
```

This example shows how to display information about the neighboring equipment:

```
Switch# show cdp neighbors
```

```

Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater

```

Device ID	Local Intrfce	Holdtme	Capability	Platform	Port ID
JAB023807H1	Fas 5/3	127	T S	WS-C2948	2/46
JAB023807H1	Fas 5/2	127	T S	WS-C2948	2/45
JAB023807H1	Fas 5/1	127	T S	WS-C2948	2/44
JAB023807H1	Gig 1/2	122	T S	WS-C2948	2/50
JAB023807H1	Gig 1/1	122	T S	WS-C2948	2/49
JAB03130104	Fas 5/8	167	T S	WS-C4003	2/47
JAB03130104	Fas 5/9	152	T S	WS-C4003	2/48

About Cisco Discovery Protocol Bypass

When a Cisco IP Phone is plugged into a port that is configured with a Voice VLAN and single-host mode, the phone will be silently allowed onto the network by way of a feature known as Cisco Discovery Protocol Bypass. The phone (or any device) that sends the appropriate Type Length Value (TLV) in a Cisco Discovery Protocol message will be allowed access to the voice VLAN.

In Cisco Discovery Protocol Bypass mode, Cisco Discovery Protocol packets are received and transmitted unchanged. Received packets are not processed. No packets are generated. In this mode, 'bump-in-the-wire' behavior is applied to Cisco Discovery Protocol packets. This is a backward compatible mode, equivalent to not having Cisco Discovery Protocol support.

In Cisco Discovery Protocol Bypass mode authentication sessions are established in single and multi-host modes for IP Phones. However, if voice VLAN and 802.1x on an interface port is enabled, then Cisco Discovery Protocol Bypass is enabled when the host mode is set to single or multi-host mode.

It is possible to use the Multi-Domain Authentication (MDA) feature instead of Cisco Discovery Protocol Bypass feature as it provides better Access Control, Visibility and Authorization.

**Note**

By default the host mode is set to single mode in legacy mode and multi-authentication in the edge mode.

Cisco Discovery Protocol Enhancement for Second Port Disconnect—Allows a Cisco IP phone to send a Cisco Discovery Protocol message to the switch when a host unplugs from behind the phone. The switch is then able to clear any authenticated session for the indirectly connected host, the same as if the host had been directly connected and the switch had detected a link down event. This is supported in latest IP telephones.

Cisco Discovery Protocol Bypass provides no support for third-party phones—Cisco Discovery Protocol Bypass works only with Cisco phones.

Configuring Cisco Discovery Protocol Bypass

The following sections describe how to configure Cisco Discovery Protocol Bypass

- [Enabling Cisco Discovery Protocol Bypass, page 32-6](#)
- [Displaying Cisco Discovery Protocol Neighbors, page 32-7](#)
- [Disabling Cisco Discovery Protocol Bypass, page 32-7](#)

Enabling Cisco Discovery Protocol Bypass

To enable Cisco Discovery Protocol Bypass, use these commands:

Command	Purpose
enable Example: Switch> enable	Enables privileged EXEC mode. Enter your password if prompted.
configure terminal Example: Switch# configure terminal	Enters the global configuration mode.
interface <i>interface-id</i> Example: Switch(config)# interface GigabitEthernet1/0/12	Specifies a physical port, and enters interface configuration mode. Valid interfaces are physical ports.
switchport mode access Example: Switch(config-if)# switchport mode access	Specifies that the interface is in access mode.
switchport access vlan <i>vlan id</i> Example: Switch(config-if)# switchport access vlan 10	Assigns all ports as static-access ports in the same VLAN If you configure the port as a static-access port, assign it to only one VLAN. The range is 1 to 4094.
switchport voice vlan <i>vlan-id</i> Example: Switch(config-if)# switchport voice vlan 3	Instruct the Cisco IP phone to forward all voice traffic through the specified VLAN. By default, the Cisco IP phone forwards the voice traffic with an 802.1Q priority of 5. Valid VLAN IDs are from 1 to 4094 when the enhanced software image (EI) is installed and 1 to 1001 when the standard software image is installed. Do not enter leading zeros.
authentication port-control auto Example: Switch(config-if)# authentication port-control auto	Enables 802.1x authentication on the port.

Command	Purpose
authentication host-mode {single-host multi-host} Example: Switch(config-if)# authentication host-mode single multi-host	The keywords allow the following: single-host -Single host (client) on an IEEE 802.1X-authorized port. multi-host -Multiple hosts on an 802.1X-authorized port after authenticating a single host.
dot1x pae authenticator or mab Example: Switch(config-if)# dot1x pae authenticator or Switch(config-if)# mab	Enables 802.1X authentication on the port with default parameters

Cisco Discovery Protocol Bypass is enabled by default once **authentication port-control auto** is configured with dot1x or MAB or if voice vlan is configured on the interface along with single/multiple host mode.

Displaying Cisco Discovery Protocol Neighbors

The following configuration example displays Cisco Discovery Protocol neighbors.

```
Switch# show cdp neighbors g1/0/37 detail
Switch ID: SEP24B657B038DF
Entry address(es):
Platform: Cisco IP Phone 9971, Capabilities: Host Phone Two-port Mac Relay Interface:
GigabitEthernet1/0/37, Port ID (outgoing port): Port 1 Holdtime : 157 sec
Second Port Status: Down<<
Version:sip9971.9-1-1SR1
advertisement version: 2
Duplex: full
Power drawn: 12.804 Watts
Power request id: 57146, Power management id: 4
Power request levels are:12804 0 0 0 0
Total cdp entries displayed : 1
```

Disabling Cisco Discovery Protocol Bypass

To disable Cisco Discovery Protocol Bypass, enter the **no authentication port-control auto** command in interface configuration mode.



Configuring LLDP, LLDP-MED, and Location Service

This chapter describes how to configure the Link Layer Discovery Protocol (LLDP), LLDP Media Endpoint Discovery (LLDP-MED), and Location Service on the Catalyst 4500 series switch.

This chapter consists of these sections:

- [About LLDP, LLDP-MED, and Location Service, page 33-1](#)
- [Configuring LLDP and LLDP-MED, and Location Service, page 33-4](#)
- [Monitoring and Maintaining LLDP, LLDP-MED, and Location Service, page 33-14](#)
- [Cisco IOS Carries Ethernet Features in Cisco IOS XE 3.1.0SG, page 33-15](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Configuration Fundamentals Command Reference](#) and the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About LLDP, LLDP-MED, and Location Service

This section describes this information:

- [Restrictions for LLDP, page 33-1](#)
- [LLDP, page 33-2](#)
- [LLDP-MED, page 33-2](#)
- [Location Service, page 33-3](#)

Restrictions for LLDP

When Cisco Discovery Protocol and LLDP are both in use within the same switch, it is necessary to disable LLDP on interfaces where Cisco Discover Protocol is in use for power negotiation. LLDP can be disabled at interface level with the commands **no lldp tlv-select power-management** or **no lldp transmit / no lldp receive**.

LLDP

The Cisco Discovery Protocol (CDP) is a device discovery protocol that runs over Layer 2 (the data link layer) on all Cisco-manufactured devices (routers, bridges, access servers, and switches). CDP allows network management applications to automatically discover and learn about other Cisco devices connected to the network.

To support non-Cisco devices and to allow for interoperability between other devices, the switch supports the IEEE 802.1AB LLDP. LLDP is a neighbor discovery protocol that is used for network devices to advertise information about themselves to other devices on the network. This protocol runs over the data-link layer, which allows two systems running different network layer protocols to learn about each other.

LLDP supports a set of attributes that it uses to discover neighbor devices. These attributes contain type, length, and value descriptions and are referred to as *TLVs*. LLDP supported devices can use TLVs to receive and send information to their neighbors. Details such as configuration information, device capabilities, and device identity can be advertised using this protocol.

The switch supports the following basic management TLVs (which are optional):

- Port description TLV
- System name TLV
- System description TLV
- System capabilities TLV
- Management address TLV
- Power Management TLV

These organizationally specific LLDP TLVs are also advertised to support LLDP-MED:

- Port VLAN ID TLV ((IEEE 802.1 specific TLVs)
- MAC/PHY configuration/status TLV(IEEE 802.3 specific TLVs)

LLDP-MED

LLDP for Media Endpoint Devices (LLDP-MED) is an extension to LLDP that operates between endpoint devices such as IP phones and network devices such as switches. It specifically provides support for voice over IP (VoIP) applications and provides additional TLVs for capabilities discovery, network policy, power over Ethernet (PoE), inventory management, and location information. By default, all LLDP-MED TLVs are enabled.

LLDP-MED supports these TLVs:

- LLDP-MED capabilities TLV

Allows LLDP-MED endpoints to determine the capabilities that the connected device supports and what capabilities the device has enabled.

For configuration details, see the [“Configuring Network-Policy Profile” section on page 33-9](#).

- Network policy profile

Allows both network connectivity devices and endpoints to advertise VLAN configurations and associated Layer 2 and Layer 3 attributes for the specific application on that port. For example, the switch can notify a phone of the VLAN number that it should use. The phone can connect into any switch, obtain its VLAN number, and then start communicating with the call control.

By defining a network-policy profile TLV, you can create a profile for voice and voice-signalling by specifying the values for VLAN, class of service (CoS), differentiated services code point (DSCP), and tagging mode. These profile attributes are then maintained centrally on the switch and propagated to the phone.

For configuration details, see the [“Configuring Network-Policy Profile” section on page 33-9](#).

- Power management TLV

Enables advanced power management between LLDP-MED endpoint and network connectivity devices. Allows switches and phones to convey power information, such as how the device is powered, power priority, and how much power the device needs.

For configuration details, see the [“Configuring LLDP Power Negotiation” section on page 33-11](#).

- Inventory management TLV

Allows an endpoint to send detailed inventory information about itself to the switch, including information hardware revision, firmware version, software version, serial number, manufacturer name, model name, and asset ID TLV.

- Location TLV

Provides location information from the switch to the endpoint device. The location TLV can send this information:

- Civic location information

Provides the civic address information and postal information. Examples of civic location information are street address, road name, and postal community name information.

- ELIN location information

Provides the location information for a caller. The location is determined by the emergency location identifier number (ELIN), which is a phone number that routes an emergency call to the local public safety answering point (PSAP) and which the PSAP can use to call back the emergency caller.

For configuration details, see the [“Configuring Location TLV and Location Service” section on page 33-12](#).

**Note**

A switch cannot send LLDP and LLDP-MED simultaneously to an end-point device. By default, a network device sends only LLDP packets until it receives LLDP-MED packets from an end-point device. The network device then sends LLDP-MED packets until it receives only LLDP packets.

Location Service

The location service feature enables the switch to provide location and attachment tracking information for its connected devices to a Cisco Mobility Services Engine (MSE). The tracked device can be a wireless endpoint, a wired endpoint, or a wired switch or controller. The switch informs device link up and link-down events through Network Mobility Services Protocol (NMSP) location and attachment notifications to the MSE.

The MSE initiates the NMSP connection to the switch. When the MSE connects to the switch messages are exchanged to establish version compatibility, service exchange, and location information synchronization. After the connection is established, the switch sends location and attachment notifications periodically to the MSE. Any link-up event, link-down event, or location configuration change detected during the interval are aggregated and sent at the end of the interval using attachment or location notifications.

When the switch discovers the presence or absence of a device on a link-up or link-down event on a port, it obtains the client's MAC address, IP address, and 802.1x username if applicable. If the device is LLDP-MED or CDP enabled, the switch continues to gather client-specific information such as the model number and software version.

Depending on the device capabilities, the switch obtains this client attachment information at link up:

- Slot, port, and port-type
- Client's MAC address
- Client's IP address
- 802.1X username if applicable
- Device category is specified as a *wired station*
- State is specified as *Connected*
- Serial number, UDI
- Model number
- Software version
- VLAN ID and VLAN name

Depending on the device capabilities, the switch obtains this client information at link down:

- Slot and port that was disconnected
- Client's MAC address
- Client's IP address
- 802.1X username if applicable
- Device category is specified as a *wired station*
- State is specified as *Disconnected*
- Serial number, UDI
- Model number
- Software version
- VLAN ID and VLAN name

If an administrator changes a location address at the switch, the information is reported to the MSE. The switch sends a NMSP location notification message that identifies the list of ports affected by the change and the changed address information.

Configuring LLDP and LLDP-MED, and Location Service

This section contains this configuration information:

- [Default LLDP Configuration, page 33-5](#)
- [Configuring LLDP Characteristics, page 33-5](#)
- [Disabling and Enabling LLDP Globally, page 33-6](#)
- [Disabling and Enabling LLDP on an Interface, page 33-7](#)
- [Configuring LLDP-MED TLVs, page 33-8](#)
- [Configuring Network-Policy Profile, page 33-9](#)

- [Configuring LLDP Power Negotiation, page 33-11](#)
- [Configuring Location TLV and Location Service, page 33-12](#)
- [Monitoring and Maintaining LLDP, LLDP-MED, and Location Service, page 33-14](#)

Default LLDP Configuration

Table 33-1 shows the default LLDP configuration. To change the default settings, use the LLDP global configuration and LLDP interface configuration commands.

Table 33-1 **Default LLDP Configuration**

Feature	Default Setting
LLDP global state	Disabled
LLDP holdtime (before discarding)	120 seconds
LLDP timer (packet update frequency)	30 seconds
LLDP reinitialization delay	2 seconds
LLDP tlv-select	Enabled to send and receive all TLVs
LLDP interface state	Enabled
LLDP receive	Enabled
LLDP transmit	Enabled
LLDP med-tlv-select	Enabled to send all LLDP-MED TLVs

Configuring LLDP Characteristics

You can configure the frequency of LLDP updates, the amount of time to hold the information before discarding it, the initialization delay time. You can also select the LLDP and LLDP-MED TLVs for sending and receiving. The location service feature is available only when the switch is running the cryptographic (encrypted) software image.

To configure these characteristics, perform this task:



Note

Steps 2 through 5 can be performed in any order.

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# lldp holdtime <i>seconds</i>	(Optional) Specifies the amount of time a receiving device should hold the information sent by your device before discarding it. The range is 0 to 65535 seconds; the default is 120 seconds.
Step 3	Switch(config)# lldp reinit	(Optional) Specifies the delay time in seconds for LLDP to initialize on any interface. The range is 2 to 5 seconds; the default is 2 seconds.

	Command	Purpose
Step 4	Switch(config)# lldp timer <i>seconds</i>	(Optional) Sets the transmission frequency of LLDP updates in seconds. The range is 5 to 65534 seconds; the default is 30 seconds.
Step 5	Switch(config)# lldp tlv-select	(Optional) Specifies the LLDP TLVs to send or receive.
Step 6	Switch(config)# copy running-config startup-config	Saves your entries in the configuration file.
Step 7	Switch(config)# lldp med-tlv-select	(Optional) Specifies the LLDP-MED TLVs to send or receive.



Note

Use the **no** form of each of the LLDP commands to return to the default setting.

This example shows how to configure a holdtime of 120 seconds, a delay time of 2 seconds and an update frequency of 30:

```
Switch# configure terminal
Switch(config)# lldp holdtime 120
Switch(config)# lldp reinit 2
Switch(config)# lldp timer 30
Switch(config)# end
```

This example shows how to transmit only LLDP packets:

```
Switch# configure terminal
Switch(config)# no lldp receive
Switch(config)# end
```

This example shows how to receive LLDP packets again:

```
Switch# configure terminal
Switch(config)# lldp receive
Switch(config)# end
```

For additional LLDP **show** commands, see the [“Monitoring and Maintaining LLDP, LLDP-MED, and Location Service”](#) section on page 33-14.

Disabling and Enabling LLDP Globally



Note

LLDP is disabled by default.

To disable LLDP globally, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# no lldp run	Disables LLDP.
Step 3	Switch(config)# end	Returns to privileged EXEC mode.

To enable LLDP once it has been disabled, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# lldp run	Enables LLDP.
Step 3	Switch(config)# end	Returns to privileged EXEC mode.

This example shows how to globally disable LLDP:

```
Switch# configure terminal
Switch(config)# no lldp run
Switch(config)# end
```

This example shows how to globally enable LLDP:

```
Switch# configure terminal
Switch(config)# lldp run
Switch(config)# end
```

Disabling and Enabling LLDP on an Interface

LLDP is disabled globally on all supported interfaces. You must enable LLDP globally to allow a device to send LLDP packets. However, no changes are required at the interface level.

You can configure the interface to selectively not to send and receive LLDP packets with the **no lldp transmit** and **no lldp receive** commands.



Note

If the interface is configured as a tunnel port, LLDP is automatically disabled.

To disable LLDP on an interface, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface interface-id	Specifies the interface on which you are disabling LLDP, and enter interface configuration mode.
Step 3	Switch(config)# no lldp transmit	Disallows sending LLDP packets on the interface.
Step 4	Switch(config)# no lldp receive	Disallows receiving LLDP packets on the interface.
Step 5	Switch(config)# end	Returns to privileged EXEC mode.
Step 6	Switch(config)# copy running-config startup-config	Saves your entries in the configuration file.

To enable LLDP on an interface once it has been disabled, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface interface-id	Specifies the interface on which you are enabling LLDP, and enter interface configuration mode.

	Command	Purpose
Step 3	Switch(config)# lldp transmit	Sends LLDP packets on the interface.
Step 4	Switch(config)# lldp receive	Receives LLDP packets on the interface.
Step 5	Switch(config)# end	Returns to privileged EXEC mode.
Step 6	Switch# copy running-config startup-config	Saves your entries in the configuration file.

This example shows how to enable LLDP on an interface:

```
Switch# configure terminal
Switch(config)# interface GigabitEthernet 1/1
Switch(config-if)# lldp transmit
Switch(config-if)# lldp receive
Switch(config-if)# end
```

Configuring LLDP-MED TLVs

By default, the switch only sends LLDP packets until it receives LLDP-MED packets from the end device. The switch continues to send LLDP-MED packets until it only receives LLDP packets.

By using the **lldp** interface configuration command, you can configure the interface not to send the TLVs listed in [Table 33-2](#).

Table 33-2 LLDP-MED TLVs

LLDP-MED TLV	Description
inventory-management	LLDP-MED inventory management TLV
location	LLDP-MED location TLV
network-policy	LLDP-MED network policy TLV
power-management	LLDP-MED power management TLV

To disable a TLV on an interface, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface interface-id	Specifies the interface on which you are configuring a LLDP-MED TLV, and enter interface configuration mode.
Step 3	Switch(config-if)# no lldp med-tlv-select tlv	Specifies the TLV to disable.
Step 4	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 5	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To enable a TLV on an interface, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Specifies the interface on which you are configuring an LLDP-MED TLV, and enter interface configuration mode.
Step 3	Switch(config-if)# lldp med-tlv-select <i>tlv</i>	Specifies the TLV to enable.
Step 4	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 5	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to enable a TLV on an interface when it has been disabled:

```
Switch# configure terminal
Switch(config)# interface GigabitEthernet0/1
Switch(config-if)# lldp med-tlv-select inventory management
Switch(config-if)# end
```

Configuring Network-Policy Profile

To create a network-policy profile, configure the policy attributes, and apply it to an interface, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# network-policy profile <i>profile number</i>	Specifies the network-policy profile number, and enter network-policy configuration mode. The range is 1 to 4294967295.

	Command	Purpose
Step 3	Switch(config-network-policy)# {voice voice-signaling} vlan [vlan-id {cos cvalue dscp dvalue}] [[dot1p {cos cvalue dscp dvalue}] none untagged]	Configures the policy attributes: voice —Specifies the voice application type. voice-signaling —Specifies the voice-signaling application type. vlan —Specifies the native VLAN for voice traffic. vlan-id —(Optional) Specifies the VLAN for voice traffic. The range is 1 to 4094. cos cvalue —(Optional) Specifies the Layer 2 priority class of service (CoS) for the configured VLAN. The range is 0 to 7; the default is 0. dscp dvalue —(Optional) Specifies the differentiated services code point (DSCP) value for the configured VLAN. The range is 0 to 63; the default is 0. dot1p —(Optional) Configures the telephone to use IEEE 802.1p priority tagging and use VLAN 0 (the native VLAN). none —(Optional) Do not instruct the IP telephone about the voice VLAN. The telephone uses the configuration from the telephone key pad. untagged —(Optional) Configures the telephone to send untagged voice traffic (the default for the telephone).
Step 4	Switch(config)# exit	Returns to global configuration mode.
Step 5	Switch# configure terminal	Enters global configuration mode.
Step 6	Switch(config)# interface interface-id	Specifies the interface on which you are configuring a network-policy profile, and enter interface configuration mode.
Step 7	Switch(config-if)# network-policy profile number	Specifies the network-policy profile number.
Step 8	Switch(config-if)# lldp med-tlv-select network-policy	Specifies the network-policy TLV.
Step 9	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 10	Switch# show network-policy profile	Verifies the configuration.
Step 11	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Use the **no** form of each command to return to the default setting.

This example shows how to configure VLAN 100 for voice application with CoS and to enable the network-policy profile and network-policy TLV on an interface:

```
Switch# configure terminal
Switch(config)# network-policy profile 1
Switch(config-network-policy)# voice vlan 100 cos 4
Switch(config)# exit
Switch# configure terminal
Switch# interface_id
Switch(config-if)# network-policy profile 1
Switch(config-if)# lldp med-tlv-select network-policy
```

This example shows how to configure the voice application type for the native VLAN with priority tagging:

```
Switch(config-network-policy)# voice vlan dot1p cos 4
```

```
Switch(config-network-policy)# voice vlan dot1p dscp 34
```

**Note**

As of Cisco IOS Release 12.2(54)SG, the Catalyst 4500 series switch supports only 2 applications: voice and voice signaling. The default cos/dscp values for a voice application is 5/46 and for voice signaling is 3/24. You must configure the network policy profile and attach it to the interface if you need to override the default values. These values are sent as a part of the network-policy TLV in LLDP MED.

Configuring LLDP Power Negotiation

Starting with Cisco IOS Release 12.2(54)SG, Catalyst 4500 series switches can perform inline power negotiation using LLDP as specified in the IEEE 802.3at standard. (The LLDP TLV used is DTE Power-via-MDI TLV.) With this feature, inline powered devices based on the IEEE standard can be powered in the PoE+ power range (12.95W to 25.5W at the device end) by the switch on PoE+ supported modules.

**Note**

To verify inline power utilization negotiated by using LLDP using the LLDP-MED TLV, use the **show lldp neighbors detail** command. To verify inline power utilization negotiated by using the IEEE 802.3at TLV, use the **show power inline interface detail** command. The **show power inline interface detail** command does not display power negotiated with LLDP.

**Note**

When an inline powered device that performs power negotiation using multiple protocols (CDP/LLDP 802.3at/LLDP-MED) is connected to a switch, the switch locks to the first protocol packet (CDP or LLDP) that contains the power negotiation TLV. The LLDP 802.3at power negotiation TLV overrides the LLDP-MED power negotiation TLV if both are received by the switch. If you need to use any single protocol for power negotiation each time, you must administratively disable the other power negotiation protocols on the switch interface or the end device.

To enable LLDP power negotiation, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Specifies the interface on which you are configuring LLDP power negotiation.
Step 3	Switch(config-if)# lldp tlv-select power-management	Enables LLDP power negotiation.
Step 4	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 5	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to enable LLDP power negotiation on interface Gigabit Ethernet 3/1:

```
Switch# config t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# int gi 3/1
Switch(config-if)# lldp tlv-select power-management
```

Configuring Location TLV and Location Service

To configure location information for an end-point and to apply it to an interface, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# location { admin-tag <i>string</i> civic-location identifier <i>id</i> elin-location <i>string</i> identifier <i>id</i> }	<p>Specifies the location information for an endpoint.</p> <ul style="list-style-type: none"> • admin-tag—Specify an administrative tag or site information. • civic-location—Specify civic location information. <p>Note The civic location identifier in the LLDP-MED TLV is limited to 250 bytes or less. To avoid receiving error messages regarding available buffer space during switch configuration, never allow the total length of all civic location information specified for each civic-location identifier to exceed 250 bytes.</p> <ul style="list-style-type: none"> • elin-location—Specify emergency location information (ELIN). • identifier <i>id</i>—Specify the ID for the civic location. • <i>string</i>—Specify the site or location information in alphanumeric format.
Step 3	Switch(config-civic)# exit	Returns to global configuration mode.
Step 4	Switch(config)# interface <i>interface-id</i>	Specifies the interface on which you are configuring the location information, and enter interface configuration mode.
Step 5	Switch(config-if)# location { additional-location-information <i>word</i> / civic-location-id <i>id</i> elin-location-id <i>id</i> }	<p>Enters location information for an interface:</p> <p>additional-location-information—Specifies additional information for a location or place.</p> <p>civic-location-id—Specifies global civic location information for an interface.</p> <p>elin-location-id—Specifies emergency location information for an interface.</p> <p><i>id</i>—Specifies the ID for the civic location or the ELIN location. The ID range is 1 to 4095.</p> <p><i>word</i>—Specifies a word or phrase that provides additional location information.</p>
Step 6	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 7	Switch# show location	Verifies the configuration.
Step 8	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Use the **no** form of each command to return to the default setting.

This example shows how to configure civic location information on the switch:

```
Switch# configure terminal
Switch(config)# location civic-location identifier 1
Switch(config-civic)# number 3550
Switch(config-civic)# primary-road-name "Cisco Way"
Switch(config-civic)# city "San Jose"
Switch(config-civic)# state CA
Switch(config-civic)# building 19
Switch(config-civic)# room C6
Switch(config-civic)# county "Santa Clara"
Switch(config-civic)# country US
Switch(config-civic)# end
```

To enable location service on the switch, perform this task:



Note

Your switch must be running the cryptographic (encrypted) software image in order to enable the location service feature. Your Cisco Mobility Service Engine (MSE) must be running Heitz 6.0 or later software image to support wired location service

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# nmosp enable	Enables the NMSP features on the switch.
Step 3	Switch(config)# ip device tracking	Enables IP device tracking.
Step 4	Switch(config)# nmosp notification interval {attachment location} interval-seconds	Specifies the NMSP notification interval. attachment —Specify the attachment notification interval. location —Specify the location notification interval. <i>interval-seconds</i> —Duration in seconds before a switch sends the location or attachment updates to the MSE. The range is 1 to 30; the default is 30.
Step 5	Switch(config)# interface interface-id	Specifies the interface on which you want to prevent all learned attachment information from being sent to the MSE.
Step 6	Switch(config-if)# nmosp attachment suppress	Prevents the attachment information learned on this interface from being sent to the MSE. Note Location service is intended for tracking end devices directly connected to the switch. Please apply this command on all trunk interfaces to avoid inaccurate tracking information.
Step 7	Switch# end	Returns to privileged EXEC mode.
Step 8	Switch# show nmosp notification	Verifies the notification interval configuration.
Step 9	Switch# show nmosp status	Verifies if NMSP is enabled.
Step 10	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to enable NMSP on a switch and set the location notification time to 10 seconds:

```
Switch# configure terminal
Switch(config)# nmosp enable
Switch(config)# ip device tracking
Switch(config)# nmosp notification interval location 10
Switch(config)# end
```


Note

Location service tracks IP devices only on Layer 2 and Layer 3 physical ports. IP devices that are connected through SVIs or port-channels are not tracked and reported to the MSE.

Monitoring and Maintaining LLDP, LLDP-MED, and Location Service

To monitor and maintain LLDP, LLDP-MED, and location service on your device, perform one or more of the following commands in privileged EXEC mode:

Command	Description
clear lldp counters	Resets the traffic and error counters to zero.
clear lldp table	Deletes the LLDP table of information about neighbors.
clear nmosp statistics	Clears the NMSP statistic counters.
show lldp	Displays global information, such as frequency of transmissions, the holdtime for packets being sent, and the delay time for LLDP to initialize on an interface.
show lldp entry <i>entry-name</i>	Displays information about a specific neighbor. You can enter an asterisk (*) to display all neighbors, or you can enter the name of the neighbor about which you want information.
show lldp errors	Displays LLDP computational errors and overflows.
show lldp interface [<i>interface-id</i>]	Displays information about interfaces where LLDP is enabled. You can limit the display to the interface about which you want information.
show lldp neighbors [<i>interface-id</i>] [<i>detail</i>]	Displays information about neighbors, including device type, interface type and number, holdtime settings, capabilities, and port ID. You can limit the display to neighbors of a specific interface or expand the display to provide more detailed information.
show lldp traffic	Displays LLDP counters, including the number of packets sent and received, number of packets discarded, and number of unrecognized TLVs.
show location	Displays the location information for an endpoint.
show nmosp	Displays the NMSP information.
show power inline interface [<i>detail</i>]	Displays detailed information on the PoE status for the interface
show power inline module <i>mod</i> [<i>detail</i>]	Displays detailed information on the PoE consumption for the specified module.

Cisco IOS Carries Ethernet Features in Cisco IOS XE 3.1.0SG

This section provides a list of High Availability software features that are supported in Cisco IOS XE 3.1.0SG. Links to the feature documentation are included.

Feature guides may contain information about more than one feature. To find information about a specific feature within a feature guide, see the Feature Information table at the end of the guide.

Feature guides document features that are supported on many different software releases and platforms. Your Cisco software release or platform may not support all the features documented in a feature guide. See the Feature Information table at the end of the feature guide for information about which features in that guide are supported in your software release. Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

ANSI TIA-1057 LLDP-MED Support and IEEE 802.1ab LLDP (Link Layer Discovery Protocol)

http://www.cisco.com/en/US/docs/ios/cether/configuration/guide/ce_lddp-med.html



Configuring UDLD

This chapter describes how to configure UniDirectional Link Detection (UDLD) Ethernet on the switch, and includes the following major sections:

- [About UDLD, page 34-1](#)
- [Default UDLD Configuration, page 34-3](#)
- [Configuring UDLD on the Switch, page 34-4](#)
- [Displaying UDLD Link Status, page 34-9](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About UDLD

UDLD is a Layer 2 protocol that initializes devices connected through fiber-optic or twisted-pair Ethernet cables. This protocol monitors a physical connection (such as wrong cabling) to detect unidirectional links to avoid spanning-tree topology loops or silent drop traffic.

All connected devices must support UDLD for the protocol to successfully identify the unidirectional links. When UDLD detects a unidirectional link, it can administratively shut down the affected port and send you a warning message.

With UDLD, the time to detect a unidirectional link can vary from a few seconds to several minutes depending on how the timers are configured. Link status messages are exchanged every couple of seconds.

Starting with Cisco IOS Release 12.2(54)SG, the enhancement Fast UDLD was added, which supports timers in the few-hundred milliseconds range, which enables subsecond unidirectional link detection. With Fast UDLD, the time to detect a unidirectional link can vary from less than one second to a few seconds (the detection time also depends on how the timers are configured). Link status messages are exchanged every couple of hundred milliseconds.

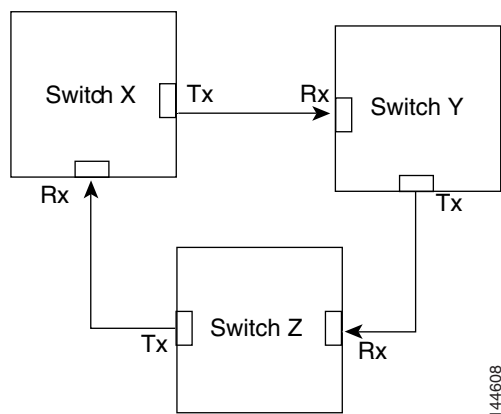
This section includes these topics:

- [UDLD Topology, page 34-2](#)
- [Fast UDLD Topology, page 34-2](#)
- [Operation Modes, page 34-3](#)
- [Default States for UDLD, page 34-3](#)

UDLD Topology

Each switch can send packets to a neighbor switch but cannot receive packets from the switch it is sending packets to. UDLD detects and disables these one-way connections. [Figure 34-1](#) illustrates a unidirectional link condition.

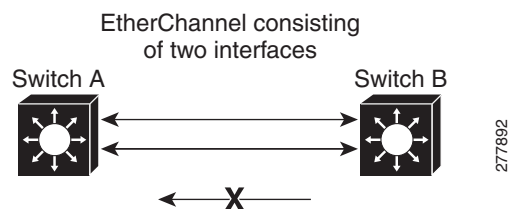
Figure 34-1 Unidirectional Link Topology



Fast UDLD Topology

[Figure 34-2](#) illustrates a typical Fast UDLD topology. Switch A and B are connected through a 2-port EtherChannel, and Fast UDLD is enabled on the individual ports. If one of the links becomes unidirectional, Fast UDLD detects this situation faster than regular UDLD and erdisables the link. Traffic is switched over to the second link by EtherChannel. Because this occurs very quickly, traffic loss is minimized.

Figure 34-2 Fast UDLD Topology



Note

For Fast UDLD, Supervisor Engine 6-E, Supervisor 6L-E, Supervisor 7-E, and Supervisor Engine 7L-E support up to 32 ports.

Operation Modes

UDLD and Fast UDLD support the following operation modes:

- **Normal**—A UDLD-capable port (A) periodically sends a UDLD probe to a second port (B). If B is not UDLD capable, no unidirectional link detection occurs. If both devices are UDLD capable and bidirectional connectivity exists, probe messages travel in both directions at the rate of the configured message time interval. When the UDLD protocol receives the probe, it attempts to synchronize the devices by sending echo messages to the peer port and waiting for an answer during the detection window. If unidirectional traffic is detected when the port link is still up (B longer sends traffic to A), B enters errdisable mode, and A is marked undetermined but does not enter errdisable mode. It continues to operate under its current STP status because this mode is informational only; it is potentially less disruptive although it does not prevent STP loops.



Note Bidirectional link failures cannot be detected using normal mode.

- **Aggressive**—If a port (A) loses its neighbor connectivity, it actively attempts to reestablish the relationship by sending a probe to a second port (B). If port B does not respond, the link is considered unidirectional and port A enters an errdisable state to avoid silent drop traffic.



Note Both unidirectional and bidirectional link failures can be detected in aggressive mode.

UDLD aggressive mode can interoperate with UDLD normal mode. When a unidirectional condition is detected, only the aggressive mode link shuts down.

Default States for UDLD

The following are the defaults for UDLD:

- UDLD is locally disabled on copper LAN ports to avoid sending unnecessary control traffic (BPDU control packets). This protocol is commonly used for access ports.
- UDLD is enabled on a fiber port if global UDLD is activated.
- Fast UDLD is disabled on all ports.

Default UDLD Configuration

Table 34-1 shows the UDLD default configuration.

Table 34-1 UDLD Default Configuration

Feature	Default Status
UDLD global enable state	Globally disabled.
UDLD per-interface enable state for fiber-optic media	Enabled on all Ethernet fiber-optic interfaces.
UDLD per-interface enable state for twisted-pair (copper) media	Disabled on all Ethernet 10/100 and 1000BASE-TX interfaces.
Fast UDLD per-interface enable state.	Disabled on all interfaces.

Configuring UDLD on the Switch

The following sections describe how to configure UDLD:

- [Fast UDLD Guidelines and Restrictions, page 34-4](#)
- [Enabling UDLD Globally, page 34-5](#)
- [Enabling UDLD on Individual Interfaces, page 34-6](#)
- [Disabling UDLD on Individual Interfaces, page 34-7](#)
- [Disabling UDLD on a Fiber-Optic Interface, page 34-7](#)
- [Configuring a UDLD Probe Message Interval Globally, page 34-8](#)
- [Resetting Disabled LAN Interfaces, page 34-8](#)

Fast UDLD Guidelines and Restrictions

When using (or configuring) Fast UDLD, consider these guidelines and restrictions:

- Fast UDLD is disabled by default.
- Configure fast UDLD only on point-to-point links between network devices that support fast UDLD.
- You can configure fast UDLD in either normal or aggressive mode.
- Do not enter the link debounce command on fast UDLD ports.
- Configure fast UDLD on at least two links between each connected network device. This reduces the number of link disablements due to false positives.
- Fast UDLD does not report a unidirectional link if the same error occurs simultaneously on more than one link to the same neighbor device.
- Fast UDLD is supported on a limited number of ports.

Enabling UDLD Globally

To enable UDLD in aggressive or normal mode and to set the configurable message timer on all fiber-optic interfaces on the switch, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters the global configuration mode.
Step 2	Switch(config)# udld {aggressive enable message time message-timer-interval} fast-hello error-reporting	<p>Specifies UDLD and Fast UDLD operation:</p> <ul style="list-style-type: none"> • aggressive —Enables UDLD in aggressive mode on all fiber-optic interfaces. • enable —Enables UDLD in normal mode on all fiber-optic interfaces on the switch. UDLD is disabled by default. <p>An individual interface configuration overrides the setting of the udld enable global configuration command.</p> <p>For more information about aggressive and normal modes, see the “Operation Modes” section on page 34-3.</p> <ul style="list-style-type: none"> • message time —<i>message-timer-interval</i>— Configures the period of time between UDLD probe messages on ports that are in the advertisement phase and are determined to be bidirectional. The range is from 1 to 90 seconds. <p>Note Prior to Cisco IOS Release 12.2(31)SGA, the timer range is 7 to 90 seconds. With Cisco IOS Release 12.2(31)SGA, the timer range is 1 to 90 seconds.</p> <ul style="list-style-type: none"> • fast-hello error reporting—If configured, Fast UDLD does not errdisable a unidirectional link. Instead, a log message informing link failure is displayed on the console (behavior for fast UDLD only).
Step 3	Switch(config)# end	Returns to privileged EXEC mode.
Step 4	Switch# show udld	Verifies the configuration.

Enabling UDLD on Individual Interfaces

To enable UDLD on individual interfaces, perform this task:

	Command	Purpose
Step 1	Switch(config-if)# udld port	Enables UDLD in normal mode on a specific interface. On a fiber-optic interface, this command overrides the udld enable global configuration command setting.
	Switch(config-if)# udld port aggressive	Enables UDLD in aggressive mode on a specific interface. On a fiber-optic interface, this command overrides the udld enable global configuration command setting.
	Switch(config-if)# udld fast-hello interval	<p>Enables Fast UDLD on the interface with message interval equal to the <i>interval</i> value in milliseconds.</p> <p>The interval value range is from 200 milliseconds to 1000 milliseconds.</p> <p>To enable Fast UDLD, UDLD must be enabled (explicitly configured or globally enabled) and operational (in bidirectional state) on the interface.</p> <p>Note Fast UDLD can only be enabled on individual interfaces (a global enable command does not exist).</p> <p>Note Fast UDLD can only be configured or enabled on a limited number of interfaces that depend on the type of supervisor installed. The number of supported interfaces for Fast UDLD can be displayed with the show udld fast-hello command.</p>
Step 2	Switch# show udld interface	Verifies the configuration.

Disabling UDLD on Individual Interfaces

To disable UDLD on individual interfaces, perform this task:

	Command	Purpose
Step 1	Switch(config-if)# no udld port	Disables UDLD on an interface. The following applies: <ul style="list-style-type: none"> On fiber-optic interfaces, the no udld port command reverts the interface configuration to the setting established with the udld enable global configuration command. For both UDLD and Fast UDLD, if aggressive mode is configured, then aggressive mode must be explicitly disabled with the no udld port aggressive command. If normal mode is configured, the no udld port command disables both UDLD and Fast UDLD.
	Switch(config-if)# no udld fast-hello	Disables Fast UDLD on an interface. The interface reverts to the UDLD configuration that was present before you enabled Fast UDLD.
Step 2	Switch# show udld interface	Verifies the configuration.

Disabling UDLD on a Fiber-Optic Interface

To disable UDLD on individual fiber-optic interfaces, perform this task:

	Command	Purpose
Step 1	Switch(config-if)# udld port disable	Disables UDLD on a fiber-optic interface and removes all UDLD and Fast UDLD related configuration on the interface. Note You can enable UDLD globally for all fiber-optic interfaces.
	Switch(config-if)# no udld fast-hello	Disables Fast UDLD on an interface, which reverts to the UDLD configuration that was present before you enabled Fast UDLD.
Step 2	Switch# show udld interface	Verifies the configuration.

Configuring a UDLD Probe Message Interval Globally

To configure the time between UDLD probe messages on ports that are in advertisement mode and are currently determined to be bidirectional, perform this task:

	Command	Purpose
Step 1	Switch(config)# udld message time interval	Configures the time between UDLD probe messages on ports that are in advertisement mode and are currently determined to be bidirectional; valid values are from 1 to 90 seconds. Note Prior to Cisco IOS Release 12.2(31)SGA, the time interval is 7 to 90 seconds. With Cisco IOS Release 12.2(31)SGA, the time interval is 1 to 90 second. The no udld message command returns the default value (15 seconds).
Step 2	Switch# show udld type-slot/interface	Verifies the configuration.

Enabling Fast UDLD Error Reporting

By default, fast UDLD error-disables ports with unidirectional links. You can globally enable fast UDLD to report unidirectional links with a message displayed on the console instead of error-disabling ports with unidirectional links.



Note

When fast UDLD error reporting is enabled, you must manually take the action appropriate for the state of the link.

To globally enable fast UDLD error reporting, perform this task:

Command	Purpose
Switch(config)# udld fast-hello error-reporting	Enables fast UDLD error reporting.

Resetting Disabled LAN Interfaces

To reset all LAN ports that have been errdisabled by UDLD, use this command:

Command	Purpose
Switch(config)# udld reset	Resets all LAN ports that have been errdisabled by UDLD and Fast UDLD.

Displaying UDLD Link Status

To verify link status reported by UDLD, enter the following command:

```
Switch# show udld neighbors
```

Port	Device Name	Device ID	Port ID	Neighbor State
Gi1/33	FOX10430380	1	Gi1/33	Bidirectional
Gi1/34	FOX10430380	1	Gi1/34	Bidirectional

To verify status for a particular link as reported by UDLD, enter the following command:

```
Switch# show udld g1/34
```

```
Interface Gi1/34
```

```
---
```

```
Port enable administrative configuration setting: Enabled / in aggressive mode
```

```
Port enable operational state: Enabled / in aggressive mode
```

```
Current bidirectional state: Bidirectional
```

```
Current operational state: Advertisement - Single neighbor detected
```

```
Message interval: 15000 ms
```

```
Time out interval: 5000 ms
```

```
Port fast-hello configuration setting: Disabled
```

```
Port fast-hello interval: 0 ms
```

```
Port fast-hello operational state: Disabled
```

```
Neighbor fast-hello configuration setting: Disabled
```

```
Neighbor fast-hello interval: Unknown
```

```
Entry 1
```

```
---
```

```
Expiration time: 43300 ms
```

```
Cache Device index: 1
```

```
Current neighbor state: Bidirectional
```

```
Device ID: FOX10430380
```

```
Port ID: Gi1/34
```

```
Neighbor echo 1 device: FOX104303NL
```

```
Neighbor echo 1 port: Gi1/34
```

```
TLV Message interval: 15 sec
```

```
No TLV fast-hello interval
```

```
TLV Time out interval: 5
```

```
TLV CDP Device name: Switch
```

To verify link status reported by Fast UDLD, enter the following command:

```
Switch# show udld fast-hello
```

```
Total ports on which fast hello can be configured: 16
```

```
Total ports with fast hello configured: 3
```

```
Total ports with fast hello operational: 3
```

```
Total ports with fast hello non-operational: 0
```

Port-ID	Hello	Neighbor-Hello	Neighbor-Device	Neighbor-Port	Status
Gi1/45	200	200	FOX104303NL	Gi1/45	Operational
Gi1/46	200	200	FOX104303NL	Gi1/46	Operational
Gi1/47	200	200	FOX104303NL	Gi1/47	Operational

To verify status for a particular link as reported by Fast UDLD, enter the following command:

```
Switch# show udld fast-hello gi1/33
```

```
Interface Gi1/33
```

```
---
```

```
Port enable administrative configuration setting: Enabled / in aggressive mode
```

```
Port enable operational state: Enabled / in aggressive mode
```

```
Current bidirectional state: Bidirectional
```

```
Current operational state: Advertisement - Single neighbor detected
```

```
Message interval: 200 ms
```

```
Time out interval: 5000 ms
```

```
Port fast-hello configuration setting: Enabled
```

```
Port fast-hello interval: 200 ms
```

```
Port fast-hello operational state: Enabled
```

```
Neighbor fast-hello configuration setting: Enabled
```

```
Neighbor fast-hello interval: 200 ms
```

```
Entry 1
```

```
---
```

```
Expiration time: 500 ms
```

```
Cache Device index: 1
```

```
Current neighbor state: Bidirectional
```

```
Device ID: FOX10430380
```

```
Port ID: Gi1/33
```

```
Neighbor echo 1 device: FOX104303NL
```

```
Neighbor echo 1 port: Gi1/33
```

```
TLV Message interval: 15
```

```
TLV fast-hello interval: 200 ms
```

```
TLV Time out interval: 5
```

```
TLV CDP Device name: Switch
```



Configuring Unidirectional Ethernet



Note

Unidirectional Ethernet is not supported on the following: Supervisor Engine 7-E and Supervisor Engine 7L-E uplinks.

This chapter describes how to configure Unidirectional Ethernet on the Catalyst 4500 series switch and contains these sections:

- [About Unidirectional Ethernet, page 35-1](#)
- [Configuring Unidirectional Ethernet, page 35-1](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About Unidirectional Ethernet

You can set stubless Gigabit Ethernet ports to unidirectionally transmit or receive traffic. Unidirectional Ethernet uses only one strand of fiber for either transmitting or receiving one-way traffic for the Gigabit Ethernet Port, instead of two strands of fiber for a full-duplex Gigabit Ethernet Port. Configuring your Gigabit Ethernet Port either to transmit or receive traffic effectively doubles the amount of traffic capabilities for applications, such as video streaming, where most traffic is sent as unacknowledged unidirectional video broadcast streams.

Configuring Unidirectional Ethernet



Note

You must configure Unidirectional Ethernet on the non-blocking Gigabit Ethernet Port, which automatically disables UDLD on the port.

To enable Unidirectional Ethernet, perform this task:

	Command	Purpose
Step 1	Switch(config)# interface { vlan <i>vlan_ID</i> { fastethernet gigabitethernet tengigabitethernet } <i>slot/interface</i> Port-channel <i>number</i> }	Selects the interface to configure.
Step 2	Switch(config-if)# [no] unidirectional { send-only receive-only }	Enables Unidirectional Ethernet. Use the no keyword to disable Unidirectional Ethernet.
Step 3	Switch(config-if)# end	Exits configuration mode.
Step 4	Switch# show interface { vlan <i>vlan_ID</i> { fastethernet gigabitethernet tengigabitethernet } <i>slot/interface</i> } unidirectional	Verifies the configuration.

This example shows how to set Gigabit Ethernet interface 1/1 to unidirectionally send traffic:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface gigabitethernet 1/1
Switch(config-if)# unidirectional send-only
Switch(config-if)# end
```

Warning!

Enable 12 port unidirectional mode will automatically disable port udld.
You must manually ensure that the unidirectional link does not create
a spanning tree loop in the network.

Enable 13 port unidirectional mode will automatically disable ip routing
on the port. You must manually configure static ip route and arp entry
in order to route ip traffic.

This example shows how to set Gigabit Ethernet interface 1/1 to receive traffic unidirectionally:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface gigabitethernet 1/1
Switch(config-if)# unidirectional receive-only
Switch(config-if)# end
```

Warning!

Enable 12 port unidirectional mode will automatically disable port udld.
You must manually ensure that the unidirectional link does not create
a spanning tree loop in the network.

Enable 13 port unidirectional mode will automatically disable ip routing
on the port. You must manually configure static ip route and arp entry
in order to route ip traffic.

This example shows how to verify the configuration:

```
Switch> show interface gigabitethernet 1/1 unidirectional
show interface gigabitethernet 1/1 unidirectional
Unidirectional configuration mode: send only
CDP neighbor unidirectional configuration mode: receive only
```

This example shows how to disable Unidirectional Ethernet on Gigabit Ethernet interface 1/1:

```
Switch# configure terminal  
Enter configuration commands, one per line. End with CNTL/Z.  
Switch(config)# interface gigabitethernet 1/1  
Switch(config-if)# no unidirectional  
Switch(config-if)# end
```

This example shows the result of entering the **show interface** command for a port that does not support Unidirectional Ethernet:

```
Switch# show interface f6/1 unidirectional  
Unidirectional Ethernet is not supported on FastEthernet6/1
```




Configuring Layer 3 Interfaces

This chapter describes the Layer 3 interfaces on a Catalyst 4500 series switch. It also provides guidelines, procedures, and configuration examples.

This chapter includes the following major sections:

- [About Layer 3 Interfaces, page 36-1](#)
- [Configuration Guidelines, page 36-5](#)
- [Configuring Logical Layer 3 VLAN Interfaces, page 36-7](#)
- [Configuring Logical Layer 3 GRE Tunnel Interfaces, page 36-6](#)
- [Configuring VLANs as Layer 3 Interfaces, page 36-8](#)
- [Configuring Physical Layer 3 Interfaces, page 36-13](#)
- [Configuring Multipoint GRE, page 36-14](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About Layer 3 Interfaces

The Catalyst 4500 series switch supports Layer 3 interfaces with the Cisco IOS IP and IP routing protocols. Layer 3, the *network* layer, is primarily responsible for the routing of data in packets across logical internetwork paths.

Layer 2, the *data link* layer, contains the protocols that control the *physical* layer (Layer 1) and how data is framed before being transmitted on the medium. The Layer 2 function of filtering and forwarding data in frames between two segments on a LAN is known as *bridging*.

The Catalyst 4500 series switch supports two types of Layer 3 interfaces. The logical Layer 3 VLAN interfaces integrate the functions of routing and bridging. The physical Layer 3 interfaces allow the Catalyst 4500 series switch to be configured like a traditional router.



Note

On a Catalyst 4500 series switch, a physical Layer 3 interface has MAC address learning enabled.

This section contains the following subsections:

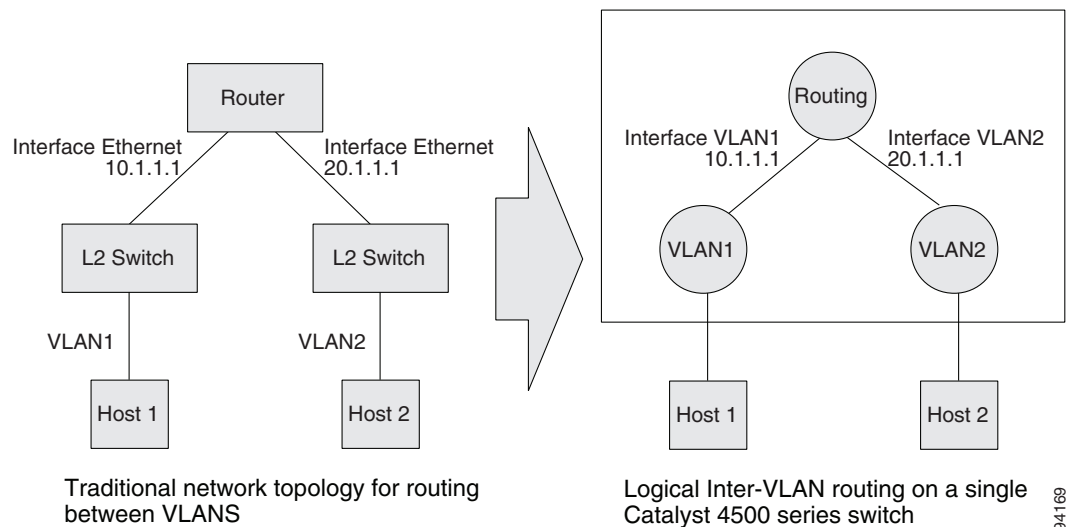
- [Logical Layer 3 VLAN Interfaces, page 36-2](#)
- [Physical Layer 3 Interfaces, page 36-2](#)
- [Understanding SVI Autostate Exclude, page 36-3](#)
- [Understanding Layer 3 Interface Counters, page 36-3](#)

Logical Layer 3 VLAN Interfaces

The logical Layer 3 VLAN interfaces provide logical routing interfaces to VLANs on Layer 2 switches. A traditional network requires a physical interface from a router to a switch to perform inter-VLAN routing. The Catalyst 4500 series switch supports inter-VLAN routing by integrating the routing and bridging functions on a single Catalyst 4500 series switch.

[Figure 36-1](#) shows how the routing and bridging functions in the three physical devices of the traditional network are performed logically on one Catalyst 4500 series switch.

Figure 36-1 Logical Layer 3 VLAN Interfaces for the Catalyst 4500 Series Switch

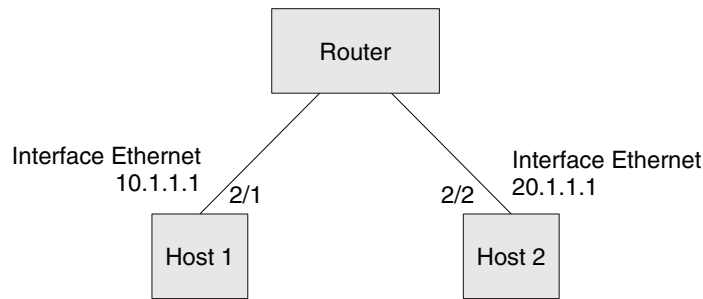


94169

Physical Layer 3 Interfaces

The physical Layer 3 interfaces support capabilities equivalent to a traditional router. These Layer 3 interfaces provide hosts with physical routing interfaces to a Catalyst 4500 series switch.

[Figure 36-2](#) shows how the Catalyst 4500 series switch functions as a traditional router.

Figure 36-2 Physical Layer 3 Interfaces for the Catalyst 4500 Series Switch

Physical Inter-VLAN Routing on a Catalyst 4500 series switch 94168

Understanding SVI Autostate Exclude

To be up/up, a router VLAN interface must fulfill the following general conditions:

- The VLAN exists and is active on the VLAN database of the switch.
- The VLAN interface exists on the router and is not administratively down.
- At least one Layer 2 (access port or trunk) port exists, has a link up on this VLAN, and is in spanning-tree forwarding state on the VLAN.



Note

The protocol line state for the VLAN interfaces comes up when the first switch port belonging to the corresponding VLAN link comes up and is in spanning-tree forwarding state.

Ordinarily, when a VLAN interface has multiple ports in the VLAN, the SVI goes down when all the ports in the VLAN go down. The SVI Autostate Exclude feature provides a knob to mark a port so that it is not counted in the SVI up and down calculation and applies to all VLANs that are enabled on that port.

A VLAN interface is brought up after the Layer 2 port has had time to converge (that is, transition from listening-learning to forwarding). This prevents routing protocols and other features from using the VLAN interface as if it were fully operational. It also prevents other problems from occurring, such as routing black holes.

Understanding Layer 3 Interface Counters



Note

Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, 7-E, 6L-E, do not support Layer 2 interface counters. However, they do support Layer 3 (SVI) interface counters.

When you run IPv4 and IPv6 on Supervisor Engines 9-E, 8L-E, 8-E, 7-LE, 7-E, 6L-E, 6-E, packets are routed in hardware by the forwarding engine. They support the following statistics for counting routed packets with a maximum of 4092 interfaces:

- Input unicast
- Input multicast

- Output unicast
- Output multicast

For each counter type, both the number of packets and the total number of bytes received or transmitted are counted. You can collect these statistics uniquely for IPv4 and IPv6 traffic.

Because the total number of supported Layer 3 interfaces exceeds the number of counters supported by hardware, all Layer 3 interfaces might not have counters. You assign counters to Layer 3 interfaces; the default configuration for a Layer 3 interface has no counters.

You can configure collection statistics at an interface level in one of the four ways (see [Table 36-1](#)). The maximum number of interfaces applied to the configuration depends on the collection mode.

Table 36-1 **Configuring Statistics Collection Mode**

Counter Mode	Configuration CLI	Function	Maximum
IPv4 only	counter ipv4	Only IPv4 statistics are collected.	4092
IPv6 only	counter ipv6	Only IPv6 statistics are collected.	4092
IPv4 and IPv6 combined	counter	Both IPv4 and IPv6 statistics are collected but are displayed only as a sum.	4092
IPv4 and IPv6 separate	counter ipv4 ipv6 separate	Both IPv4 and IPv6 statistics are collected and can be displayed individually.	2046

When mixing these configured modes, the rule is as follows:

(number of v4/v6/v4v6combined interfaces) + 2*(number of v4v6separate interfaces) <= 4092



Note

To enable Layer 3 interface counters, you need to enter the **counter** command in interface mode. For instructions, see the [“Configuring Layer 3 Interface Counters”](#) section on page 36-11.

The hardware counters are displayed in the output of the **show interface** command, as shown in the following example. Counter fields that are updated when the counter configuration is present are highlighted.

```
Switch# show interface gi3/1
GigabitEthernet3/1 is up, line protocol is up (connected)
  Hardware is Gigabit Ethernet Port, address is 001f.9e9e.f43f (bia 001f.9e9e.f43f)
  Internet address is 10.10.10.2/24
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 1000Mb/s, link type is auto, media type is 10/100/1000-TX
  input flow-control is on, output flow-control is on
  Auto-MDIX on (operational: on)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
```

```

Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 53000 bits/sec, 122 packets/sec
5 minute output rate 53000 bits/sec, 122 packets/sec
L3 in Switched: ucast: 37522 pkt, 752892 bytes - mcast: 0 pkt, 0 bytes <==== (A)
L3 out Switched: ucast: 37522 pkt, 752892 bytes - mcast: 0 pkt, 0 bytes <==== (B)
IPv6 L3 in Switched: ucast: 24328 pkt, 145968 bytes - mcast: 0 pkt, 0 bytes <== (C)
IPv6 L3 out Switched: ucast: 24328 pkt, 145968 bytes - mcast: 0 pkt, 0 bytes <== (D)
103639 packets input, 6632896 bytes, 0 no buffer
Received 0 broadcasts (0 IP multicasts)
0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
0 input packets with dribble condition detected
103674 packets output, 6641715 bytes, 0 underruns
0 output errors, 0 collisions, 1 interface resets
0 babbles, 0 late collision, 0 deferred
0 lost carrier, 0 no carrier
0 output buffer failures, 0 output buffers swapped out

```

The output of the previous configuration depends on the counter configuration (Table 36-2).

Table 36-2 Fields Updated in Previous Configuration/Counter Configuration

Counter Configuration	Updated Fields
IPv4 only	(A) and (B) only
IPv6 only	(C) and (D) only
IPv4 and IPv6 combined	(A) and (B) only
IPv4 and IPv6 separate	(A) and (B) for IPv4 (C) and (D) for IPv6

Configuration Guidelines

The Catalyst 4500 series switch supports AppleTalk routing and IPX routing. For AppleTalk routing and IPX routing information, refer to “Configuring AppleTalk” and “Configuring Novell IPX” in the *Cisco IOS AppleTalk* and *Novell IPX* configuration guides at the following URLs:

http://www.cisco.com/en/US/docs/ios/at/configuration/guide/12_4/atk_12_4_book.html

http://www.cisco.com/en/US/docs/ios/novipx/configuration/guide/config_novellipx_ps6350_TSD_Products_Configuration_Guide_Chapter.html



Note

Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, 7-E, 6L-E, and 6-E do not support AppleTalk and IPX routing.

- Catalyst 4500 series switches do not support subinterfaces or the **encapsulation** keyword on Layer 3 Fast Ethernet, Gigabit Ethernet, 10-Gigabit Ethernet interfaces.
- Starting IOS XE 3.11.0, Catalyst 4500 series switches do not support egress Access Controlled Lists (ACLs) on a tunnel interface and on the source interface of the tunnel.



Note

As with any Layer 3 interface running Cisco IOS software, the IP address and network assigned to an SVI cannot overlap those assigned to any other Layer 3 interface on the switch.

Configuring Logical Layer 3 GRE Tunnel Interfaces

Tunnels are point-to-point dedicated virtual links to transport packets from one endpoint to another. Generic Routing Encapsulation (GRE) is a tunneling protocol used to encapsulate network layer protocols inside virtual point-to-point links. A GRE tunnel only provides encapsulation and not encryption.

With GRE, devices running a given network layer protocol can communicate over a network running a different network layer protocol. A network receives and encapsulates the native packet into another network protocol and sends the encapsulated packet towards its de-encapsulation point. The encapsulation point is the tunnel entry and the de-encapsulation point is the tunnel exit.



Note

Beginning in Cisco IOS XE Release 3.7.1E, GRE tunnels are supported on the hardware on Cisco Catalyst 4500 Series switches.

When GRE is configured with tunnel options (such as key, checksum, etc.), packets are switched in software. When GRE is configured without tunnel options, packets are hardware-switched.

Restrictions and Limitations for Logical Layer 3 GRE Tunnel Interfaces:

- Multicast routing is not supported on GRE tunnels, so PIM configuration is not supported on a GRE tunnel interface.
- Limitation relating to GRE-encapsulated packets that are switched in hardware (applies only to Catalyst 4500-X Series Switches):

If a GRE tunnel is configured on a Catalyst 4500 switch and the ingress to this device is through a Layer 2 interface which has an SVI configured locally and is running HSRP (and is the current active), GRE encapsulated unicast traffic is not sent across the endpoints of the GRE tunnel, because of which routing adjacencies cannot be established and pings across the GRE tunnel do not work.

To work around this problem, configure the HSRP group to use the burned-in address (BIA) feature (**standby use-bia** interface configuration command). It enables HSRP groups to use an interface's burned-in MAC address (or physical MAC address) instead of a virtual MAC address. Note that configuring the **standby use-bia** interface configuration command may slow down convergence after an HSRP switchover, since the new active has to send a gratuitous ARP to refresh the ARP entries through the subnet. For more information, see:

<https://www.cisco.com/c/en/us/support/docs/ip/hot-standby-router-protocol-hsrp/9281-3.html>.

To configure a GRE tunnel, perform this task:

	Command	Purpose
Step 1	Switch(config)# interface tunnel <i>number</i>	Enables tunneling on the interface.
Step 2	Switch(config-if)# ipv6 address <i>ip_address</i> <i>subnet_mask</i>	Configures the IPv6 address and subnet mask.
Step 3	Switch(config-if)# ip address <i>ip_address</i> <i>subnet_mask</i>	Configures the IP address and IP subnet.
Step 4	Switch(config-if)# tunnel source { <i>ip-address</i> <i>type number</i> }	Configures the tunnel source.
Step 5	Switch(config-if)# tunnel destination { <i>hostname</i> <i>ip-address</i> }	Configures the tunnel destination.
Step 6	Switch(config-if)# tunnel mode gre ip	Configures the tunnel mode.
Step 7	Switch(config-if)# end	Exits configuration mode.

	Command	Purpose
Step 8	Switch# copy running-config startup-config	Saves your configuration changes to NVRAM.
Step 9	Switch# show running-config interface tunnel number	Verifies the configuration.

This example shows how to configure the logical Layer 3 GRE tunnel interface tunnel 2:

```
Switch> enable
Switch# config term
Switch(config)# interface tunnel 2
Switch(config-if)# ipv6 address 1001:1::1/64
Switch(config-if)# ip address 100.1.1.1 255.255.255.0
Switch(config-if)# tunnel source 10.10.10.1
Switch(config-if)# tunnel destination 10.10.10.2
Switch(config-if)# tunnel mode gre ip
Switch(config-if)# end
```

Configuring Logical Layer 3 VLAN Interfaces



Note

Before you can configure logical Layer 3 VLAN interfaces, you must create and configure the VLANs on the switch, assign VLAN membership to the Layer 2 interfaces, enable IP routing if IP routing is disabled, and specify an IP routing protocol.

To configure logical Layer 3 VLAN interfaces, perform this task:

	Command	Purpose
Step 1	Switch(config)# vlan vlan_ID	Creates the VLAN.
Step 2	Switch(config)# interface vlan vlan_ID	Selects an interface to configure.
Step 3	Switch(config-if)# ip address ip_address subnet_mask	Configures the IP address and IP subnet.
Step 4	Switch(config-if)# no shutdown	Enables the interface.
Step 5	Switch(config-if)# end	Exits configuration mode.
Step 6	Switch# copy running-config startup-config	Saves your configuration changes to NVRAM.
Step 7	Switch# show interfaces [type slot/interface] Switch# show ip interfaces [type slot/interface] Switch# show running-config interfaces [type slot/interface] Switch# show running-config interfaces vlan vlan_ID	Verifies the configuration.

This example shows how to configure the logical Layer 3 VLAN interface VLAN 2 and assign an IP address:

```
Switch> enable
Switch# config term
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# vlan 2
Switch(config)# interface vlan 2
Switch(config-if)# ip address 10.1.1.1 255.255.255.248
Switch(config-if)# no shutdown
Switch(config-if)# end
```

This example shows how to use the **show interfaces** command to display the interface IP address configuration and status of Layer 3 VLAN interface VLAN 2:

```
Switch# show interfaces vlan 2
Vlan2 is up, line protocol is down
  Hardware is Ethernet SVI, address is 00D.588F.B604 (bia 00D.588F.B604)
  Internet address is 172.20.52.106/29
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
    0 packets input, 0 bytes, 0 no buffer
    Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
    0 packets output, 0 bytes, 0 underruns
    0 output errors, 0 interface resets
    0 output buffer failures, 0 output buffers swapped out
Switch#
```

This example shows how to use the **show running-config** command to display the interface IP address configuration of Layer 3 VLAN interface VLAN 2:

```
Switch# show running-config
Building configuration...

Current configuration : !
interface Vlan2
  ip address 10.1.1.1 255.255.255.248
  !
  ip classless
  no ip http server
  !
  !line con 0
  line aux 0
  line vty 0 4
  !
end
```

Configuring VLANs as Layer 3 Interfaces

This section consists of the following subsections:

- [Configuring SVI Autostate Exclude, page 36-9](#)
- [Configuring IP MTU Sizes, page 36-10](#)
- [Configuring Layer 3 Interface Counters, page 36-11](#)

Configuring SVI Autostate Exclude



Note

The SVI Autostate Exclude feature is enabled by default and is synchronized with the STP state.

The SVI Autostate Exclude feature shuts down (or brings up) the Layer 3 interfaces of a switch when the following port configuration changes occur:

- When the last port on a VLAN goes down, the Layer 3 interface on that VLAN is shut down (SVI- autostated).
- When the first port on the VLAN is brought back up, the Layer 3 interface on the VLAN that was previously shut down is brought up.

SVI Autostate Exclude enables you to exclude the access ports and trunks in defining the status of the SVI (up or down) even if it belongs to the same VLAN. If the excluded access port and trunk is in up state and other ports are in down state in the VLAN, the SVI state is changed to down.

To make the SVI state up, at least one port in the VLAN should be up and not excluded. This action helps to exclude the monitoring port status when you are determining the status of the SVI.

To apply SVI Autostate Exclude, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface interface-id	Enters interface configuration mode.
Step 3	Switch(config-if)# switchport autostate exclude	Excludes the access ports and trunks in defining the status of an SVI (up or down).
Step 4	Switch(config)# end	Exits configuration mode.
Step 5	Switch# show run interface	Displays the running configuration.
Step 6	Switch# show interface switchport	Verifies the configuration.

This example shows how to apply SVI Autostate Exclude on interface g3/1:

```
Switch# conf terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface g3/1
Switch(config-if)# switchport autostate exclude
Switch(config-if)# end
Switch# show run int g3/4
Building configuration...

Current configuration : 162 bytes
!
interface GigabitEthernet3/4
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 2,3
 switchport autostate exclude
 switchport mode trunk
end
```

<=====

```

Switch# show int g3/4 switchport
Name: Gi3/4
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On Access Mode VLAN: 1 (default) Trunking Native Mode VLAN: 1
(default) Administrative Native VLAN tagging: enabled Voice VLAN: none Administrative
private-vlan host-association: none Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none Administrative private-vlan trunk
Native VLAN tagging: enabled Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none Administrative private-vlan trunk
associations: none Administrative private-vlan trunk mappings: none Operational
private-vlan: none Trunking VLANs Enabled: 2,3 Pruning VLANs Enabled: 2-1001 Capture Mode
Disabled Capture VLANs Allowed: ALL
Autostate mode exclude

Unknown unicast blocked: disabled
Unknown multicast blocked: disabled
Appliance trust: none
Switch#

```

Configuring IP MTU Sizes

You can set the protocol-specific maximum transmission unit (MTU) size of IPv4 or IPv6 packets that are sent on an interface.

For information on MTU limitations, refer to “Maximum Transmission Units” on page 41.



Note

To set the nonprotocol-specific MTU value for an interface, use the **mtu** interface configuration command. Changing the MTU value (with the **mtu** interface configuration command) can affect the IP MTU value. If the current IP MTU value matches the MTU value, and you change the MTU value, the IP MTU value is modified automatically to match the new MTU. However, the reverse is not true; changing the IP MTU value has no effect on the value for the **mtu** command.

For information on how to configure MTU size, refer to “Configuring MTU Sizes” on page 43.

To set the protocol-specific maximum transmission unit (MTU) size of IPv4 or IPv6 packets sent on an interface, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Enters interface configuration mode.
Step 3	Switch(config-if)# [no] ip mtu <i>mtu_size</i> or Switch(config-if)# [no] ipv6 mtu <i>mtu_size</i>	Configures the IPv4 MTU size Configures the IPv6 MTU size. The no form of the command reverts to the default MTU size (1500 bytes).
Step 4	Switch(config-if)# exit	Exits configuration interface mode.
Step 5	Switch(config)# end	Exits configuration mode.
Step 6	Switch# show run interface <i>interface-id</i>	Displays the running configuration.

This example shows how to configure IPv4 MTU on an interface:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface vlan 1
Switch(config-if)# ip mtu 68
Switch(config-if)# exit
Switch(config)# end
Switch# show ip interface vlan 1
Vlan1 is up, line protocol is up
  Internet address is 10.10.10.1/24
  Broadcast address is 255.255.255.255
  Address determined by setup command
  MTU is 68 bytes
  Helper address is not set
  .....(continued)
```

The following example shows how to configure IPv6 MTU on an interface:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface vlan 1
Switch(config-if)# ipv6 mtu 1280
Switch(config)# end
```

This example shows how to verify the configuration

```
Switch# show ipv6 interface vlan 1
Vlan1 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::214:6AFF:FEBC:DEEA
  Global unicast address(es):
    1001::1, subnet is 1001::/64
  Joined group address(es):
    FF02::1
    FF02::1:FF00:1
    FF02::1:FFBC:DEEA
  MTU is 1280 bytes
  .....(continued)
```



Note

When IPv6 is enabled on an interface using any CLI command, you may see the following message:

```
% Hardware MTU table exhausted
```

In this situation, the IPv6 MTU value programmed in hardware differs from the IPv6 interface MTU value. This situation occurs if no room exists in the hardware MTU table to store additional values. You must free up some space in the table by unconfiguring some unused MTU values and subsequently disable and reenabling IPv6 on the interface or reapplying the MTU configuration.

Configuring Layer 3 Interface Counters



Note

Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, 7-E, 6L-E, 6-E, do not support Layer 2 interface counters.

To configure Layer 3 interface counters (assign counters to a Layer 3 interface), perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Enters interface configuration mode.
Step 3	Switch(config-if)# counter { ipv4 ipv6 ipv4 ipv6 separate }	Enables counters. counter — Enables collection of IPv4 and IPv6 statistics and displays them as a sum counter ipv4 — Enables collection of IPv4 statistics only counter ipv6 — Enables collection of IPv6 statistics only counter ipv4 ipv6 separate — Enables collection of IPv4 and IPv6 statistics and displays them individually
Step 4	Switch(config)# end	Exits configuration mode.
Step 5	Switch# show run interface <i>interface-id</i>	Displays the running configuration.

This example shows how to enable counters on interface VLAN 1:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface vlan 1
Switch(config-if)# counter ipv4
Switch(config-if)# end
Switch#
00:17:15: %SYS-5-CONFIG_I: Configured from console by console
Switch# show run interface vlan 1
Building configuration...

Current configuration : 63 bytes
!
interface Vlan1
 ip address 10.0.0.1 255.0.0.0
 counter ipv4
end
```



Note

To remove the counters, use the **no counter** command.

If you have already assigned the maximum number of counters, the **counter** command fails and displays an error message:

```
Switch# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface fa3/2
Switch(config-if)# no switchport
Switch(config-if)# counter ipv6
Counter resource exhausted for interface fa3/2
Switch(config-if)# end
Switch#
00:24:18: %SYS-5-CONFIG_I: Configured from console by console
```

In this situation, you must release a counter from another interface for use by the new interface.

Configuring Physical Layer 3 Interfaces



Note

Before you can configure physical Layer 3 interfaces, you must enable IP routing if IP routing is disabled, and specify an IP routing protocol.

To configure physical Layer 3 interfaces, perform this task:

	Command	Purpose
Step 1	Switch(config)# ip routing	Enables IP routing (required only if disabled)
Step 2	Switch(config)# interface { fastethernet gigabitethernet tengigabitethernet } <i>slot/port</i> { port-channel <i>port_channel_number</i> }	Selects an interface to configure.
Step 3	Switch(config-if)# no switchport	Converts this port from physical Layer 2 port to physical Layer 3 port.
Step 4	Switch(config-if)# ip address <i>ip_address</i> <i>subnet_mask</i>	Configures the IP address and IP subnet.
Step 5	Switch(config-if)# no shutdown	Enables the interface.
Step 6	Switch(config-if)# end	Exits configuration mode.
Step 7	Switch# copy running-config startup-config	Saves your configuration changes to NVRAM.
Step 8	Switch# show interfaces [<i>type slot/interface</i>] Switch# show ip interfaces [<i>type slot/interface</i>] Switch# show running-config interfaces [<i>type slot/interface</i>]	Verifies the configuration.

This example shows how to configure an IP address on Fast Ethernet interface 2/1:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# ip routing
Switch(config)# interface fastethernet 2/1
Switch(config-if)# no switchport
Switch(config-if)# ip address 10.1.1.1 255.255.255.248
Switch(config-if)# no shutdown
Switch(config-if)# end
Switch#
```

This example shows how to use the **show running-config** command to display the interface IP address configuration of Fast Ethernet interface 2/1:

```
Switch# show running-config
Building configuration...
!
interface FastEthernet2/1
  no switchport
  ip address 10.1.1.1 255.255.255.248
!
...
ip classless
no ip http server
!
!
line con 0
line aux 0
```

```
line vty 0 4
!  
end
```

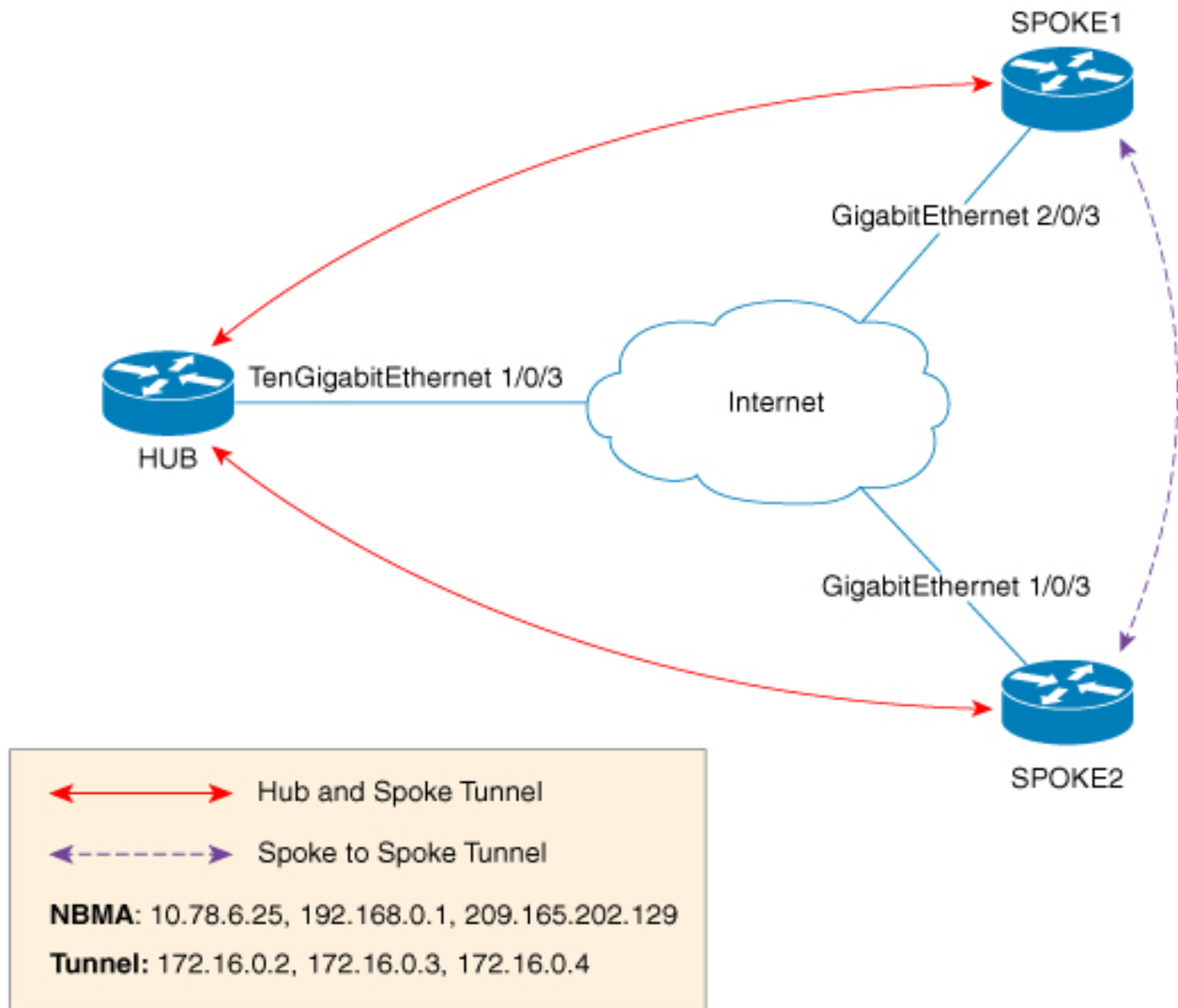
Configuring Multipoint GRE

This section consists of the following subsections:

- [About Multipoint GRE](#)
- [Configuring Unicast mGRE at Hub](#)
- [Configuring Unicast mGRE at Spoke](#)
- [Sample mGRE Configuration at Hub and Spokes](#)

About Multipoint GRE

Point-to-Multipoint (P2MP) is a hub-n-spoke topology that uses Multipoint GRE protocol (mGRE). mGRE is built over IPv4 core/underlying network and allows multiple destinations to be grouped into a single multipoint interface. It supports Open Shortest Path First (OSPF) and Enhanced Interior Gateway Routing Protocol (EIGRP) protocols, IPv4 and IPv6 unicast payload, and IPv4 multicast payload. mGRE does static and dynamic Next Hop Resolution Protocol (NHRP) tunneling for hub-to-spoke and spoke-to-spoke technologies, providing scalability and also reducing configuration complexity. Spokes dynamically register themselves with the hub and individual spokes also dynamically learn about other spokes using the NHRP protocol forming a dynamic-mesh network, that is, a non-broadcast multi-access network (NBMA). In NBMA, all routing protocols send their updates to a physical NBMA address. mGRE in conjunction with IPSEC and NHRP can be used in Dynamic Multipoint VPN (DMVPN).



In this figure, each spoke acts as a Next Hop Client (NHC) and is configured with static mapping information (hub's tunnel IP address and NBMA address) to reach hub which acts as Next Hop Server (NHS). NHCs send Next Hop Resolution Protocol (NHRP) registration request to NHS which allows NHS to learn mapping information of the spoke and form a tunnel (hub and spoke) dynamically.

In addition to NHRP registration of NHCs (spokes) with NHS (hub), NHRP provides the capability for NHC to dynamically discover another NHC on demand and form spoke-to-spoke tunnel. Without this discovery, IP packets traversing from hosts behind one spoke to hosts behind another spoke have to traverse by way of the NHS router. This increases the utilization of the hub's physical bandwidth and CPU to process these packets that come into the hub on the multipoint interface and go right back out the multipoint interface. This is often called hairpinning. With NHRP, systems attached to an NBMA network dynamically learn the NBMA address of the other systems that are part of that network,

allowing these systems to directly communicate without requiring traffic to use an intermediate hop. This alleviates the load on the intermediate hop (NHS) and can increase the overall bandwidth of the NBMA network to be greater than the bandwidth of the hub router and effectively creates a full-mesh-capable network without having to discover all possible connections beforehand.

This is called a dynamic-mesh network, where there is a base hub-and-spoke network of NHCs and NHSs for transporting NHRP, dynamic routing protocol information, data traffic, and dynamic direct spoke-to-spoke links that are built when there is data traffic to use the link and torn down when the data traffic stops.

Configuring Unicast mGRE at Hub

	Command	Purpose
Step 1	enable Device> enable	Enables privileged EXEC mode..
Step 2	configure terminal Device# configure terminal	Enters global configuration mode.
Step 3	interface type number Device(config)# interface tunnel 1	Configures an interface and enters interface configuration mode.
Step 4	ip nhrp map ip-address nbma-address Device(config-if)# ip nhrp map 10.0.0.1	Configures multipoint GRE as the tunnel mode.
Step 5	ip ospf network point-to-multipoint Device(config-if)# ip ospf network point-to-multipoint	If the underlying protocol is OSPF, execute this command to set the network type to point-to-multipoint.
Step 6	ip address address mask Device(config-if)# ip address 10.1.1.1 255.255.255.255	Configures IP address of the tunnel.
Step 7	ipv6 address address prefix Device(config-if)# ipv6 address 2001:DB8:1::1	Configures IPv6 address of the tunnel.
Step 8	tunnel source address Device(config-if)# tunnel source 172.16.1.3	Configures the source IP address of the tunnel.
Step 9	tunnel mode gre multipoint Device(config-if)# tunnel mode gre multipoint	Configures multipoint GRE as the tunnel mode.

	Command	Purpose
Step 10	{ip ipv6} nhrp network-id <i>id</i> Switch(config-if)# ip nhrp network id 1	Defines the NHRP domain which differentiates if multiple NHRP domains (GRE tunnel interfaces) are available on the same NHRP router.
Step 11	{ip ipv6} nhrp registration timeout <i>seconds</i> Device(config-if)# ip nhrp registration timeout 30	Changes the interval that NHRP NHCs take to send NHRP registration requests to configured NHRP NHSs..
Step 12	{ip ipv6} nhrp holdtime <i>seconds</i> Device(config-if)# ip nhrp holdtime 400#	Changes the number of seconds that NHRP NBMA addresses are advertised as valid in positive NHRP responses
Step 13	{ip ipv6} nhrp authentication <i>string</i> Device(config-if)# ip nhrp authentication DMVPN	Specifies an authentication string.
Step 14	ip pim nbma-mode Device(config-if)# ip pim nbma-mode	Configures a multiaccess WAN interface to be in non-broadcast multiaccess (NBMA) mode.
Step 15	ip nhrp map multicast dynamic Device(config-if)# ip nhrp map multicast dynamic	(Optional) Enables NHRP server (hub) to create a broadcast/multicast mapping for the spoke when spoke routers register their unicast NHRP mapping with the hub.
Step 16	ip next-hop self eigrp <i>number</i> Device(config-if)# ip next-hop self eigrp 10	(Optional) Enables the hub to use the next received hop while sending routing protocol updates of one spoke to another, so that hosts behind hosts can reach directly.
Step 17	ip split-horizon eigrp <i>number</i> Device(config-if)# ip split-horizon eigrp 10	(Optional) Enables routing protocol updates of one spoke to be sent to another spoke.
Step 18	end Device(config-if)# end	Exits interface configuration mode and returns to user EXEC mode.

Configuring Unicast mGRE at Spoke

	Command	Purpose
Step 1	enable Device> enable	Enables privileged EXEC mode..
Step 2	configure terminal Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Device(config)# interface tunnel1	Configures an interface and enters interface configuration mode. Ensure that this configuration is on a tunnel interface.
Step 4	ip nhrp map <i>ip-address nbma-address</i> Device(config-if)# ip nhrp map 10.0.0.1 192.0.0.1	Configures static IP-to-NBMA address mapping of hub on spoke.
Step 5	{ip ipv6} nhrp map multicast <i>nbma-address</i> Device(config-if)# ip nhrp map multicast 10.0.0.2	Enables IP multicast and broadcast packets (example: routing protocol information) to be sent from spoke to hub.
Step 6	ip nhrp nhs <i>nhs-address</i> Device(config-if)# ip nhrp nhs 192.0.2.1	Enables spoke to send NHRP registration request to hub. Here nhs-address is the hub tunnel' s address.
Step 7	tunnel source <i>address</i> Device(config-if)# tunnel source 172.16.1.3	Configures the source IP address of the tunnel.
Step 8	tunnel mode gre multipoint Device(config-if)# tunnel mode gre multipoint	Configures multipoint GRE as the tunnel mode.
Step 9	end Device(config-if)# end	Exits interface configuration mode and returns to user EXEC mode.

Sample mGRE Configuration at Hub and Spokes

On SPOKE 1:

```
interface Tunnel1
 ip address 192.168.1.3 255.255.255.0
 no ip redirects
 ip nhrp map 192.168.1.1 172.16.1.1
 ip nhrp network-id 1
 ip nhrp holdtime 600
```



```
ip nhrp nhs 192.168.1.1
ip nhrp registration timeout 30
tunnel source 172.16.1.3
tunnel mode gre multipoint
end
```

On SPOKE 2:

```
interface Tunnel1
 ip address 192.168.1.2 255.255.255.0
 no ip redirects
ip nhrp map 192.168.1.1 172.16.1.1
ip nhrp network-id 1
ip nhrp holdtime 600
ip nhrp nhs 192.168.1.1
ip nhrp registration timeout 30
tunnel source 172.16.1.2
tunnel mode gre multipoint
```

On HUB:

```
interface Tunnel1
 ip address 192.168.1.1 255.255.255.0
 no ip redirects
ip nhrp network-id 1
ip nhrp holdtime 600
ip nhrp registration timeout 30
ip ospf 1 area 0
tunnel source 172.16.1.1
tunnel mode gre multipoint
end
```




Configuring Cisco Express Forwarding

This chapter describes Cisco Express Forwarding (CEF) on the switch. It also provides guidelines, procedures, and examples to configure this feature.

This chapter includes the following major sections:

- [About CEF, page 37-1](#)
- [Catalyst 4500 Series Switch Implementation of CEF, page 37-3](#)
- [CEF Configuration Restrictions, page 37-6](#)
- [Configuring CEF, page 37-6](#)
- [Monitoring and Maintaining CEF, page 37-8](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About CEF

This section contains information on the two primary components that comprise the CEF operation:

- [CEF Features, page 37-1](#)
- [Forwarding Information Base, page 37-2](#)
- [Adjacency Tables, page 37-2](#)

CEF Features

CEF is advanced Layer 3 IP switching technology that optimizes performance and scalability for large networks with dynamic traffic patterns or networks with intensive web-based applications and interactive sessions.

CEF provides the following features:

- Improves performance over the caching schemes of multilayer switches, which often flush the entire cache when information changes in the routing tables.

- Provides load balancing that distributes packets across multiple links based on Layer 3 routing information. If a network device discovers multiple paths to a destination, the routing table is updated with multiple entries for that destination. Traffic to that destination is then distributed among the various paths.

CEF stores information in several data structures rather than the route cache of multilayer switches. The data structures optimize lookup for efficient packet forwarding.

Forwarding Information Base

The Forwarding Information Base (FIB) is a table that contains a copy of the forwarding information in the IP routing table. When routing or topology changes occur in the network, the route processor updates the IP routing table and CEF updates the FIB. Because there is a one-to-one correlation between FIB entries and routing table entries, the FIB contains all known routes and eliminates the need for route cache maintenance that is associated with switching paths, such as fast switching and optimum switching. CEF uses the FIB to make IP destination-based switching decisions and maintain next-hop address information based on the information in the IP routing table.

On the Catalyst 4500 series switches, CEF loads the FIB in to the Integrated Switching Engine hardware to increase the performance of forwarding. The Integrated Switching Engine has a finite number of forwarding slots for storing routing information. If this limit is exceeded, CEF is automatically disabled and all packets are forwarded in software. In this situation, you should reduce the number of routes on the switch and then reenable hardware switching with the **ip cef** command.

Adjacency Tables

In addition to the FIB, CEF uses adjacency tables to prepend Layer 2 addressing information. Nodes in the network are termed *adjacent* if they are within a single hop from each other. The adjacency table maintains Layer 2 next-hop addresses for all FIB entries.

Adjacency Discovery

The adjacency table is populated as new adjacent nodes are discovered. Each time an adjacency entry is created (such as using the Address Resolution Protocol (ARP), a link-layer header for that adjacent node is stored in the adjacency table. Once a route is determined, the link-layer header points to a next hop and corresponding adjacency entry. The link-layer header is subsequently used for encapsulation during CEF switching of packets.

Adjacency Resolution

A route might have several paths to a destination prefix, such as when a router is configured for simultaneous load balancing and redundancy. For each resolved path, a pointer is added for the adjacency corresponding to the next-hop interface for that path. This method is used for load balancing across several paths.

Adjacency Types That Require Special Handling

In addition to adjacencies for next-hop interfaces (host-route adjacencies), other types of adjacencies are used to expedite switching when certain exception conditions exist. When the prefix is defined, prefixes requiring exception processing are cached with one of the special adjacencies listed in [Table 37-1](#).

Table 37-1 Adjacency Types for Exception Processing

Adjacency Type	Processing Method
Null adjacency	Packets destined for a Null0 interface are dropped. A Null0 interface can be used as an effective form of access filtering.
Glean adjacency	When a router is connected directly to several hosts, the FIB table on the router maintains a prefix for the subnet rather than for each individual host. The subnet prefix points to a glean adjacency. When packets must be forwarded to a specific host, the adjacency database is gleaned for the specific prefix.
Punt adjacency	Features that require special handling or features that are not yet supported by CEF switching are sent (punted) to the next higher switching level.
Discard adjacency	Packets are discarded.
Drop adjacency	Packets are dropped.

Unresolved Adjacency

When a link-layer header is prepended to packets, FIB requires the prepend to point to an adjacency corresponding to the next hop. If an adjacency was created by FIB and was not discovered through a mechanism such as ARP, the Layer 2 addressing information is not known and the adjacency is considered incomplete. When the Layer 2 information is known, the packet is forwarded to the route processor, and the adjacency is determined through ARP.

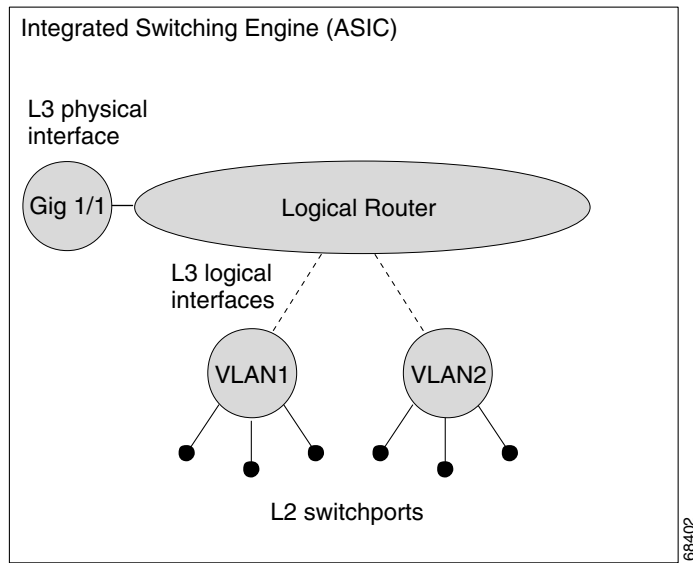
Catalyst 4500 Series Switch Implementation of CEF

Catalyst 4500 series switch support an ASIC-based Integrated Switching Engine that provides these features:

- Ethernet bridging at Layer 2
- IP routing at Layer 3

Because the ASIC is specifically designed to forward packets, the Integrated Switching Engine hardware can run this process much faster than CPU subsystem software.

[Figure 37-1](#) shows a high-level view of the ASIC-based Layer 2 and Layer 3 switching process on the Integrated Switching Engine.

Figure 37-1 Logical L2/L3 Switch Components

The Integrated Switching Engine performs inter-VLAN routing on logical Layer 3 interfaces with the ASIC hardware. The ASIC hardware also supports a physical Layer 3 interface that can be configured to connect with a host, a switch, or a router.

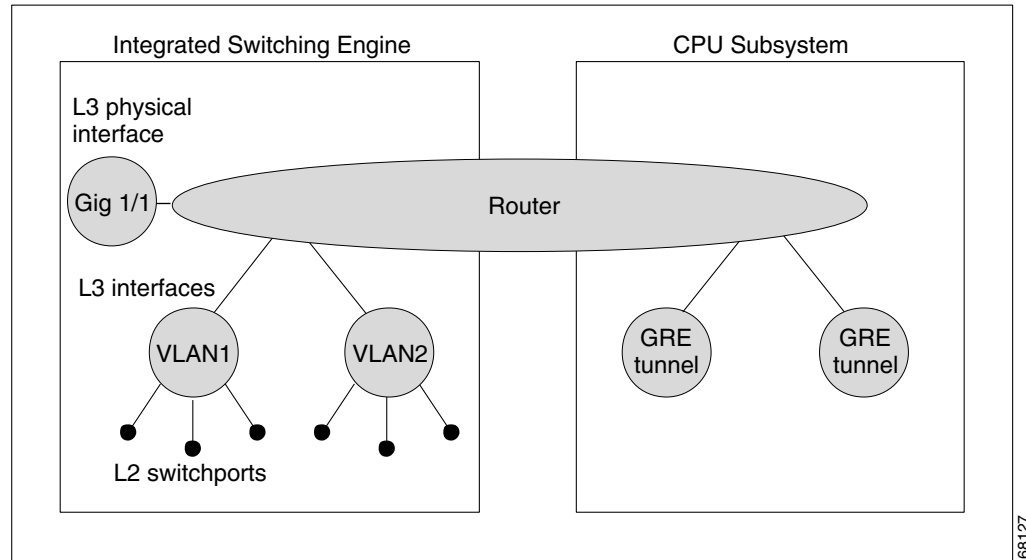
This section contains the following subsections:

- [Hardware and Software Switching, page 37-4](#)
- [Load Balancing, page 37-6](#)
- [Software Interfaces, page 37-6](#)

Hardware and Software Switching

For the majority of packets, the Integrated Switching Engine performs the packet forwarding function in hardware. These packets are hardware-switched at very high rates. Exception packets are forwarded by the CPU subsystem software. Statistic reports should show that the Integrated Switching Engine is forwarding the vast majority of packets in hardware. Software forwarding is significantly slower than hardware forwarding, but packets forwarded by the CPU subsystem do not reduce hardware forwarding speed.

[Figure 37-2](#) shows a logical view of the Integrated Switching Engine and the CPU subsystem switching components.

Figure 37-2 Hardware and Software Switching Components

The Integrated Switching Engine performs inter-VLAN routing in hardware. The CPU subsystem software supports Layer 3 interfaces to VLANs that use Subnetwork Access Protocol (SNAP) encapsulation. The CPU subsystem software also supports generic routing encapsulation (GRE) tunnels.

Hardware Switching

- Hardware switching is the normal operation for switches with supervisor engines.
- Beginning in Cisco IOS XE Release 3.7.1E, GRE tunnels are supported on the hardware on Catalyst 4500 series switches. When GRE is configured without tunnel options, packets are hardware-switched. Otherwise, packets are switched in the software.

Software Switching

Software switching occurs when traffic cannot be processed in hardware. The following types of exception packets are processed in software at a much slower rate:

- Packets that use IP header options



Note Packets that use TCP header options are switched in hardware because they do not affect the forwarding decision.

- Packets that have an expiring IP time-to-live (TTL) counter
- Packets that are forwarded to a tunnel interface.



Note When GRE tunnels are configured without tunnel options, packets are hardware-switched.

- Packets that arrive with non-supported encapsulation types
- Packets that are routed to an interface with non-supported encapsulation types

- Packets that exceed the MTU of an output interface and must be fragmented
- Packets that require an IGMP redirect for routing
- 802.3 Ethernet packets

Load Balancing

The Catalyst 4500 series switch supports load balancing for routing packets in the Integrated Switching Engine hardware. Load balancing is always enabled. It works when multiple routes for the same network with different next-hop addresses are configured. These routes can be configured either statically or through a routing protocol such as OSPF or EIGRP.

The hardware makes a forwarding decision by using a hardware load sharing hash function to compute a value, based on the source and destination IP addresses and the source and destination TCP port numbers (if available). This load sharing hash value is then used to select which route to use to forward the packet. All hardware switching within a particular flow (such as a TCP connection) is routed to the same next hop, which reduces the chance that packet reordering occurs. Up to eight different routes for a particular network are supported.

Software Interfaces

Cisco IOS for the Catalyst 4500 series switch supports GRE and IP tunnel interfaces that are not part of the hardware forwarding engine. All packets that flow to or from these interfaces must be processed in software and have a significantly lower forwarding rate than that of hardware-switched interfaces. Also, Layer 2 features are not supported on these interfaces.

CEF Configuration Restrictions

The CEF Integrated Switching Engine supports only ARPA and ISL/802.1q encapsulation types for Layer 3 switching in hardware. The CPU subsystem supports a number of encapsulations such as SNAP for Layer 2 switching that you can use for Layer 3 switching in software.

Configuring CEF

These sections describe how to configure CEF:

- [Enabling CEF, page 37-6](#)
- [Configuring Load Balancing for CEF, page 37-7](#)

Enabling CEF

By default, CEF is enabled globally on the Catalyst 4500 series switch. No configuration is required.

To reenabling CEF, perform this task:

Command	Purpose
Switch(config)# ip cef distributed	Enables standard CEF operation.

Configuring Load Balancing for CEF

CEF load balancing is based on a combination of source and destination packet information; it allows you to optimize resources by distributing traffic over multiple paths for transferring data to a destination. You can configure load balancing on a per-destination basis. Load-balancing decisions are made on the outbound interface. You can configure per-destination load balancing for CEF on outbound interfaces.

The following topics are discussed:

- [Configuring Per-Destination Load Balancing, page 37-7](#)
- [Configuring Load Sharing Hash Function, page 37-7](#)
- [Viewing CEF Information, page 37-8](#)

Configuring Per-Destination Load Balancing

Per-destination load balancing is enabled by default when you enable CEF. To use per-destination load balancing, you do not perform any additional tasks once you enable CEF.

Per-destination load balancing allows the router to use multiple paths to achieve load sharing. Packets for a given source-destination host pair are guaranteed to take the same path, even if multiple paths are available. Traffic destined for different pairs tend to take different paths. Per-destination load balancing is enabled by default when you enable CEF; it is the load balancing method of choice in most situations.

Because per-destination load balancing depends on the statistical distribution of traffic, load sharing becomes more effective as the number of source-destination pairs increases.

Use per-destination load balancing to ensure that packets for a given host pair arrive in order. All packets for a certain host pair are routed over the same link or links.

Configuring Load Sharing Hash Function

When multiple unicast routes exist to a particular destination IP prefix, the hardware sends packets matching that prefix across all possible routes, which shares the load across all next hop routers. By default, the route used is chosen by computing a hash of the source and destination IP addresses and using the resulting value to select the route. This preserves packet ordering for packets within a flow by ensuring that all packets within a single IP source/destination flow are sent on the same route, but it provides a near-random distribution of flows to routes.

You can change the load-sharing hash function. So, in addition to the source and destination IP addresses, the source TCP/UDP port, the destination TCP/UDP port, or both can also be included in the hash.

To the configure load sharing hash function to use the source and/or destination ports, perform this task:

Command	Purpose
Switch (config)# [no] ip cef load-sharing algorithm include-ports source destination	Enables load sharing hash function to use source and destination ports. Use the no keyword to set the switch to use the default Cisco IOS load-sharing algorithm.

**Note**

The **include-ports** option does not apply to software-switched traffic on the Catalyst 4500 series switches.

Viewing CEF Information

You can view the collected CEF information. To view CEF information, perform this task:

Command	Purpose
Switch# show ip cef	Displays the collected CEF information.

Monitoring and Maintaining CEF

To display information about IP traffic, perform this task:

Command	Purpose
Switch# show interface type slot/interface begin L3	Displays a summary of IP unicast traffic.

This example shows how to display information about IP unicast traffic on interface Fast Ethernet 3/3:

```
Switch# show interface fastethernet 3/3 | begin L3
L3 in Switched: ucast: 0 pkt, 0 bytes - mcast: 12 pkt, 778 bytes mcast
L3 out Switched: ucast: 0 pkt, 0 bytes - mcast: 0 pkt, 0 bytes
4046399 packets input, 349370039 bytes, 0 no buffer
Received 3795255 broadcasts, 2 runs, 0 giants, 0 throttles
<...output truncated...>
Switch#
```

**Note**

The IP unicast packet count is updated approximately every five seconds.

Displaying IP Statistics

IP unicast statistics are gathered on a per-interface basis. To display IP statistics, perform this task:

Command	Purpose
Switch# show interface <i>type number</i> counters detail	Displays IP statistics.

This example shows how to display IP unicast statistics for fastethernet 3/1:

Switch# **show interface fastethernet 3/1 counters detail**

```

Port                InBytes          InUcastPkts      InMcastPkts      InBcastPkts
Fa3/1                7263539133      5998222          6412307          156

Port                OutBytes          OutUcastPkts      OutMcastPkts      OutBcastPkts
Fa3/1                7560137031      5079852          12140475          38

Port                InPkts 64        OutPkts 64        InPkts 65-127    OutPkts 65-127
Fa3/1                11274          168536          7650482          12395769

Port                InPkts 128-255   OutPkts 128-255   InPkts 256-511   OutPkts 256-511
Fa3/1                31191          55269           26923            65017

Port                InPkts 512-1023  OutPkts 512-1023
Fa3/1                133807          151582

Port                InPkts 1024-1518 OutPkts 1024-1518 InPkts 1519-1548 OutPkts 1519-1548
Fa3/1                N/A             N/A             N/A             N/A

Port                InPkts 1024-1522 OutPkts 1024-1522 InPkts 1523-1548 OutPkts 1523-1548
Fa3/1                4557008          4384192          0                0

Port                Tx-Bytes-Queue-1  Tx-Bytes-Queue-2  Tx-Bytes-Queue-3  Tx-Bytes-Queue-4
Fa3/1                64                0                 91007             7666686162

Port                Tx-Drops-Queue-1  Tx-Drops-Queue-2  Tx-Drops-Queue-3  Tx-Drops-Queue-4
Fa3/1                0                 0                 0                 0

Port                Rx-No-Pkt-Buff    RxPauseFrames      TxPauseFrames      PauseFramesDrop
Fa3/1                0                 0                  0                  N/A

Port                UnsupOpcodePause
Fa3/1                0
Switch#

```

To display CEF (software switched) and hardware IP unicast adjacency table information, perform this task:

Command	Purpose
Switch# show adjacency [<i>interface</i>] [detail internal summary]	Displays detailed adjacency information, including Layer 2 information, when the optional detail keyword is used.

This example shows how to display adjacency statistics:

Switch# **show adjacency gigabitethernet 3/5 detail**

```

Protocol Interface          Address
IP          GigabitEthernet9/5  172.20.53.206(11)
                                     504 packets, 6110 bytes
                                     00605C865B82
                                     000164F83FA50800
ARP          03:49:31

```



Note

Adjacency statistics are updated approximately every 10 seconds.



Configuring Unicast Reverse Path Forwarding

This chapter describes the Unicast Reverse Path Forwarding (Unicast RPF) feature. The Unicast RPF feature helps to mitigate problems that are caused by malformed or forged IP source addresses that are passing through a switch.

For a complete description of the Unicast RPF commands in this chapter, refer to the chapter “Unicast Reverse Path Forwarding Commands” of the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the “Identifying Supported Platforms” section in the chapter “Using Cisco IOS Software.”

This chapter includes the following sections:

- [About Unicast Reverse Path Forwarding, page 38-1](#)
- [Unicast RPF Configuration Tasks, page 38-9](#)
- [Monitoring and Maintaining Unicast RPF, page 38-11](#)
- [Unicast RPF Configuration Example: Inbound and Outbound Filters, page 38-12](#)
- [Unicast RPF with ACL Support Configuration Example:, page 38-13](#)
- [Feature Information for Unicast Reverse Path Forwarding, page 38-13](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About Unicast Reverse Path Forwarding

The Unicast RPF feature helps to mitigate problems that are caused by the introduction of malformed or forged (spoofed) IP source addresses into a network by discarding IP packets that lack a verifiable IP source address. For example, a number of common types of denial-of-service (DoS) attacks, including Smurf and Tribal Flood Network (TFN), can take advantage of forged or rapidly changing source IP addresses to allow attackers to thwart efforts to locate or filter the attacks. For Internet service providers

(ISPs) that provide public access, Unicast RPF deflects such attacks by forwarding only packets that have source addresses that are valid and consistent with the IP routing table. This action protects the network of the ISP, its customer, and the rest of the Internet.

This section covers the following information:

- [How Unicast RPF Works, page 38-2](#)
- [Unicast Reverse Path Forwarding with ACL Support, page 38-4](#)
- [Unicast Reverse Path Forwarding with ACL Support, page 38-4](#)
- [Restrictions, page 38-8](#)
- [Related Features and Technologies, page 38-8](#)
- [Prerequisites to Configuring Unicast RPF, page 38-9](#)

How Unicast RPF Works

When Unicast RPF is enabled on an interface, the switch examines all packets received as input on that interface to make sure that the source address and source interface appear in the routing table and match the interface on which the packet was received. This ability to look backwards is available only when Cisco Express Forwarding (CEF) is enabled on the switch, because the lookup relies on the presence of the Forwarding Information Base (FIB). CEF generates the FIB as part of its operation.



Note

Unicast RPF is an input function and is applied only on the input interface of a switch at the upstream end of a connection.

Unicast RPF checks to see if any packet received at a switch interface arrives on the best return path (return route) to the source of the packet. Unicast RPF does this by doing a reverse lookup in the CEF table. If the packet was received from one of the best reverse path routes, the packet is forwarded as normal. If there is no reverse path route on the same interface from which the packet was received, it might mean that the source address was modified. If Unicast RPF does not find a reverse path for the packet, the packet is dropped.



Note

With Unicast RPF, all equal-cost “best” return paths are considered valid. This means that Unicast RPF works in cases where multiple return paths exist, provided that each path is equal to the others in terms of the routing cost (number of hops, weights, and so on) and as long as the route is in the FIB. Unicast RPF also functions where EIGRP variants are being used and unequal candidate paths back to the source IP address exist.

When a packet is received at the interface where Unicast RPF and ACLs have been configured, the following actions occur:

- Step 1** Input ACLs configured on the inbound interface are checked.
- Step 2** Unicast RPF checks to see if the packet has arrived on the best return path to the source, which it does by doing a reverse lookup in the FIB table.
- Step 3** CEF table (FIB) lookup is carried out for packet forwarding.
- Step 4** Output ACLs are checked on the outbound interface.

Step 5 The packet is forwarded.

This section provides information about Unicast RPF enhancements:

- Access control lists and logging
- Per-interface statistics

Figure 38-1 illustrates how Unicast RPF and CEF work together to validate IP source addresses by verifying packet return paths. In this example, a customer has sent a packet having a source address of 192.168.1.1 from interface Gigabit Ethernet 1/1. Unicast RPF checks the FIB to see if 192.168.1.1 has a path to Gigabit Ethernet 1/1. If there is a matching path, the packet is forwarded. If there is no matching path, the packet is dropped.

Figure 38-1 Unicast RPF Validating IP Source Addresses

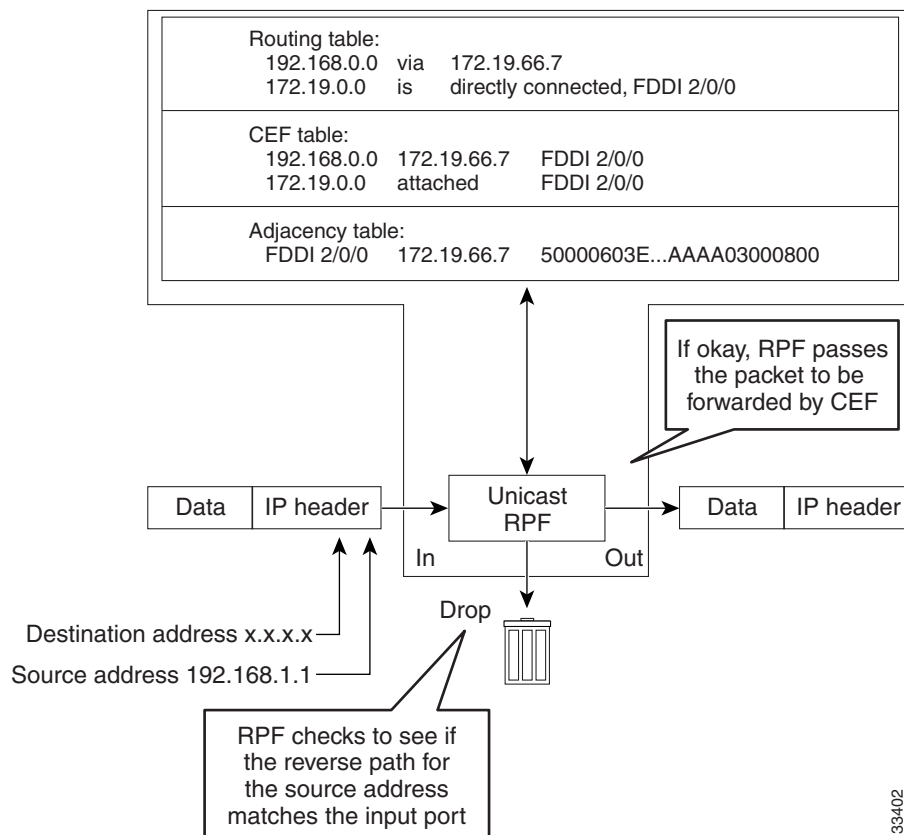
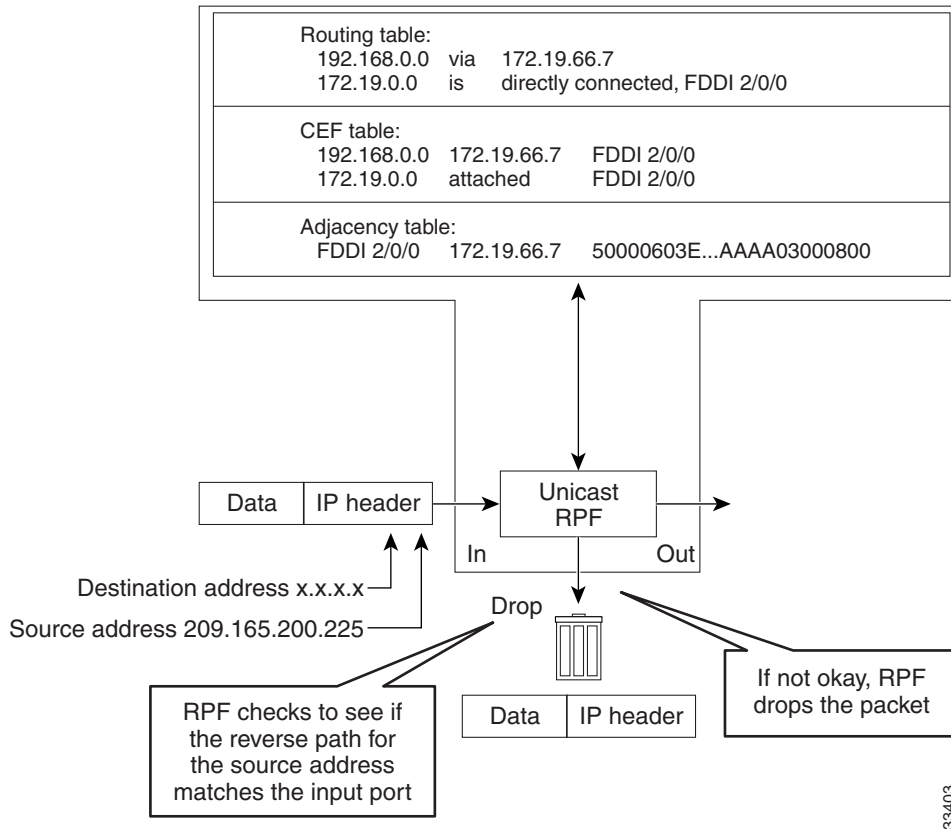


Figure 38-2 illustrates how Unicast RPF drops packets that fail validation. In this example, a customer has sent a packet having a source address of 209.165.200.225, which is received at interface Gigabit Ethernet 1/1. Unicast RPF checks the FIB to see if 209.165.200.225 has a return path to Gigabit Ethernet 1/1. If there is a matching path, the packet is forwarded. There is no reverse entry in the routing table that routes the customer packet back to source address 209.165.200.225 on interface Gigabit Ethernet 1/1, and so the packet is dropped.

Figure 38-2 Unicast RPF Dropping Packets That Fail Verification

Unicast Reverse Path Forwarding with ACL Support

The Unicast Reverse Path Forwarding ACL Support feature adds the access control list (ACL) support to the Unicast Reverse Path Forwarding feature. With the ACL support, Unicast Reverse Path Forwarding (RPF) can determine whether to drop or to forward data packets that have malformed or forged IP source addresses. The ACL support for Unicast Reverse Path Forwarding feature was introduced for IPv4 family on Catalyst 4500-E series switches and Catalyst 4500-X series switches starting with the Cisco IOS Release 15.2(6)E2.

Implementing Unicast RPF

Unicast RPF has several key implementation principles:

- The packet must be received at an interface that has the best return path (route) to the packet source (a process called *symmetric routing*). There must be a route in the FIB matching the route to the receiving interface. Adding a route in the FIB is done with a static route, network statement, or dynamic routing. (ACLs permit the use of Unicast RPF when packets will arrive by specific, less optimal asymmetric input paths.)
- IP source addresses at the receiving interface must match the routing entry for the interface.
- Unicast RPF is an input function and is applied only on the input interface of a switch at the upstream end of a connection.

Given these implementation principles, Unicast RPF becomes a tool that network administrators can use not only for their customers but also for their downstream network or ISP, even if the downstream network or ISP has other connections to the Internet.

**Caution**

Using optional BGP attributes such as weight and local preference, you can modify the best path back to the source address. Modification affects the operation of Unicast RPF.

This section provides information about the implementation of Unicast RPF:

- [Security Policy and Unicast RPF, page 38-5](#)
- [Where to Use Unicast RPF, page 38-5](#)
- [Routing Table Requirements, page 38-7](#)
- [Where Not to Use Unicast RPF, page 38-7](#)
- [Unicast RPF with BOOTP and DHCP, page 38-8](#)

Security Policy and Unicast RPF

Consider the following points in determining your policy for deploying Unicast RPF:

- Unicast RPF must be applied at the interface downstream from the larger portion of the network, preferably at the edges of your network.
- The farther downstream you apply Unicast RPF, the finer the granularity you have in mitigating address spoofing and in identifying the sources of spoofed addresses. For example, applying Unicast RPF on an aggregation switch helps mitigate attacks from many downstream networks or clients and is simple to administer, but it does not help identify the source of the attack. Applying Unicast RPF at the network access server helps limit the scope of the attack and trace the source of the attack; however, deploying Unicast RPF across many sites does add to the administration cost of operating the network.
- The more entities that deploy Unicast RPF across Internet, intranet, and extranet resources, the better the chances of mitigating large-scale network disruptions throughout the Internet community, and the better the chances of tracing the source of an attack.
- Unicast RPF will not inspect IP packets encapsulated in tunnels, such as GRE, LT2P, or PPTP. Unicast RPF must be configured at a home gateway so that Unicast RPF processes network traffic only after the tunneling and encryption layers have been stripped off the packets.

Where to Use Unicast RPF

Unicast RPF can be used in any single-homed environment where there is essentially only one access point out of the network; that is, one upstream connection. Networks having one access point offer the best example of symmetric routing, which means that the interface where a packet enters the network is also the best return path to the source of the IP packet. Unicast RPF is best used at the network perimeter for Internet, intranet, or extranet environments, or in ISP environments for customer network terminations.

Enterprise Networks with a Single Connection to an ISP

In enterprise networks, one objective of using Unicast RPF for filtering traffic at the input interface (a process called *ingress filtering*) is for protection from malformed packets arriving from the Internet. Traditionally, local networks with one connection to the Internet use ACLs at the receiving interface to prevent spoofed packets from the Internet from entering their local network.

ACLs work well for many single-homed customers; however, there are trade-offs when ACLs are used as ingress filters, including two commonly referenced limitations:

- Packet per second (PPS) performance at very high packet rates



Note This restriction applies only to software packet forwarding. Hardware packet forwarding is the same on both ACL and uRPF.

- Maintenance of the ACL (whenever new addresses are added to the network)

Unicast RPF is one tool that addresses both of these limitations. With Unicast RPF, ingress filtering is done at CEF PPS rates. This processing speed makes a difference when the link is more than 1 Mbps. Additionally, since Unicast RPF uses the FIB, no ACL maintenance is necessary, and thus the administration overhead of traditional ACLs is reduced. The following figure and example demonstrate how Unicast RPF is configured for ingress filtering.

Figure 38-3 illustrates an enterprise network that has a single link to an upstream ISP. In this example, Unicast RPF is applied at interface Gigabit Ethernet 1/1 on the Enterprise switch for protection from malformed packets arriving from the Internet. Unicast RPF is also applied at interface Gigabit Ethernet 2/1 on the ISP switch for protection from malformed packets arriving from the enterprise network.

Figure 38-3 Enterprise Network Using Unicast RPF for Ingress Filtering



Using the topography in Figure 38-3, a typical configuration (assuming that CEF is turned on) on the ISP switch appears as follows:

```
interface Gigabit Ethernet 2/1
  description Link to Enterprise Network
  ip address 192.168.3.1 255.255.255.255
  no switchport
  ip address 10.1.1.2 255.255.255.0
  ip verify unicast source reachable-via rx allow-default
```

The gateway switch configuration of the enterprise network (assuming that CEF is turned on) appears as follows:

```
interface Gigabit Ethernet 1/2
  description ExampleCorp LAN
  ip address 192.168.10.1 255.255.252.0
  no ip redirects
  no ip directed-broadcast
  no ip proxy-arp

interface Gigabit Ethernet 1/1
```

```
description Link to Internet
no switchport
ip address 10.1.1.1 255.255.255.0
ip verify unicast source reachable-via rx allow-default
no ip proxy-arp
no ip redirects
no ip directed-broadcast
```

Unicast RPF works with a single default route. No additional routes or routing protocols exist. Network 192.168.10.0/22 is a connected network. Packets arriving from the Internet with a source address in the range 192.168.10.0/22 are dropped by Unicast RPF.

Routing Table Requirements

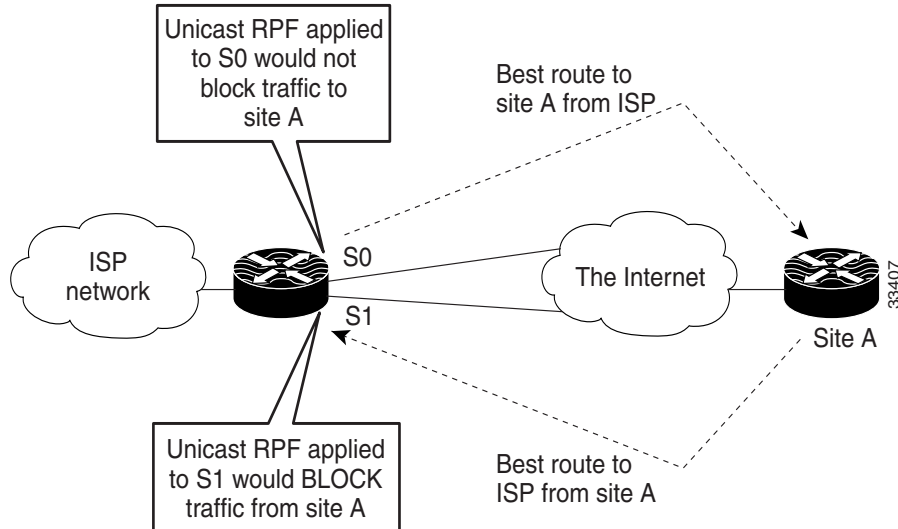
To work correctly, Unicast RPF needs proper information in the CEF tables. This requirement does not mean that the switch must have the entire Internet routing table. The amount of routing information needed in the CEF tables depends on where Unicast RPF is configured and what functions the switch performs in the network. For example, in an ISP environment, a switch that is a leased-line aggregation switch for customers needs only the information based on the static routes redistributed into the IGP or IBGP (depending on which technique is used in the network). Unicast RPF is configured on the customer interfaces, creating the requirement for minimal routing information. In another scenario, a single-homed ISP could place Unicast RPF on the gateway link to the Internet. The full Internet routing table is required. Requiring the full routing table helps protect the ISP from external DoS attacks that use addresses that are not in the Internet routing table.

Where Not to Use Unicast RPF

Do not use Unicast RPF on interfaces that are internal to the network. Internal interfaces are likely to have routing asymmetry (see [Figure 38-4](#)), meaning multiple routes to the source of a packet. Apply Unicast RPF only where there is natural or configured symmetry. Provided administrators carefully plan which interfaces they activate Unicast RPF on, routing asymmetry is not a serious problem.

For example, switches at the edge of the network of an ISP are more likely to have symmetrical reverse paths than switches that are in the core of the ISP network. Switches that are in the core of the ISP network have no guarantee that the best forwarding path out of the switch is the path selected for packets returning to the switch. We do not recommend that you apply Unicast RPF where there is a chance of asymmetric routing, unless you use ACLs to allow the switch to accept incoming packets. ACLs permit the use of Unicast RPF when packets will arrive by specific, less optimal asymmetric input paths. However, it is simplest to place Unicast RPF only at the edge of a network or, for an ISP, at the customer edge of the network.

[Figure 38-4](#) illustrates how Unicast RPF can block legitimate traffic in an asymmetrical routing environment.

Figure 38-4 Unicast RPF Blocking Traffic in an Asymmetrical Routing Environment

Unicast RPF with BOOTP and DHCP

Unicast RPF will allow packets with 0.0.0.0 source and 255.255.255.255 destination to pass so that Bootstrap Protocol (BOOTP) and Dynamic Host Configuration Protocol (DHCP) functions work properly.

Restrictions

Restrictions for applying Unicast RPF to multihomed clients include the following:

- Clients should not be multihomed to the same switch because action defeats the purpose of building a redundant service for the client.
- Customers must ensure that the packets flowing up the link (out to the Internet) match the route advertised out the link. Otherwise, Unicast RPF filters those packets as malformed packets.

Limitation

Unicast loose mode is not supported.

Related Features and Technologies

For more information about Unicast RPF-related features and technologies, review the following:

- Unicast RPF requires Cisco express forwarding (CEF) to function properly on the switch. For more information about CEF, refer to the *Cisco IOS Switching Services Configuration Guide*.
- Unicast RPF can be more effective at mitigating spoofing attacks when combined with a policy of *ingress* and *egress* filtering using Cisco IOS access control lists (ACLs).

- Ingress filtering applies filters to traffic received at a network interface from either internal or external networks. With ingress filtering, packets that arrive from other networks or the Internet and that have a source address that matches a local network, private, or broadcast address are dropped. In ISP environments, for example, ingress filtering can apply to traffic received at the switch from either the client (customer) or the Internet.
- Egress filtering applies filters to traffic exiting a network interface (the sending interface). By filtering packets on switches that connect your network to the Internet or to other networks, you can permit only packets with valid source IP addresses to leave your network.

For more information on network filtering, refer to RFC 2267 and to the *Cisco IOS IP Configuration Guide*.

Prerequisites to Configuring Unicast RPF

Prior to configuring Unicast RPF, configure ACLs:

- Configure standard or extended ACLs to mitigate transmission of invalid IP addresses (perform egress filtering). Permit only valid source addresses to leave your network and get onto the Internet. Prevent all other source addresses from leaving your network for the Internet.
- Configure standard or extended ACLs entries to drop (deny) packets that have invalid source IP addresses (perform ingress filtering). Invalid source IP addresses include the following types:
 - Reserved addresses
 - Loopback addresses
 - Private addresses (RFC 1918, *Address Allocation for Private Internets*)
 - Broadcast addresses (including multicast addresses)
 - Source addresses that fall outside the range of valid addresses associated with the protected network

Unicast RPF Configuration Tasks

The following sections describe the configuration tasks for Unicast RPF. Each task in the list is identified as either optional or required.

- [Configuring Unicast RPF, page 38-9](#) (Required)
- [Configuring Unicast RPF with ACL Support, page 38-10](#)
- [Verifying Unicast RPF, page 38-11](#) (Optional)

See the section “[Unicast RPF Configuration Example: Inbound and Outbound Filters](#)” at the end of this chapter.

Configuring Unicast RPF

Unicast RPF is an input-side function that is enabled on an interface operates on IP packets received by the switch.

To configure Unicast RPF, perform the following task:

	Command	Purpose
Step 1	Switch(config-if)# interface <i>type</i>	<p>Selects the input interface on which you want to apply Unicast RPF. It is the receiving interface, allowing Unicast RPF to verify the best return path before forwarding the packet on to the next destination.</p> <p>The interface type is specific to your switch and the types of interface cards installed on the switch. To display a list of available interface types, enter the interface ? command.</p>
Step 2	Switch(config-if)# ip verify unicast source reachable-via rx allow-default	Enables Unicast RPF on the interface.
Step 3	Switch(config-if)# exit	Exits interface configuration mode. Repeat Steps 2 and 3 for each interface on which you want to apply Unicast RPF.

Configuring Unicast RPF with ACL Support

To configure Unicast RPF with ACL Support, perform the following task:

	Command	Purpose
Step 1	Switch(config-if)# interface <i>type</i>	<p>Selects the input interface on which you want to apply Unicast RPF. It is the receiving interface, allowing Unicast RPF to verify the best return path before forwarding the packet on to the next destination.</p> <p>The interface type is specific to your switch and the types of interface cards installed on the switch. To display a list of available interface types, enter the interface ? command.</p>
Step 2	Switch(config-if)# ip address ipv4-address/prefix-length	Configures an IPv4 address and enables IPv4 processing on an interface.
Step 3	Switch(config-if)# ip verify unicast source reachable-via rx allow-default access-list	Enables Unicast RPF on the interface.
Step 4	Switch(config-if)# exit	Exits interface configuration mode. Repeat Steps 2 and 3 for each interface on which you want to apply Unicast RPF.

Verifying Unicast RPF

To verify that Unicast RPF is operational, use the **show cef interface** command. The following example shows that Unicast RPF is enabled at interface Gigabit Ethernet 3/1:

```
Switch# show cef interface gigabitEthernet 3/1
GigabitEthernet3/1 is up (if_number 79)
  Corresponding hwidb fast_if_number 79
  Corresponding hwidb firstsw->if_number 79
  Internet address is 10.1.1.1/24
  ICMP redirects are always sent
  IP unicast RPF check is enabled <=====
  Input features: uRPF <=====
  Inbound access list is not set
  Outbound access list is not set
  IP policy routing is disabled
  BGP based policy accounting on input is disabled
  BGP based policy accounting on output is disabled
  Hardware idb is GigabitEthernet3/1
  Fast switching type 1, interface type 155
  IP CEF switching enabled
  IP CEF switching turbo vector
  IP Null turbo vector
  IP prefix lookup IPv4 mtrie 8-8-8-8 optimized
  Input fast flags 0x4000, Output fast flags 0x0
  ifindex 78(78)
  Slot 3 Slot unit 1 VC -1
  Transmit limit accumulator 0x0 (0x0)
  IP MTU 1500
```

Monitoring and Maintaining Unicast RPF

To monitor and maintain Unicast RPF, perform this task:

Command	Purpose
Switch# show ip traffic	Displays global switch statistics about Unicast RPF drops and suppressed drops.
Switch(config-if)# no ip verify unicast	Disables Unicast RPF at the interface.

Unicast RPF counts the number of packets dropped or suppressed because of malformed or forged source addresses. Unicast RPF counts dropped or forwarded packets that include the following global and per-interface information:

- Global Unicast RPF drops
- Per-interface Unicast RPF drops
- Per-interface Unicast RPF suppressed drops

The **show ip traffic** command shows the total number (global count) of dropped or suppressed packets as dropped by software; it does not include those dropped by hardware. The Unicast RPF drop count is included in the IP statistics section.

```
Switch# show ip traffic
```

```
IP statistics:
```

```
Rcvd: 1471590 total, 887368 local destination
      0 format errors, 0 checksum errors, 301274 bad hop count
      0 unknown protocol, 0 not a gateway
      0 security failures, 0 bad options, 0 with options
Opts: 0 end, 0 nop, 0 basic security, 0 loose source route
      0 timestamp, 0 extended security, 0 record route
      0 stream ID, 0 strict source route, 0 alert, 0 other
Frgs: 0 reassembled, 0 timeouts, 0 couldn't reassemble
      0 fragmented, 0 couldn't fragment
Bcast: 205233 received, 0 sent
Mcast: 463292 received, 462118 sent
Sent: 990158 generated, 282938 forwarded
! The second line below ("0 unicast RPF") displays Unicast RPF packet dropping
information.
Drop: 3 encapsulation failed, 0 unresolved, 0 no adjacency
      0 no route, 0 unicast RPF, 0 forced drop
```

A nonzero value for the count of dropped or suppressed packets can mean one of two things:

- Unicast RPF is dropping or suppressing packets that have a bad source address (normal operation).
- Unicast RPF is dropping or suppressing legitimate packets because the route is misconfigured to use Unicast RPF in environments where asymmetric routing exists; that is, where multiple paths can exist as the best return path for a source address.

The **show ip interface** command shows the total of dropped or suppressed packets at a specific interface. If Unicast RPF is configured to use a specific ACL, that ACL information is displayed along with the drop statistics.

```
Switch> show ip interface fast 2/1
```

```
Unicast RPF ACL 197
1 unicast RPF drop
1 unicast RPF suppressed drop
```

The **show access-lists** command displays the number of matches found for a specific entry in a specific access list.

```
Switch> show access-lists
```

```
Extended IP access list 197
deny ip 192.168.201.0 0.0.0.63 any log-input (1 match)
permit ip 192.168.201.64 0.0.0.63 any log-input (1 match)
deny ip 192.168.201.128 0.0.0.63 any log-input
permit ip 192.168.201.192 0.0.0.63 any log-input
```

Unicast RPF Configuration Example: Inbound and Outbound Filters

The following example uses a very simple single-homed ISP to demonstrate the concepts of ingress and egress filters used in conjunction with Unicast RPF. The example illustrates an ISP-allocated classless interdomain routing (CIDR) block 209.165.202.128/28 that has both inbound and outbound filters on the upstream interface. Be aware that ISPs are usually not single-homed. Provisions for asymmetrical flows (when outbound traffic goes out one link and returns by using a different link) must be designed into the filters on the border switches of the ISP.


```

ip cef distributed
!
interface Serial 5/0/0
  description Connection to Upstream ISP
  ip address 209.165.200.225 255.255.255.252
  no ip redirects
  no ip directed-broadcast
  no ip proxy-arp
  ip verify unicast reverse-path rx allow-default
  ip access-group 111 in
  ip access-group 110 out
!
access-list 110 permit ip 209.165.202.128 0.0.0.31 any
access-list 110 deny ip any any log
access-list 111 deny ip host 0.0.0.0 any log
access-list 111 deny ip 127.0.0.0 0.255.255.255 any log
access-list 111 deny ip 10.0.0.0 0.255.255.255 any log
access-list 111 deny ip 172.16.0.0 0.15.255.255 any log
access-list 111 deny ip 192.168.0.0 0.0.255.255 any log
access-list 111 deny ip 209.165.202.128 0.0.0.31 any log
access-list 111 permit ip any any

```

Unicast RPF with ACL Support Configuration Example:

The following example shows URPF with ACL support configured on an interface of the switch.

```

Switch# configure terminal
Switch(config)# interface gigabitethernet 0/0/1
Switch(config-if)# ip address 192.168.200.2 255.255.255.0
Switch(config-if)# ip verify unicast source reachable-via rx allow-default 100
Switch(config-if)# end

```

Feature Information for Unicast Reverse Path Forwarding

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.



Note

Table 38-1 lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 38-1 Feature Information for Power over Ethernet

Feature Name	Releases	Feature Information
Unicast Reverse Path Forwarding	Cisco IOS Release 12.2(40)SG	This feature was introduced.
	Cisco IOS Release 15.2(6)E2	The ACL support for the Unicast Reverse Path Forwarding feature was introduced for IPv4 family.



Configuring IP Multicast

This chapter describes IP multicast routing on the Catalyst 4500 series switch. It also provides procedures and examples to configure IP multicast routing.

This chapter includes the following major sections:

- [About IP Multicast, page 39-1](#)
- [Configuring IP Multicast Routing, page 39-13](#)
- [Monitoring and Maintaining IP Multicast Routing, page 39-23](#)
- [Configuration Examples, page 39-29](#)



Note

For more detailed information on IP Multicast, refer to this URL:

http://www.cisco.com/en/US/products/ps6552/products_ios_technology_home.html

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About IP Multicast



Note

Controlling the transmission rate to a multicast group is not supported.

At one end of the IP communication spectrum is IP unicast, where a source IP host sends packets to a specific destination IP host. In IP unicast, the destination address in the IP packet is the address of a single, unique host in the IP network. These IP packets are forwarded across the network from the source to the destination host by routers. At each point on the path between source and destination, a router uses a unicast routing table to make unicast forwarding decisions, based on the IP destination address in the packet.

At the other end of the IP communication spectrum is an IP broadcast, where a source host sends packets to all hosts on a network segment. The destination address of an IP broadcast packet has the host portion of the destination IP address set to all ones and the network portion set to the address of the subnet. IP hosts, including routers, understand that packets, which contain an IP broadcast address as the

destination address, are addressed to all IP hosts on the subnet. Unless specifically configured otherwise, routers do not forward IP broadcast packets, so IP broadcast communication is normally limited to a local subnet.

IP multicasting falls between IP unicast and IP broadcast communication. IP multicast communication enables a host to send IP packets to a *group* of hosts anywhere within the IP network. To send information to a specific group, IP multicast communication uses a special form of IP destination address called an *IP multicast group address*. The IP multicast group address is specified in the IP destination address field of the packet.

To multicast IP information, Layer 3 switches and routers must forward an incoming IP packet to all output interfaces that lead to *members* of the IP multicast group. In the multicasting process on the Catalyst 4500 series switch, a packet is replicated in the Integrated Switching Engine, forwarded to the appropriate output interfaces, and sent to each member of the multicast group.

We tend to think of IP multicasting and video conferencing as the same thing. Although the first application in a network to use IP multicast is often video conferencing, video is only one of many IP multicast applications that can add value to a company's business model. Other IP multicast applications that have potential for improving productivity include multimedia conferencing, data replication, real-time data multicasts, and simulation applications.

This section contains the following subsections:

- [IP Multicast Protocols, page 39-2](#)
- [IP Multicast Implementation on the Catalyst 4500 Series Switch, page 39-4](#)
- [Configuring IP Multicast Routing, page 39-13](#)

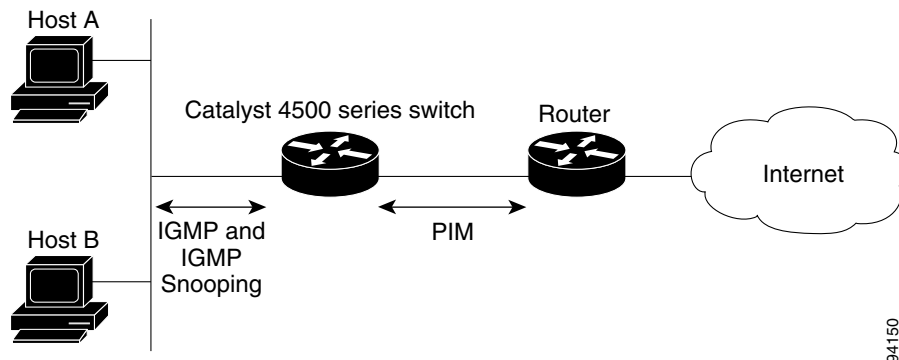
IP Multicast Protocols

The Catalyst 4500 series switch primarily uses these protocols to implement IP multicast routing:

- Internet Group Management Protocol (IGMP)
- Protocol Independent Multicast (PIM)
- IGMP snooping and Cisco Group Management Protocol

[Figure 39-1](#) shows where these protocols operate within the IP multicast environment.

Figure 39-1 IP Multicast Routing Protocols



Internet Group Management Protocol

IGMP messages are used by IP multicast hosts to send their local Layer 3 switch or router a request to join a specific multicast group and begin receiving multicast traffic. With some extensions in IGMPv2, IP hosts can also send a request to a Layer 3 switch or router to leave an IP multicast group and not receive the multicast group traffic.

Using the information obtained by using IGMP, a Layer 3 switch or router maintains a list of multicast group memberships on a per-interface basis. A multicast group membership is active on an interface if at least one host on the interface sends an IGMP request to receive multicast group traffic.

Protocol-Independent Multicast

PIM is *protocol independent* because it can leverage whichever unicast routing protocol is used to populate the unicast routing table, including EIGRP, OSPF, BGP, or static route, to support IP multicast. PIM also uses a unicast routing table to perform the reverse path forwarding (RPF) check function instead of building a completely independent multicast routing table. PIM does not send and receive multicast routing updates between routers like other routing protocols do.

PIM Dense Mode

PIM Dense Mode (PIM-DM) uses a *push* model to flood multicast traffic to every corner of the network. PIM-DM is intended for networks in which most LANs need to receive the multicast, such as LAN TV and corporate or financial information broadcasts. It can be an efficient delivery mechanism if active receivers exist on every subnet in the network.

For more detailed information on PIM Dense Mode, refer to this URL:

http://www.cisco.com/en/US/docs/ios-xml/ios/ipmulti_optim/configuration/12-2sx/imc_pim_dense_rfrsh.html

PIM Sparse Mode

PIM Sparse Mode (PIM-SM) uses a *pull* model to deliver multicast traffic. Only networks with active receivers that have explicitly requested the data are forwarded the traffic. PIM-SM is intended for networks with several different multicasts, such as desktop video conferencing and collaborative computing, that go to a small number of receivers and are typically in progress simultaneously.

Bidirectional PIM Mode

In bidirectional PIM (Bidir-PIM) mode, traffic is routed only along a bidirectional shared tree that is rooted at the rendezvous point (RP) for the group. The IP address of the RP functions as a key enabling all routers to establish a loop-free spanning tree topology rooted in that IP address.

Bidir-PIM is intended for many-to-many applications within individual PIM domains. Multicast groups in bidirectional mode can scale to an arbitrary number of sources without incurring overhead due to the number of sources.

For more detailed information on Bidirectional Mode, refer to this URL:

http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6552/ps6592/prod_white_paper0900aecd80310db2.pdf.

Rendezvous Point (RP)

If you configure PIM to operate in sparse mode, you must also choose one or more routers to be rendezvous points (RPs). Senders to a multicast group use RPs to announce their presence. Receivers of multicast packets use RPs to learn about new senders. You can configure Cisco IOS software so that packets for a single multicast group can use one or more RPs.

The RP address is used by first hop routers to send PIM register messages on behalf of a host sending a packet to the group. The RP address is also used by last hop routers to send PIM join and prune messages to the RP to inform it about group membership. You must configure the RP address on all routers (including the RP router).

A PIM router can be an RP for more than one group. Only one RP address can be used at a time within a PIM domain for the same group. The conditions specified by the access list determine for which groups the router is an RP (as different groups can have different RPs).

IGMP Snooping

IGMP snooping is used for multicasting in a Layer 2 switching environment. With IGMP snooping, a Layer 3 switch or router examines Layer 3 information in the IGMP packets in transit between hosts and a router. When the switch receives the IGMP Host Report from a host for a particular multicast group, the switch adds the host's port number to the associated multicast table entry. When the switch receives the IGMP Leave Group message from a host, it removes the host's port from the table entry.

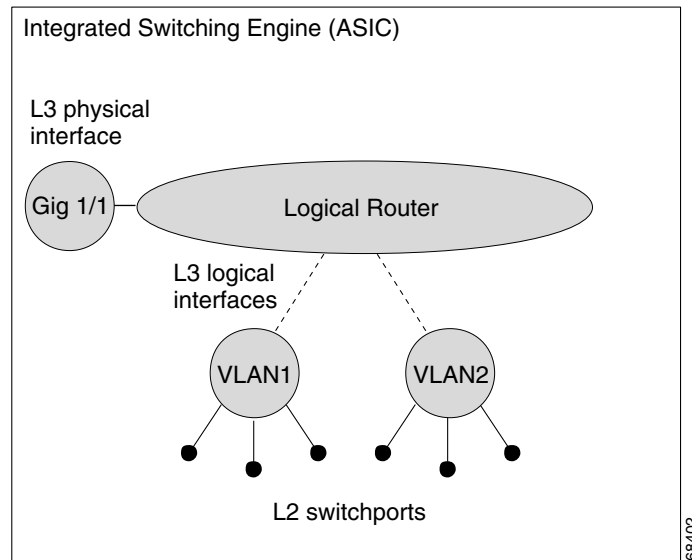
Because IGMP control messages are transmitted as multicast packets, they are indistinguishable from multicast data if only the Layer 2 header is examined. A switch running IGMP snooping examines every multicast data packet to determine whether it contains any pertinent IGMP control information. If IGMP snooping is implemented on a low end switch with a slow CPU, performance could be severely impacted when data is transmitted at high rates. On the Catalyst 4500 series switches, IGMP snooping is implemented in the forwarding ASIC, so it does not impact the forwarding rate.

IP Multicast Implementation on the Catalyst 4500 Series Switch

The Catalyst 4500 series switch supports an ASIC-based Integrated Switching Engine that provides Ethernet bridging at Layer 2 and IP routing at Layer 3. Because the ASIC is specifically designed to forward packets, the Integrated Switching Engine hardware provides very high performance with ACLs and QoS enabled. At wire-speed, forwarding in hardware is significantly faster than the CPU subsystem software, which is designed to handle exception packets.

The Integrated Switching Engine hardware supports interfaces for inter-VLAN routing and switch ports for Layer 2 bridging. It also provides a physical Layer 3 interface that can be configured to connect with a host, a switch, or a router.

[Figure 39-2](#) shows a logical view of Layer 2 and Layer 3 forwarding in the Integrated Switching Engine hardware.

Figure 39-2 Logical View of Layer 2 and Layer 3 Forwarding in Hardware

This section contains the following subsections:

- [Restrictions on IP Multicast, page 39-5](#)
- [CEF, MFIB, and Layer 2 Forwarding, page 39-6](#)
- [IP Multicast Tables, page 39-7](#)
- [Hardware and Software Forwarding, page 39-9](#)
- [Non-Reverse Path Forwarding Traffic, page 39-10](#)
- [Multicast Fast Drop, page 39-11](#)
- [Multicast Forwarding Information Base, page 39-12](#)
- [S/M, 224/4, page 39-13](#)
- [Multicast HA, page 39-13](#)

Restrictions on IP Multicast

Restrictions on IP Multicast include the following:

- Packets that have a multicast destination IP address and unicast MAC address are dropped.
- Starting with Release IOS XE 3.3.0SG and IOS 15.1(1)SG, the seven RP restriction was removed.
- IPv4 Bidirectional (Bidir) PIM is supported on the Catalyst 4500 series switch. IPv6 Bidir PIM is not.
- For some multicast groups, when more than 8K mroutes are installed in a system, the network may experience higher traffic losses upon switchover of the HA system. This is due to flushing the old multicast forwarding entries before the new entries are updated. As the number of routes increase, more time is required for the entries to be updated in the MFIB. To reduce the traffic loss in this scenario, you should increase the multicast route-flush timer (using the **ip multicast redundancy routeflush maxtime** command) to a value exceeding the default (30 seconds).

CEF, MFIB, and Layer 2 Forwarding

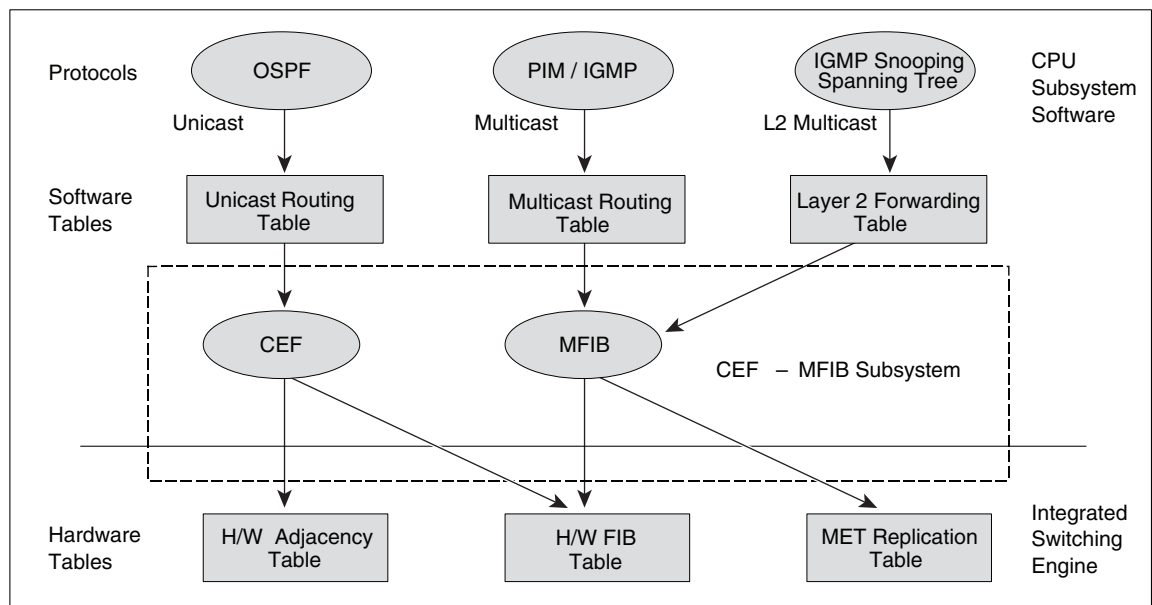
The implementation of IP multicast on the Catalyst 4500 series switch is an extension of centralized Cisco Express Forwarding (CEF). CEF extracts information from the unicast routing table, which is created by unicast routing protocols, such as BGP, OSPF, and EIGRP and loads it into the hardware Forwarding Information Base (FIB). With the unicast routes in the FIB, when a route is changed in the upper-layer routing table, only one route needs to be changed in the hardware routing state. To forward unicast packets in hardware, the Integrated Switching Engine looks up source and destination routes in ternary content addressable memory (TCAM), takes the adjacency index from the hardware FIB, and gets the Layer 2 rewrite information and next-hop address from the hardware adjacency table.

The new Multicast Forwarding Information Base (MFIB) subsystem is the multicast analog of the unicast CEF. The MFIB subsystem extracts the multicast routes that PIM and IGMP create and refines them into a protocol-independent format for forwarding in hardware. The MFIB subsystem removes the protocol-specific information and leaves only the essential forwarding information. Each entry in the MFIB table consists of an (S,G) or (*,G) route, an input RPF VLAN, and a list of Layer 3 output interfaces. The MFIB subsystem, together with platform-dependent management software, loads this multicast routing information into the hardware FIB and Replica Expansion Table (RET).

The Catalyst 4500 series switch performs Layer 3 routing and Layer 2 bridging at the same time. There can be multiple Layer 2 switch ports on any VLAN interface.

Figure 39-3 shows a functional overview of how the Catalyst 4500 series switch combines unicast routing, multicast routing, and Layer 2 bridging information to forward in hardware.

Figure 39-3 Combining CEF, MFIB, and Layer 2 Forwarding Information in Hardware



Like the CEF unicast routes, the MFIB routes are Layer 3 and must be merged with the appropriate Layer 2 information. The following example shows an MFIB route:

```

(*,224.1.1.3)
RPF interface is Vlan3
Output Interfaces are:
Vlan 1
Vlan 2
  
```


The route (*,224.1.2.3) is loaded in the hardware FIB table and the list of output interfaces is loaded into the MET. A pointer to the list of output interfaces, the MET index, and the RPF interface are also loaded in the hardware FIB with the (*,224.1.2.3) route. With this information loaded in hardware, merging of the Layer 2 information can begin. For the output interfaces on VLAN1, the Integrated Switching Engine must send the packet to all switch ports in VLAN1 that are in the spanning tree forwarding state. The same process applies to VLAN 2. To determine the set of switch ports in VLAN 2, the Layer 2 Forwarding Table is used.

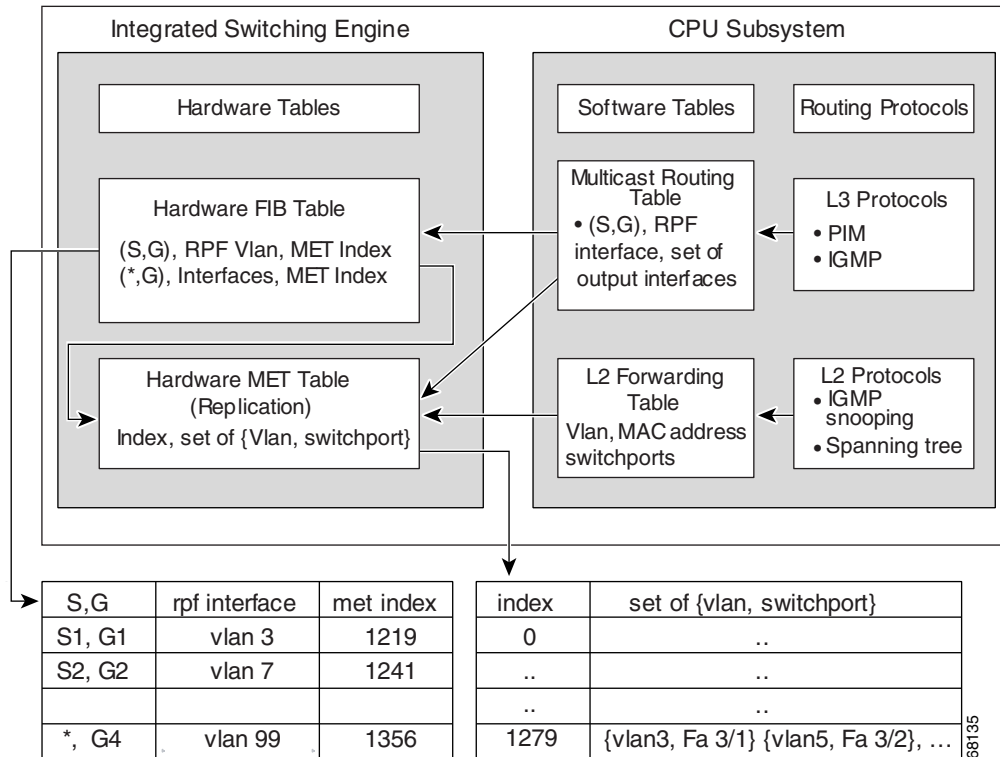
When the hardware routes a packet, in addition to sending it to all of the switch ports on all output interfaces, the hardware also sends the packet to all switch ports (other than the one it arrived on) in the input VLAN. For example, assume that VLAN 3 has two switch ports in it, Gig 3/1 and Gig 3/2. If a host on Gig 3/1 sends a multicast packet, the host on Gig 3/2 might also need to receive the packet. To send a multicast packet to the host on Gig 3/2, all of the switch ports in the ingress VLAN must be added to the port set that is loaded in the MET.

If VLAN 1 contains 1/1 and 1/2, VLAN 2 contains 2/1 and 2/2, and VLAN 3 contains 3/1 and 3/2, the MET chain for this route would contain these switch ports: (1/1,1/2,2/1,2/2,3/1, and 3/2).

If IGMP snooping is on, the packet should not be forwarded to all output switch ports on VLAN 2. The packet should be forwarded only to switch ports where IGMP snooping has determined that there is either a group member or router. For example, if VLAN 1 had IGMP snooping enabled, and IGMP snooping determined that only port 1/2 had a group member on it, then the MET chain would contain these switch ports: (1/1,1/2, 2/1, 2/2, 3/1, and 3/2).

IP Multicast Tables

Figure 39-4 shows some key data structures that the Catalyst 4500 series switch uses to forward IP multicast packets in hardware.

Figure 39-4 IP Multicast Tables and Protocols

The Integrated Switching Engine maintains the hardware FIB table to identify individual IP multicast routes. Each entry consists of a destination group IP address and an optional source IP address. Multicast traffic flows on primarily two types of routes: (S,G) and (*,G). The (S,G) routes flow from a source to a group based on the IP address of the multicast source and the IP address of the multicast group destination. Traffic on a (*,G) route flows from the PIM RP to all receivers of group G. Only sparse-mode groups use (*,G) routes. The Integrated Switching Engine hardware contains space for a total of 128,000 routes, which are shared by unicast routes, multicast routes, and multicast fast-drop entries.

Output interface lists are stored in the multicast expansion table (MET). The MET has room for up to 32,000 output interface lists. (For RET, we can have up to 102 K entries (32 K used for floodsets, 70,000 used for multicast entries)). The MET resources are shared by both Layer 3 multicast routes and by Layer 2 multicast entries. The actual number of output interface lists available in hardware depends on the specific configuration. If the total number of multicast routes exceed 32,000, multicast packets might not be switched by the Integrated Switching Engine. They would be forwarded by the CPU subsystem at much slower speeds.

**Note**

For RET, a maximum of 102 K entries is supported (32 K used for floodsets, 70 K used for multicast entries).

**Note**

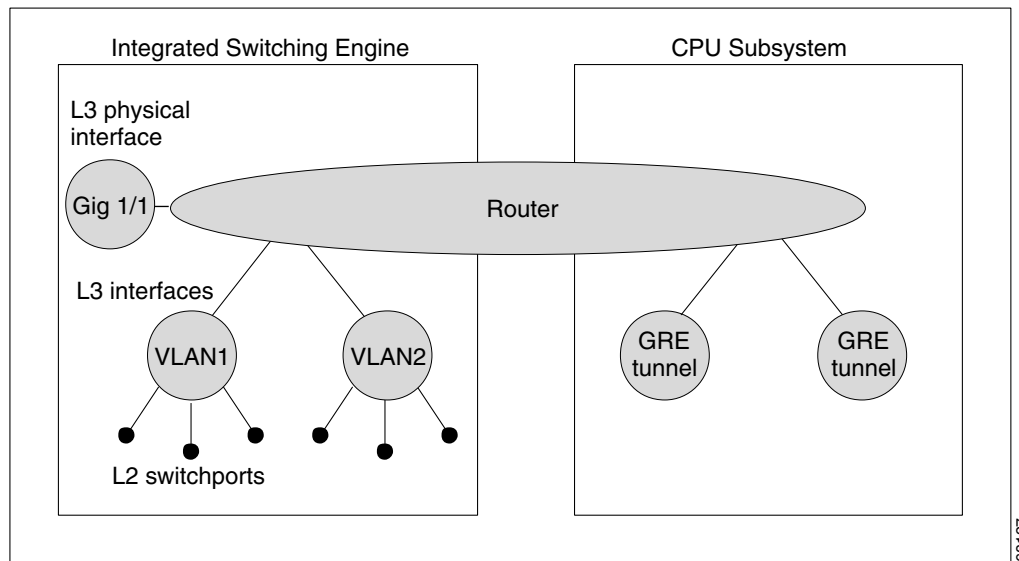
Prior to Release IOS XE 3.3.0SG and IOS 15.1(1)SG, partial routing is not supported on Supervisor Engine 6-E, Supervisor Engine 6L-E, Supervisor Engine 7-E, and Supervisor Engine 7L-E; only hardware and software routing are supported. Starting with Release IOS XE 3.3.0SG and IOS 15.1(1)SG, partial routing is supported on all supervisor engines.

Hardware and Software Forwarding

The Integrated Switching Engine forwards the majority of packets in hardware at very high rates of speed. The CPU subsystem forwards exception packets in software. Statistical reports should show that the Integrated Switching Engine is forwarding the vast majority of packets in hardware.

Figure 39-5 shows a logical view of the hardware and software forwarding components.

Figure 39-5 Hardware and Software Forwarding Components



In the normal mode of operation, the Integrated Switching Engine performs inter-VLAN routing in hardware. The CPU subsystem supports generic routing encapsulation (GRE) tunnels for forwarding in software.

Replication is a particular type of forwarding where, instead of sending out one copy of the packet, the packet is replicated and multiple copies of the packet are sent out. At Layer 3, replication occurs only for multicast packets; unicast packets are never replicated to multiple Layer 3 interfaces. In IP multicasting, for each incoming IP multicast packet that is received, many replicas of the packet are sent out.

IP multicast packets can be transmitted on the following types of routes:

- Hardware routes
- Software routes
- Partial routes

Hardware routes occur when the Integrated Switching Engine hardware forwards all replicas of a packet. Software routes occur when the CPU subsystem software forwards all replicas of a packet. Partial routes occur when the Integrated Switching Engine forwards some of the replicas in hardware and the CPU subsystem forwards some of the replicas in software.

Partial Routes



Note

The conditions listed below cause the replicas to be forwarded by the CPU subsystem software, but the performance of the replicas that are forwarded in hardware is not affected.

The following conditions cause some replicas of a packet for a route to be forwarded by the CPU subsystem:

- The switch is configured with the **ip igmp join-group** command as a member of the IP multicast group on the RPF interface of the multicast source.
- The switch is the first-hop to the source in PIM sparse mode. The switch must send PIM-register messages to the RP.

Software Routes



Note

If any one of the following conditions is configured on the RPF interface or the output interface, all replication of the output is performed in software.

The following conditions cause all replicas of a packet for a route to be forwarded by the CPU subsystem software:

- The interface is configured with multicast helper.
- The interface is a generic routing encapsulation (GRE) or Distance Vector Multicast Routing Protocol (DVMRP) tunnel.
- The interface uses non-Advanced Research Products Agency (ARPA) encapsulation.

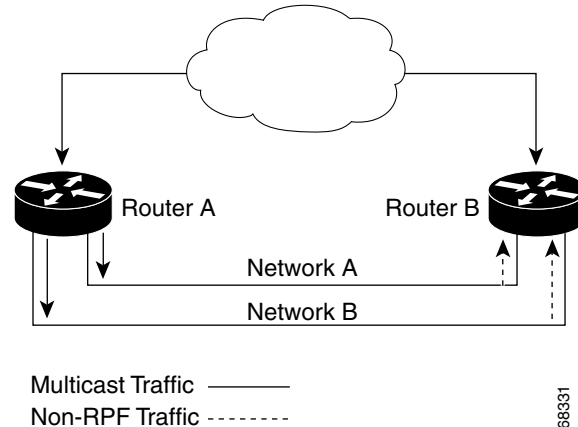
The following packets are always forwarded in software:

- Packets sent to multicast groups that fall into the range 224.0.0.* (where * is in the range from 0 to 255). This range is used by routing protocols. Layer 3 switching supports all other multicast group addresses.
- Packets with IP options.

Non-Reverse Path Forwarding Traffic

Traffic that fails an Reverse Path Forwarding (RPF) check is called non-RPF traffic. Non-RPF traffic is forwarded by the Integrated Switching Engine by filtering (persistently dropping) or rate limiting the non-RPF traffic.

In a redundant configuration where multiple Layer 3 switches or routers connect to the same LAN segment, only one device forwards the multicast traffic from the source to the receivers on the outgoing interfaces. [Figure 39-6](#) shows how non-RPF traffic can occur in a common network configuration.

Figure 39-6 Redundant Multicast Router Configuration in a Stub Network

In this kind of topology, only Router A, the PIM designated router (PIM DR), forwards data to the common VLAN. Router B receives the forwarded multicast traffic, but must drop this traffic because it has arrived on the wrong interface and fails the RPF check. Traffic that fails the RPF check is called non-RPF traffic.

Multicast Fast Drop

In IP multicast protocols, such as PIM-SM and PIM-DM, every (S,G) or (*,G) route has an incoming interface associated with it. This interface is referred to as the reverse path forwarding interface. In some cases, when a packet arrives on an interface other than the expected RPF interface, the packet must be forwarded to the CPU subsystem software to allow PIM to perform special protocol processing on the packet. One example of this special protocol processing that PIM performs is the PIM Assert protocol.

By default, the Integrated Switching Engine hardware sends all packets that arrive on a non-RPF interface to the CPU subsystem software. However, processing in software is not necessary in many cases, because these non-RPF packets are often not needed by the multicast routing protocols. The problem is that if no action is taken, the non-RPF packets that are sent to the software can overwhelm the CPU.

Prior to Release IOS XE 3.3.0SG and IOS 15.1(1)SG, to prevent this situation from happening, the CPU subsystem software would load fast-drop entries in the hardware when it receives an RPF failed packet that is not needed by the PIM protocols running on the switch. Any packet matching a fast-drop entry would be bridged in the ingress VLAN, but is not sent to the software so the CPU subsystem is not overloaded by processing these RPF failures unnecessarily. However, this process involved maintaining fast-drop entries in hardware. Because the FLCAM space is limited, the number of fast-drop entries installed in hardware was also limited.

Beginning with Release IOS XE 3.3.0SG and IOS 15.1(1)SG, rather than installing fast-drop entries, your switch uses Dynamic Buffer Limiting (DBL). This flow-based congestion avoidance mechanism provides active queue management by tracking the queue length for each traffic flow. When the queue length of a flow exceeds its set limit, DBL drops packets. Rate DBL limits the non-rpf traffic to the cpu subsystem so that the CPU is not overwhelmed. The packets are rate limited per flow to the CPU. Because installing fast-drop entries in the CAM is inaccessibly, the number of fast-drop flows that can be handled by the switch need not be limited.

Protocol events, such as a link going down or a change in the unicast routing table, can impact the set of packets that can safely be fast dropped. A packet that was correctly fast dropped before might, after a topology change, need to be forwarded to the CPU subsystem software so that PIM can process it. The CPU subsystem software handles flushing fast-drop entries in response to protocol events so that the PIM code in IOS can process all the necessary RPF failures.

The use of fast-drop entries in the hardware is critical in some common topologies because you may have persistent RPF failures. Without the fast-drop entries, the CPU is exhausted by RPF failed packets that it did not need to process.

Multicast Forwarding Information Base

The Multicast Forwarding Information Base (MFIB) subsystem supports IP multicast routing in the Integrated Switching Engine hardware on the Catalyst 4500 series switch. The MFIB logically resides between the IP multicast routing protocols in the CPU subsystem software (PIM, IGMP, MSDP, MBGP, and DVMRP) and the platform-specific code that manages IP multicast routing in hardware. The MFIB translates the routing table information created by the multicast routing protocols into a simplified format that can be efficiently processed and used for forwarding by the Integrated Switching Engine hardware.

To display the information in the multicast routing table, use the **show ip mroute** command. To display the MFIB table information, use the **show ip mfib** command.

The MFIB table contains a set of IP multicast routes. IP multicast routes include (S,G) and (*,G). Each route in the MFIB table can have one or more optional flags associated with it. The route flags indicate how a packet that matches a route should be forwarded. For example, the Internal Copy (IC) flag on an MFIB route indicates that a process on the switch needs to receive a copy of the packet. The following flags can be associated with MFIB routes:

- Internal Copy (IC) flag—Sets on a route when a process on the router needs to receive a copy of all packets matching the specified route.
- Signalling (S) flag—Sets on a route when a process needs to be notified when a packet matching the route is received; the expected behavior is that the protocol code updates the MFIB state in response to receiving a packet on a signalling interface.
- Connected (C) flag—When set on an MFIB route, has the same meaning as the Signaling (S) flag, except that the C flag indicates that only packets sent by directly connected hosts to the route should be signaled to a protocol process.

A route can also have a set of optional flags associated with one or more interfaces. For example, an (S,G) route with the flags on VLAN 1 indicates how packets arriving on VLAN 1 should be handled, and whether packets matching the route should be forwarded onto VLAN 1. The per-interface flags supported in the MFIB include the following:

- Accepting (A)—Sets on the interface that is known in multicast routing as the RPF interface. A packet that arrives on an interface that is marked as Accepting (A) is forwarded to all Forwarding (F) interfaces.
- Forwarding (F)—Used in conjunction with the Accepting (A) flag as described above. The set of Forwarding interfaces that form what is often referred to as the multicast “olist” or output interface list.
- Signaling (S)—Sets on an interface when some multicast routing protocol process in Cisco IOS needs to be notified of packets arriving on that interface.

**Note**

When PIM-SM routing is in use, the MFIB route might include an interface as in this example:

```
PimTunnel [1.2.3.4].
```

it is a virtual interface that the MFIB subsystem creates to indicate that packets are being tunnelled to the specified destination address. A PimTunnel interface cannot be displayed with the normal **show interface** command.

S/M, 224/4

An (S/M, 224/4) entry is created in the MFIB for every multicast-enabled interface. This entry ensures that all packets sent by directly connected neighbors can be register-encapsulated to the PIM-SM RP. Typically, only a small number of packets are forwarded using the (S/M,224/4) route, until the (S,G) route is established by PIM-SM.

For example, on an interface with IP address 10.0.0.1 and netmask 255.0.0.0, a route is created matching all IP multicast packets in which the source address is anything in the class A network 10. This route can be written in conventional subnet/masklength notation as (10/8,224/4). If an interface has multiple assigned IP addresses, then one route is created for each such IP address.

Multicast HA

Starting with Release IOS XE 3.4.0SG and IOS 15.1(2)SG, the Catalyst 4500 series switches support multicast HA, which ensures uninterrupted multicast traffic flow in the event of a supervisor engine failure. MFIB states are synced to the standby supervisor engine prior to a switchover, ensuring NSF availability with a fast convergence upon switchover during a supervisor engine failure. Multicast HA (SSO / NSF / ISSU) is supported for the PIM Sparse, Dense, Bidir, and SSM Modes; and at Layer 2 for IGMP and MLD Snooping.

For details on Multicast HA, please refer to the following URLs:

http://www.cisco.com/en/US/docs/ios-xml/ios/ipmulti_resil/configuration/xe-3s/imc_high_availability.html

Configuring IP Multicast Routing

The following sections describe IP multicast routing configuration tasks:

- [Default Configuration in IP Multicast Routing, page 39-14](#)
- [Enabling IP Multicast Routing, page 39-14](#)
- [Enabling PIM on an Interface, page 39-15](#)
- [Enabling Bidirectional Mode, page 39-16](#)
- [Enabling PIM-SSM Mapping, page 39-17](#)
- [Configuring a Rendezvous Point, page 39-17](#)
- [Configuring a Single Static RP, page 39-21](#)
- [Load Splitting of IP Multicast Traffic, page 39-22](#)

For more detailed information on IP multicast routing, such as Auto-RP, PIM Version 2, and IP multicast static routes, refer to the *Cisco IOS IP and IP Routing Configuration Guide, Cisco IOS Release 12.3*.

Default Configuration in IP Multicast Routing

Table 39-1 shows the IP multicast default configuration.

Table 39-1 Default IP Multicast Configuration

Feature	Default Value
Rate limiting of RPF	Enabled globally
IP multicast routing	Disabled globally Note When IP multicast routing is disabled, IP multicast traffic data packets are not forwarded by the Catalyst 4500 series switch. However, IP multicast control traffic continues to be processed and forwarded. IP multicast routes can remain in the routing table even if IP multicast routing is disabled.
PIM	Disabled on all interfaces
IGMP snooping	Enabled on all VLAN interfaces Note If you disable IGMP snooping on an interface, all output ports are forwarded by the Integrated Switching Engine. When IGMP snooping is disabled on an input VLAN interface, multicast packets related to that interface are sent to all forwarding switch ports in the VLAN.



Note

Source-specific multicast and IGMP v3 are supported.

For more information about source-specific multicast with IGMPv3 and IGMP, see the following URL:

http://www.cisco.com/en/US/docs/ios/ipmulti/configuration/guide/imc_cfg_ssm.html

Enabling IP Multicast Routing

Enabling IP multicast routing allows the Catalyst 4500 series switch to forward multicast packets. To enable IP multicast routing on the router, enter this command:

Command	Purpose
Switch(config)# ip multicast-routing	Enables IP multicast routing.

Enabling PIM on an Interface

Enabling PIM on an interface also enables IGMP operation on that interface. An interface can be configured to be in dense mode, sparse mode, or sparse-dense mode. The mode determines how the Layer 3 switch or router populates its multicast routing table and how the Layer 3 switch or router forwards multicast packets it receives from its directly connected LANs. You must enable PIM in one of these modes for an interface to perform IP multicast routing.

When the switch populates the multicast routing table, dense-mode interfaces are always added to the table. Sparse-mode interfaces are added to the table only when periodic join messages are received from downstream routers, or when there is a directly connected member on the interface. When forwarding from a LAN, sparse-mode operation occurs if there is an RP known for the group. If so, the packets are encapsulated and sent toward the RP. When no RP is known, the packet is flooded in a dense-mode fashion. If the multicast traffic from a specific source is sufficient, the receiver's first-hop router can send join messages toward the source to build a source-based distribution tree.

There is no default mode setting. By default, multicast routing is disabled on an interface.

Enabling Dense Mode

To configure PIM on an interface to be in dense mode, enter this command:

Command	Purpose
Switch(config-if)# ip pim dense-mode	Enables dense-mode PIM on the interface.

For an example of how to configure a PIM interface in dense mode, see the “[PIM Dense Mode Example](#)” section.

Enabling Sparse Mode

To configure PIM on an interface to be in sparse mode, enter this command:

Command	Purpose
Switch(config-if)# ip pim sparse-mode	Enables sparse-mode PIM on the interface.

For an example of how to configure a PIM interface in sparse mode, see the “[PIM Sparse Mode Example](#)” section.

Enabling Sparse-Dense Mode

When you enter either the **ip pim sparse-mode** or **ip pim dense-mode** command, sparseness or denseness is applied to the interface as a whole. However, some environments might require PIM to run in a single region in sparse mode for some groups and in dense mode for other groups.

An alternative to enabling only dense mode or only sparse mode is to enable sparse-dense mode. The interface is treated as dense mode if the group is in dense mode; the interface is treated in sparse mode if the group is in sparse mode. If you want to treat the group as a sparse group, and the interface is in sparse-dense mode, you must have an RP.

If you configure sparse-dense mode, the idea of sparseness or denseness is applied to the group on the switch, and the network manager should apply the same concept throughout the network.

Another benefit of sparse-dense mode is that Auto-RP information can be distributed in a dense-mode manner; yet, multicast groups for user groups can be used in a sparse-mode manner. You do not need to configure a default RP at the leaf routers.

When an interface is treated in dense mode, it is populated in a multicast routing table's outgoing interface list when either of the following is true:

- When members or DVMRP neighbors exist on the interface
- When PIM neighbors exist and the group has not been pruned

When an interface is treated in sparse mode, it is populated in a multicast routing table's outgoing interface list when either of the following is true:

- When members or DVMRP neighbors exist on the interface
- When an explicit join has been received by a PIM neighbor on the interface

To enable PIM to operate in the same mode as the group, enter this command:

Command	Purpose
Switch(config-if) # ip pim sparse-dense-mode	Enables PIM to operate in sparse or dense mode, depending on the group.

Enabling Bidirectional Mode

Most of the configuration requirements for Bidir-PIM are the same as those for configuring PIM-SM. You need not enable or disable an interface for carrying traffic for multicast groups in bidirectional mode. Instead, you configure which multicast groups you want to operate in bidirectional mode. Similar to PIM-SM, you can perform this configuration with Auto-RP, static RP configurations, or the PIM Version 2 bootstrap router (PIMv2 BSR) mechanism.

To enable Bidir-PIM, perform this task in global configuration mode:

Command	Purpose
Switch(config) # ip pim bidir-enable	Enables bidir-PIM on a switch.

To configure Bidir-PIM, enter one of these commands, depending on which method you use to distribute group-to-RP mappings:

Command	Purpose
Switch(config)# ip pim rp-address <i>rp-address</i> [<i>access-list</i>] [<i>override</i>] bidir	Configures the address of a PIM RP for a particular group, and specifies bidirectional mode. Use this command when you are not distributing group-to-RP mappings using either Auto-RP or the PIMv2 BSR mechanism
Switch(config)# ip pim rp-candidate <i>type number</i> [<i>group-list</i> <i>access-list</i>] bidir	Configures the router to advertise itself as a PIM Version 2 candidate RP to the BSR, and specifies bidirectional mode. Use this command when you are using the PIMv2 BSR mechanism to distribute group-to-RP mappings.
Switch(config)# ip pim send-rp-address <i>type number scope ttl-value</i> [<i>group-list</i> <i>access-list</i>] [<i>interval</i> <i>seconds</i>] bidir	Configures the router to use Auto-RP to configure the groups the router is willing to act as RP, and specifies bidirectional mode. Use this command when you are using Auto-RP to distribute group-to-RP mappings.

For an example of how to configure bidir-PIM, see the “[Bidirectional PIM Mode Example](#)” section on page 39-29.

Enabling PIM-SSM Mapping

The Catalyst 4500 series switch supports SSM mapping, enabling an SSM transition in cases either where neither URD nor IGMP v3-lite is available, or when supporting SSM on the end system is impossible or unwanted due to administrative or technical reasons. With SSM mapping, you can leverage SSM for video delivery to legacy set-top boxes (STBs) that do not support IGMPv3 or for applications that do not take advantage of the IGMPv3 host stack.

For more details, refer to this URL:

http://www.cisco.com/en/US/docs/ios-xml/ios/ipmulti_igmp/configuration/15-s/imc_ssm_map.html

Configuring a Rendezvous Point

A rendezvous point (RP) is required in networks running Protocol Independent Multicast sparse mode (PIM-SM). In PIM-SM, traffic is forwarded only to network segments with active receivers that have explicitly requested multicast data.

The most commonly used methods to configure a rendezvous point (described here) are the use of Static RP and the use of the Auto-RP protocol. Another method (not described here) is the use of the Bootstrap Router (BSR) protocol.

Configuring Auto-RP

Auto-rendezvous point (Auto-RP) automates the distribution of group-to-rendezvous point (RP) mappings in a PIM network. To make Auto-RP work, a router must be designated as an RP mapping agent, which receives the RP announcement messages from the RPs and arbitrates conflicts. The RP mapping agent then sends the consistent group-to-RP mappings to all other routers by way of dense mode flooding.

All routers automatically discover which RP to use for the groups they support. The Internet Assigned Numbers Authority (IANA) has assigned two group addresses, 224.0.1.39 and 224.0.1.40, for Auto-RP.

The mapping agent receives announcements of intention to become the RP from Candidate-RPs. The mapping agent then announces the winner of the RP election. This announcement is made independently of the decisions by the other mapping agents.

To configure a rendezvous point, perform this task:

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode.
Step 2	Switch# configure terminal	Enters global configuration mode.
Step 3	Switch(config)# ip multicast-routing	Enables IP multicast routing.
Step 4	Switch(config)# interface [FastEthernet GigabitEthernet Loopback Null Port-channel TenGigabitEthernet Tunnel Vlan] <i>number</i>	Selects an interface that is connected to hosts on which PIM can be enabled.
Step 5	Switch(config-if)# ip pim [sparse-mode sparse-dense-mode]	Enables PIM sparse or sparse-dense mode on an interface. When configuring Auto-RP in sparse mode, you must also configure the Auto-RP listener in the next step.
Step 6	Switch(config-if)# exit	Returns to global configuration mode.
Step 7	Repeat Steps 4 and 5 on all PIM interfaces.	—
Step 8	Switch(config)# ip pim autorp listener	Causes IP multicast traffic for the two Auto-RP groups 224.0.1.39 and 224.0.1.40 to be PIM dense mode flooded across interfaces operating in PIM sparse mode. <ul style="list-style-type: none"> • Skip this step if you are configuring sparse-dense mode in Step 8.

	Command or Action	Purpose
Step 9	<pre>Switch(config)# ip pim send-rp-announce {interface-type interface-number ip-address} scope ttl-value [group-list access-list] [interval seconds] [bidir]</pre>	<p>Sends RP announcements out all PIM-enabled interfaces.</p> <ul style="list-style-type: none"> • Perform this step on the RP router only. • Use the <i>interface-type</i> and <i>interface-number</i> arguments to define which IP address is to be used as the RP address. • Use the <i>ip-address</i> argument to specify a directly connected IP address as the RP address. <p>Note If the <i>ip-address</i> argument is configured for this command, the RP-announce message is sourced by the interface to which this IP address is connected (that is, the source address in the IP header of the RP-announce message is the IP address of that interface).</p> <ul style="list-style-type: none"> • This example shows that the interface is enabled with a maximum of 31 hops. The IP address by which the router wants to be identified as RP is the IP address associated with loopback interface 0. Access list 5 describes the groups for which this router serves as RP.
Step 10	<pre>Switch(config)# ip pim send-rp-discovery [interface-type interface-number] scope ttl-value [interval seconds]</pre>	<p>Configures the router to be an RP mapping agent.</p> <ul style="list-style-type: none"> • Perform this step on the RP router only. • Use the optional <i>interface-type</i> and <i>interface-number</i> arguments to define which IP address is to be used as the source address of the RP mapping agent. • Use the scope keyword and <i>ttl-value</i> argument to specify the Time-to-Live (TTL) value in the IP header of Auto-RP discovery messages. • Use the optional interval keyword and <i>seconds</i> argument to specify the interval at which Auto-RP discovery messages are sent. <p>Note Lowering the interval at which Auto-RP discovery messages are sent from the default value of 60 seconds results in more frequent floodings of the group-to-RP mappings. In some network environments, the disadvantages of lowering the interval (more control packet overhead) may outweigh the advantages (more frequent Requirementsgroup-to-RP mapping updates).</p> <ul style="list-style-type: none"> • The example shows limiting the Auto-RP discovery messages to 31 hops on loopback interface 1.

	Command or Action	Purpose
Step 11	Switch(config)# ip pim rp-announce-filter rp-list access-list group-list access-list	Filters incoming Auto-RP announcement messages coming from the RP. <ul style="list-style-type: none"> Perform this step on the RP router only. Two example access lists that apply to this step could be: <pre>access-list 1 permit 10.0.0.1 access-list 1 permit 10.0.0.2 access-list 2 permit 224.0.0.0 15.255.255.255</pre>
Step 12	Switch(config)# interface type number	Selects an interface that is connected to hosts on which PIM can be enabled.
Step 13	Switch(config-if)# interface ethernet 1 ip multicast boundary access-list [filter-autorp]	Configures an administratively scoped boundary. <ul style="list-style-type: none"> Perform this step on the interfaces that are boundaries to other routers. The access list is not shown in this task. An access list entry that uses the deny keyword creates a multicast boundary for packets that match that entry.
Step 14	Switch(config-if)# end	Returns to EXEC mode.
Step 15	Switch# show ip pim autorp	(Optional) Displays the Auto-RP information.
Step 16	Switch# show ip pim rp [mapping] [rp-address]	(Optional) Displays RPs known in the network and shows how the router learned about each RP.
Step 17	Switch# show ip igmp groups [group-name group-address interface-type interface-number] [detail]	(Optional) Displays the multicast groups having receivers that are directly connected to the router and that were learned through Internet Group Management Protocol (IGMP). <ul style="list-style-type: none"> A receiver must be active on the network at the time that this command is issued to ensure the presence of receiver information on the resulting display.
Step 18	Switch# show ip mroute [group-address group-name] [source-address source-name] [interface-type interface-number] [summary] [count] [active kbps]	(Optional) Displays the contents of the IP multicast routing (mroute) table.

This example illustrates how to configure Auto-RP:

```
Switch> enable
Switch# configure terminal
Switch(config)# ip multicast-routing
Switch(config)# interface ethernet 1
Switch(config-if)# ip pim sparse-mode
Switch(config-if)# end
Switch(config)# ip pim autorp listener
Switch(config)# ip pim send-rp-announce loopback0 scope 31 group-list 5
Switch(config)# ip pim send-rp-discovery loopback 1 scope 31
Switch(config)# ip pim rp-announce-filter rp-list 1 group-list 2
Switch(config)# interface ethernet 1
Switch(config-if)# ip multicast boundary 10 filter-autorp
Switch(config-if)# end
Switch# show ip pim autorp
```

```
Switch# show ip pim rp mapping
Switch# show ip igmp groups
Switch# show ip mroute cbone-audio
```

Configuring a Single Static RP

If you are configuring PIM sparse mode, you must configure a PIM RP for a multicast group. An RP can either be configured statically in each device, or learned through a dynamic mechanism. This task explains how to statically configure an RP, as opposed to the router learning the RP through a dynamic mechanism such as Auto-RP.

PIM designated routers (DRs) forward data from directly connected multicast sources to the RP for distribution down the shared tree. Data is forwarded to the RP in one of two ways. It is encapsulated in register packets and unicast directly to the RP, or, if the RP has itself joined the source tree, it is multicast forwarded per the RPF forwarding algorithm. Last hop routers directly connected to receivers may, at their discretion, join themselves to the source tree and prune themselves from the shared tree.

A single RP can be configured for multiple groups that are defined by an access list. If no RP is configured for a group, the router treats the group as dense using the PIM dense mode techniques. (You can prevent this occurrence by configuring the **no ip pim dm-fallback** command.)

If a conflict exists between the RP configured with the **ip pim rp-address** command and one learned by Auto-RP, the Auto-RP information is used, unless the override keyword is configured.

To configure a single static RP, perform this task:

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode.
Step 2	Switch# configure terminal	Enters global configuration mode.
Step 3	Switch(config)# ip multicast-routing	Enables IP multicast routing.
Step 4	Switch(config)# interface <i>type number</i>	Selects an interface that is connected to hosts on which PIM can be enabled.
Step 5	Switch(config-if)# ip pim [sparse-mode sparse-dense-mode]	Enables PIM on an interface. You must use sparse mode.
Step 6	Repeat Steps 4 and 5 on every interface that uses IP multicast.	—
Step 7	Switch(config-if)# exit	Returns to global configuration mode.
Step 8	Switch(config)# ip pim rp-address <i>rp-address</i> [<i>access-list</i>] [override]	Configures the address of a PIM RP for a particular group. <ul style="list-style-type: none"> • Perform this step on any router. • The <i>access-list</i> argument specifies the number or name of an access list that defines for which multicast groups the RP should be used. • The override keyword specifies that if there is a conflict between the RP configured with this command and one learned by Auto-RP, the RP configured with this command prevails.
Step 9	Switch(config)# end	Ends the current configuration session and returns to EXEC mode.

	Command or Action	Purpose
Step 10	Switch# show ip pim rp [mapping] [rp-address]	(Optional) Displays RPs known in the network and shows how the router learned about each RP.
Step 11	Switch# show ip igmp groups [group-name group-address interface-type interface-number] [detail]	(Optional) Displays the multicast groups having receivers that are directly connected to the router and that were learned through IGMP. <ul style="list-style-type: none"> A receiver must be active on the network at the time that this command is issued to ensure that receiver information is present on the resulting display.
Step 12	Switch# show ip mroute [group-address group-name] [source-address source-name] [interface-type interface-number] [summary] [count] [active kbps]	(Optional) Displays the contents of the IP multicast routing (mroute) table.

This example shows how to configure a single-static RP:

```
Switch> enable
Switch# configure terminal
Switch(config)# ip multicast-routing
Switch(config)# interface ethernet 1
Switch(config-if)# ip pim sparse-mode
Switch(config-if)# exit
Switch(config)# ip pim rp-address 192.168.0.0
Switch(config)# end
Switch# show ip pim rp mapping
Switch# show ip igmp groups
Switch# show ip mroute cbone-audio
```

Load Splitting of IP Multicast Traffic



Note

This feature is only supported on Enterprise Services. It is not supported on IP Base and LAN Base.

If two or more equal-cost paths from a source are available, unicast traffic is load split across those paths. However, by default, multicast traffic is not load split across multiple equal-cost paths. In general, multicast traffic flows down from the reverse path forwarding (RPF) neighbor. According to the Protocol Independent Multicast (PIM) specifications, this neighbor must have the highest IP address if more than one neighbor has the same metric.

Use the **ip multicast multipath** command to enable load splitting of IP multicast traffic across multiple equal-cost paths.



Note

The **ip multicast multipath** command does not work with bidirectional Protocol Independent Multicast (PIM).

To enable IP multicast multipath, perform this task:

	Command	Purpose
Step 1	Switch# config t	Enters configuration mode.
Step 2	Switch(config)# ip multicast multipath	Enables IP multicast multipath.
Step 3	Switch(config)# end	Exits configuration mode.



Note

The **ip multicast multipath** command load splits the traffic but does not load balance the traffic. Traffic from a source uses only one path, even if the traffic far outweighs traffic from other sources.

Configuring load splitting with the **ip multicast multipath** command causes the system to load split multicast traffic across multiple equal-cost paths based on source address using the S-hash algorithm. When the **ip multicast multipath** command is configured and multiple equal-cost paths exist, the path in which multicast traffic travel is selected based on the source IP address. Multicast traffic from different sources is load split across the different equal-cost paths. Load splitting does not occur across equal-cost paths for multicast traffic from the same source sent to different multicast groups.

The following example shows how to enable ECMP multicast load splitting on a router based on a source address using the S-hash algorithm:

```
Switch(config)# ip multicast multipath
```

The following example shows how to enable ECMP multicast load splitting on a router based on a source and group address using the basic S-G-hash algorithm:

```
Switch(config)# ip multicast multipath s-g-hash basic
```

The following example shows how to enable ECMP multicast load splitting on a router based on a source, group, and next-hop address using the next-hop-based S-G-hash algorithm:

```
Switch(config)# ip multicast multipath s-g-hash next-hop-based
```

Monitoring and Maintaining IP Multicast Routing

You can remove all contents of a particular cache, table, or database. You also can display specific statistics. The following sections describe how to monitor and maintain IP multicast:

- [Displaying System and Network Statistics, page 39-24](#)
- [Displaying the Multicast Routing Table, page 39-24](#)
- [Displaying IP MFIB, page 39-26](#)
- [Displaying Bidirectional PIM Information, page 39-27](#)
- [Displaying PIM Statistics, page 39-28](#)
- [Clearing Tables and Databases, page 39-28](#)

Displaying System and Network Statistics

You can display specific statistics, such as the contents of IP routing tables and databases. Information provided can be used to determine resource utilization and solve network problems. You can also display information about node reachability and discover the routing path your device's packets are taking using the network.

To display various routing statistics, enter any of these commands:

Command	Purpose
Switch# ping [<i>group-name</i> <i>group-address</i>]	Sends an ICMP Echo Request to a multicast group address.
Switch# show ip mroute [<i>hostname</i> <i>group_number</i>]	Displays the contents of the IP multicast routing table.
Switch# show ip pim interface [<i>type number</i>] [<i>count</i>]	Displays information about interfaces configured for PIM.
Switch# show ip interface	Displays PIM information for all interfaces.

Displaying the Multicast Routing Table

The following is sample output from the **show ip mroute** command for a router operating in dense mode. This command displays the contents of the IP multicast FIB table for the multicast group named cbone-audio.

```
Switch# show ip mroute cbone-audio
```

```
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, C - Connected, L - Local, P - Pruned
R - RP-bit set, F - Register flag, T - SPT-bit set
Timers: Uptime/Expires
Interface state: Interface, Next-Hop, State/Mode

(*, 224.0.255.1), uptime 0:57:31, expires 0:02:59, RP is 0.0.0.0, flags: DC
  Incoming interface: Null, RPF neighbor 0.0.0.0, Dvmrp
  Outgoing interface list:
    Ethernet0, Forward/Dense, 0:57:31/0:02:52
    Tunnel0, Forward/Dense, 0:56:55/0:01:28

(198.92.37.100/32, 224.0.255.1), uptime 20:20:00, expires 0:02:55, flags: C
  Incoming interface: Tunnel0, RPF neighbor 10.20.37.33, Dvmrp
  Outgoing interface list:
    Ethernet0, Forward/Dense, 20:20:00/0:02:52
```

The following is sample output from the **show ip mroute** command for a router operating in sparse mode:

```
Switch# show ip mroute
```

```
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, C - Connected, L - Local, P - Pruned
R - RP-bit set, F - Register flag, T - SPT-bit set
Timers: Uptime/Expires
Interface state: Interface, Next-Hop, State/Mode

(*, 224.0.255.3), uptime 5:29:15, RP is 198.92.37.2, flags: SC
```

```

Incoming interface: Tunnel0, RPF neighbor 10.3.35.1, Dvmrp
Outgoing interface list:
    Ethernet0, Forward/Sparse, 5:29:15/0:02:57

(198.92.46.0/24, 224.0.255.3), uptime 5:29:15, expires 0:02:59, flags: C
Incoming interface: Tunnel0, RPF neighbor 10.3.35.1
Outgoing interface list:
    Ethernet0, Forward/Sparse, 5:29:15/0:02:57

```

**Note**

Interface timers are not updated for hardware-forwarded packets. Entry timers are updated approximately every five seconds.

The following is sample output from the **show ip mroute** command with the **summary** keyword:

```

Switch# show ip mroute summary

IP Multicast Routing Table
Flags: D - Dense, S - Sparse, C - Connected, L - Local, P - Pruned
      R - RP-bit set, F - Register flag, T - SPT-bit set, J - Join SPT
Timers: Uptime/Expires
Interface state: Interface, Next-Hop, State/Mode

(*, 224.255.255.255), 2d16h/00:02:30, RP 171.69.10.13, flags: SJPC
(*, 224.2.127.253), 00:58:18/00:02:00, RP 171.69.10.13, flags: SJ
(*, 224.1.127.255), 00:58:21/00:02:03, RP 171.69.10.13, flags: SJ
(*, 224.2.127.254), 2d16h/00:00:00, RP 171.69.10.13, flags: SJCL
(128.9.160.67/32, 224.2.127.254), 00:02:46/00:00:12, flags: CLJT
(129.48.244.217/32, 224.2.127.254), 00:02:15/00:00:40, flags: CLJT
(130.207.8.33/32, 224.2.127.254), 00:00:25/00:02:32, flags: CLJT
(131.243.2.62/32, 224.2.127.254), 00:00:51/00:02:03, flags: CLJT
(140.173.8.3/32, 224.2.127.254), 00:00:26/00:02:33, flags: CLJT
(171.69.60.189/32, 224.2.127.254), 00:03:47/00:00:46, flags: CLJT

```

The following is sample output from the **show ip mroute** command with the **active** keyword:

```

Switch# show ip mroute active

Active IP Multicast Sources - sending >= 4 kbps

Group: 224.2.127.254, (sdr.cisco.com)
  Source: 146.137.28.69 (mbone.ipd.anl.gov)
    Rate: 1 pps/4 kbps(1sec), 4 kbps(last 1 secs), 4 kbps(life avg)

Group: 224.2.201.241, ACM 97
  Source: 130.129.52.160 (webcast3-e1.acm97.interop.net)
    Rate: 9 pps/93 kbps(1sec), 145 kbps(last 20 secs), 85 kbps(life avg)

Group: 224.2.207.215, ACM 97
  Source: 130.129.52.160 (webcast3-e1.acm97.interop.net)
    Rate: 3 pps/31 kbps(1sec), 63 kbps(last 19 secs), 65 kbps(life avg)

```

The following is sample output from the **show ip mroute** command with the **count** keyword:

```

Switch# show ip mroute count

IP Multicast Statistics - Group count: 8, Average sources per group: 9.87
Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kilobits per second

Group: 224.255.255.255, Source count: 0, Group pkt count: 0
RP-tree: 0/0/0/0

```

```

Group: 224.2.127.253, Source count: 0, Group pkt count: 0
RP-tree: 0/0/0/0

Group: 224.1.127.255, Source count: 0, Group pkt count: 0
RP-tree: 0/0/0/0

Group: 224.2.127.254, Source count: 9, Group pkt count: 14
RP-tree: 0/0/0/0
Source: 128.2.6.9/32, 2/0/796/0
Source: 128.32.131.87/32, 1/0/616/0
Source: 128.125.51.58/32, 1/0/412/0
Source: 130.207.8.33/32, 1/0/936/0
Source: 131.243.2.62/32, 1/0/750/0
Source: 140.173.8.3/32, 1/0/660/0
Source: 146.137.28.69/32, 1/0/584/0
Source: 171.69.60.189/32, 4/0/447/0
Source: 204.162.119.8/32, 2/0/834/0

Group: 224.0.1.40, Source count: 1, Group pkt count: 3606
RP-tree: 0/0/0/0
Source: 171.69.214.50/32, 3606/0/48/0, RPF Failed: 1203

Group: 224.2.201.241, Source count: 36, Group pkt count: 54152
RP-tree: 7/0/108/0
Source: 13.242.36.83/32, 99/0/123/0
Source: 36.29.1.3/32, 71/0/110/0
Source: 128.9.160.96/32, 505/1/106/0
Source: 128.32.163.170/32, 661/1/88/0
Source: 128.115.31.26/32, 192/0/118/0
Source: 128.146.111.45/32, 500/0/87/0
Source: 128.183.33.134/32, 248/0/119/0
Source: 128.195.7.62/32, 527/0/118/0
Source: 128.223.32.25/32, 554/0/105/0
Source: 128.223.32.151/32, 551/1/125/0
Source: 128.223.156.117/32, 535/1/114/0
Source: 128.223.225.21/32, 582/0/114/0
Source: 129.89.142.50/32, 78/0/127/0
Source: 129.99.50.14/32, 526/0/118/0
Source: 130.129.0.13/32, 522/0/95/0
Source: 130.129.52.160/32, 40839/16/920/161
Source: 130.129.52.161/32, 476/0/97/0
Source: 130.221.224.10/32, 456/0/113/0
Source: 132.146.32.108/32, 9/1/112/0

```

**Note**

Multicast route byte and packet statistics are supported only for the first 1024 multicast routes. Output interface statistics are not maintained.

Displaying IP MFIB

You can display all routes in the MFIB, including routes that might not exist directly in the upper-layer routing protocol database but that are used to accelerate fast switching. These routes appear in the MFIB, even if dense-mode forwarding is in use.

To display various MFIB routing routes, enter one of these commands:

Command	Purpose
Switch# show ip mfib	Displays the (S,G) and (*,G) routes that are used for packet forwarding. Displays counts for fast, slow, and partially switched packets for every multicast route.
Switch# show ip mfib all	Displays all routes in the MFIB, including routes that may not exist directly in the upper-layer routing protocol database, but that are used to accelerate fast switching. These routes include the (S/M,224/4) routes.
Switch# show ip mfib log [n]	Displays a log of the most recent <i>n</i> MFIB-related events, the most recent first. <i>n</i> represents the number of events.

The following is sample output from the **show ip mfib** command:

```
IP Multicast Forwarding Information Base
Entry Flags: C - Directly Connected, S - Signal,
              IC - Internal Copy
Interface Flags: A - Accept, F - Forward, S - Signal,
                 NP - Not platform switched
Packets: Fast/Partial/Slow Bytes: Fast/Partial/Slow:
(171.69.10.13, 224.0.1.40), flags (IC)
  Packets: 2292/2292/0, Bytes: 518803/0/518803
  Vlan7 (A)
  Vlan100 (F NS)
  Vlan105 (F NS)
(*, 224.0.1.60), flags ()
  Packets: 2292/0/0, Bytes: 518803/0/0
  Vlan7 (A NS)
(*, 224.0.1.75), flags ()
  Vlan7 (A NS)
(10.34.2.92, 239.192.128.80), flags ()
  Packets: 24579/100/0, 2113788/15000/0 bytes
  Vlan7 (F NS)
  Vlan100 (A)
(*, 239.193.100.70), flags ()
  Packets: 1/0/0, 1500/0/0 bytes
  Vlan7 (A)
..
```

The fast-switched packet count represents the number of packets that were switched in hardware on the corresponding route.

The partially switched packet counter represents the number of times that a fast-switched packet was also copied to the CPU for software processing or for forwarding to one or more non-platform switched interfaces (such as a PimTunnel interface).

The slow-switched packet count represents the number of packets that were switched completely in software on the corresponding route.

Displaying Bidirectional PIM Information

To display bidir-PIM information, enter one of these commands:

Command	Purpose
Switch(config)# show ip pim interface [<i>type number</i>] [<i>df</i> <i>count</i>] [<i>rp-address</i>]	Displays information about the elected designated forward (DF) for each RP of an interface, along with the unicast routing metric associated with the DF.
Switch(config)# show ip pim rp [<i>mapping</i> <i>metric</i>] [<i>rp-address</i>]	Displays information about configured RPs, learned by using Auto-RP or BSR, along with their unicast routing metric.

Displaying PIM Statistics

The following is sample output from the **show ip pim interface** command:

```
Switch# show ip pim interface
```

Address	Interface	Mode	Neighbor Count	Query Interval	DR
198.92.37.6	Ethernet0	Dense	2	30	198.92.37.33
198.92.36.129	Ethernet1	Dense	2	30	198.92.36.131
10.1.37.2	Tunnel0	Dense	1	30	0.0.0.0

The following is sample output from the **show ip pim interface** command with a **count**:

```
Switch# show ip pim interface count
```

Address	Interface	FS	Mpackets In/Out
171.69.121.35	Ethernet0	*	548305239/13744856
171.69.121.35	Serial0.33	*	8256/67052912
198.92.12.73	Serial0.1719	*	219444/862191

The following is sample output from the **show ip pim interface** command with a **count** when IP multicast is enabled. The example lists the PIM interfaces that are fast-switched and process-switched, and the packet counts for these. The H is added to interfaces where IP multicast is enabled.

```
Switch# show ip pim interface count
```

```
States: FS - Fast Switched, H - Hardware Switched
Address      Interface      FS  Mpackets In/Out
192.1.10.2   Vlan10         * H 40886/0
192.1.11.2   Vlan11         * H 0/40554
192.1.12.2   Vlan12         * H 0/40554
192.1.23.2   Vlan23         *   0/0
192.1.24.2   Vlan24         *   0/0
```

Clearing Tables and Databases

You can remove all contents of a particular cache, table, or database. Clearing a cache, table, or database might be necessary when the contents of the particular structure have become, or are suspected to be, invalid.

To clear IP multicast caches, tables, and databases, enter one of these commands:

Command	Purpose
Switch# clear ip mroute	Deletes entries from the IP routing table.
Switch# clear ip mfib counters	Deletes all per-route and global MFIB counters.

**Note**

IP multicast routes can be regenerated in response to protocol events and as data packets arrive.

Configuration Examples

The following sections provide IP multicast routing configuration examples:

- [PIM Dense Mode Example, page 39-29](#)
- [PIM Sparse Mode Example, page 39-29](#)
- [Bidirectional PIM Mode Example, page 39-29](#)
- [Sparse Mode with a Single Static RP Example, page 39-30](#)
- [Sparse Mode with Auto-RP: Example, page 39-30](#)

PIM Dense Mode Example

This example is a configuration of dense-mode PIM on an Ethernet interface:

```
ip multicast-routing
interface ethernet 0
 ip pim dense-mode
```

PIM Sparse Mode Example

This example is a configuration of sparse-mode PIM. The RP router is the router with the address 10.8.0.20.

```
ip multicast-routing
 ip pim rp-address 10.8.0.20 1
interface ethernet 1
 ip pim sparse-mode
```

Bidirectional PIM Mode Example

By default, a bidirectional RP advertises all groups as bidirectional. Use an access list on the RP to specify a list of groups to be advertised as bidirectional. Groups with the **deny** keyword operate in dense mode. A different, nonbidirectional RP address is required for groups that operate in sparse mode, because a single access list only allows either a **permit** or **deny** keyword.

The following example shows how to configure an RP for both sparse and bidirectional mode groups. 224/8 and 227/8 are bidirectional groups, 226/8 is sparse mode, and 225/8 is dense mode. The RP must be configured to use different IP addresses for sparse and bidirectional mode operations. Two loopback

interfaces are used to allow this configuration and the addresses of these interfaces must be routed throughout the PIM domain so that the other routers in the PIM domain can receive Auto-RP announcements and communicate with the RP:

```
ip multicast-routing !Enable IP multicast routing
ip pim bidir-enable !Enable bidir-PIM
!
interface loopback 0
description One Loopback address for this routers Bidir Mode RP function
ip address 10.0.1.1 255.255.255.0
ip pim sparse-dense-mode
!
interface loopback 1
description One Loopback address for this routers Sparse Mode RP function
ip address 10.0.2.1 255.255.255.0
ip pim sparse-dense-mode
ip pim send-rp-announce Loopback0 scope 10 group-list 45 bidir
ip pim send-rp-announce Loopback1 scope 10 group-list 46
ip pim send-rp-discovery scope 10
access-list 45 permit 224.0.0.0 0.255.255.255
access-list 45 permit 227.0.0.0 0.255.255.255
access-list 45 deny 225.0.0.0 0.255.255.255
access-list 46 permit 226.0.0.0 0.255.255.255
```

Sparse Mode with a Single Static RP Example

The following example sets the PIM RP address to 192.168.1.1 for all multicast groups and defines all groups to operate in sparse mode:

```
ip multicast-routing
interface ethernet 1
 ip pim sparse-mode
ip pim rp-address 192.168.1.1
no ip pim dm-fallback
```



Note

The same RP cannot be used for both bidirectional and sparse mode groups.

The following example sets the PIM RP address to 172.16.1.1 for the multicast group 225.2.2.2 only:

```
access list 1 225.2.2.2 0.0.0.0
 ip pim rp-address 172.17.1.1
```

Sparse Mode with Auto-RP: Example

The following example configures sparse mode with Auto-RP:

```
ip multicast-routing
ip pim autorp listener
ip pim send-rp-announce Loopback0 scope 16 group-list 1
ip pim send-rp-discovery Loopback1 scope 16
no ip pim dm-fallback
access-list 1 permit 239.254.2.0 0.0.0.255
access-list 1 permit 239.254.3.0 0.0.0.255
.
.
access-list 10 permit 224.0.1.39
access-list 10 permit 224.0.1.40
access-list 10 permit 239.254.2.0 0.0.0.255
access-list 10 permit 239.254.3.0 0.0.0.255
```




Configuring ANCP Client

This chapter describes Access-Network Control Protocol (ANCP) Client on a Catalyst 4500 series switch. It includes the following sections:

- [About ANCP Client, page 40-1](#)
- [Enabling and Configuring ANCP Client, page 40-2](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About ANCP Client

ANCP Multicast enables you to control multicast traffic on a Catalyst 4500 series switch using either ANCP (instead of IGMP) or direct static configuration on the CLI. You can configure the switch as an ANCP client that connects to a remote ANCP server with multicast enabled. You can then initiate joins and leaves from that server. Use the switch in a system in which a subscriber requests that a digital right management (DRM) server receive a given channel (multicast) *potentially* through any private protocol mechanism.



Note

The ANCP client does not allow more than four multicast streams per-port per-VLAN. If a fifth join arrives, it is rejected.

If the digital right management (DRM) server determines that a subscriber is allowed to receive a multicast, it requests that the ANCP server sends an ANCP join command to the ANCP client (Catalyst 4500 series switch) for the port on which the subscriber is connected.



Note

IGMP snooping must be enabled on an ANCP client (Catalyst 4500 series switch) for processing multicast commands (join, leave, leave all requests, and request for active flows report) from the ANCP server. For information on enabling IGMP snooping, Refer to [Chapter 28, “Configuring IGMP Snooping and Filtering, and MVR.”](#)

The ANCP protocol must be able to identify the port on which multicast must be added. (This port can be identified either using the identifier configured on the CLI or with the DHCP option 82 that was inserted by the Catalyst 4500 switch while the subscriber received an IP address with DHCP. Either way, you should be consistent in identifying a given port.

Enabling and Configuring ANCP Client



Note

If you intend to use DHCP option 82 rather than CLI mapping (with the **ancp client port identif...** command) you must enter the **ip dhcp snooping** command before configuring the ANCP client.

You can identify a port with the **ancp mode client** command or with DHCP option 82.

This section includes these topics:

- [Identifying a Port with the ANCP Protocol, page 40-2](#)
- [Identifying a Port with DHCP Option 82, page 40-4](#)

Identifying a Port with the ANCP Protocol

To make the Catalyst 4500 series switch operate as an ANCP client and to build and initialize its relevant data, enter the **ancp mode client** command. The **no** version of this command disables ANCP. This command disconnects the ANCP client from the ANCP server and terminates any existing multicast streams that have been enabled with ANCP.

To configure a switch to communicate with a single ANCP server, use the **[no] ancp client server interface** command. This command directs the ANCP client to initiate a TCP connection to the remote ANCP server identified with the IP address. If the TCP connection fails, the connection times out and retries for the connection every 120 seconds until it succeeds. The **interface** command specifies the interface from which the local ANCP client obtains its IP address. The **no** command terminates the TCP connection to the ANCP server but retains any existing ANCP activated multicast stream.

Separate commands enable the ANCP client and configure the IP address of the ANCP server. You can reconfigure the IP address of the remote ANCP server without losing existing ANCP activated multicast streams.

To identify a port with the ANCP protocol, follow these steps:

Step 1 Enable ANCP as follows:

```
Switch(config)> ancp mode client
```

Step 2 Configure the IP address of the remote server as the interface to acquire the source IP address:

```
Switch(config)> ancp client server ipaddress of server interface interface
```

The interface might be a loopback; this allows the client to reach the server using the interface.

Step 3 (Optional) Enable the ANCP multicast client to identify this VLAN interface using the port-identifier as opposed to the Option 82 circuit-id:

```
Switch(config)> ancp client port identifier [port-identifier] vlan [number] interface [interface]
```

The **no** version of this command prompts a warning message if any multicast stream is activated by ANCP using the port-identifier on a port:

```
Switch(config)# no ancp client port identifier bbb vlan 10 interface GigabitEthernet3/5
Warning: Multicast flows seems to exist for this port, remove mapping and delete flows
anyway?[confirm]y
Switch(config)#
```

The ANCP client tries to connect to the server. If it fails, it tries again 10 seconds later. If it fails again, it tries at 20 seconds intervals, until it reaches the timeout setting (120 seconds). It remains timed out until it reconnects.



Note

If the connection fails again and the client attempts to reconnect and it fails, the wait time returns to 10 seconds (and so on).

To determine whether the ANCP client is successfully connected to the server, enter the **show ancp status** command, which displays the status of the ANCP TCP connection with the remote ANCP server.

```
Switch# show ancp status
ANCP enabled on following interfaces

Et0/0
  ANCP end point(s) on this interface:
  =====

  ANCP state ESTAB
  Neighbor 10.1.1.1 Neighbor port 6068
  Hello interval 100 Sender instance 1 Sender name 372F61C
  Sender port 0 Partition ID 0 TCB 36E27E8
  Capabilities negotiated: Transactional Multicast

Switch#
```

In the preceding example, only one capability (transactional multicast) is negotiated (or supported). This capability is the only one that the ANCP client supports. Because the server also supports this capability, the two entities can now communicate.

The server can send ANCP multicast commands (join, leave, leave all requests, and request for active flows report) as defined in the multicast portion of the ANCP protocol. At any time, an administrator can use the **show ancp multicast [interface vlan] [group | source]** command to see the information the ANCP client has obtained about the current multicast flows.

Example 1

```
ANCP_Client# show ancp multicast group 239.6.6.6
ANCP Multicast Streams
ClientID          VLAN  Interface          Joined on
  Group 239.6.6.6
0x0106000700130103  19   Gi1/3              15:06:23 UTC Tue Aug 26 2008
ANCP_Client#
```

Example 2

```
ANCP_Client# show ancp multicast interface Fa2/3 vlan 19
ANCP Multicast Streams
Interface FastEthernet2/3          VLAN 19: client ID 0x0106000700130203
  Group          Source          Joined on
  239.5.6.7      -              15:03:14 UTC Tue Aug 26 2008
ANCP_Client#
```



Note

Specifying the **show ancp multicast** command without parameters or keywords lists everything.

Identifying a Port with DHCP Option 82



Note

To use DHCP option 82, you need to enable DHCP and DHCP snooping (see [Chapter 60, “Configuring DHCP Snooping, IP Source Guard, and IPSG for Static Hosts”](#)).

If you identify the port with DHCP option 82, you need to configure the Catalyst 4500 series switch as a DHCP relay to insert the DHCP option 82. This action adds a tag in the DHCP packet from the DHCP client so that the DHCP server knows the port connected to this specific DHCP client. The DHCP server can then map the IP address it is providing to the client with the DHCP option 82 it received from the switch. The DHCP server only needs to lookup the DHCP option 82 associated with a given IP address and provide it to the ANCP server. This allows the ANCP client on the switch to identify the proper port using an identifier the switch understands. To configure DHCP snooping on the Catalyst 4500 series switch, use the following commands:

```
Switch(config)# ip dhcp snooping
Switch(config)# ip dhcp snooping vlan vlan-range
```

By default, DHCP option 82 is inserted when DHCP snooping is activated. Turning this default off could cause ANCP to function improperly with the DHCP circuit-id; it must remain active. To activate it, enter this command:

```
Switch(config)# ip dhcp snooping information option
```



Note

The DHCP option 82 circuit-ID is inserted in the Active-Flow report (when queried for all multicast flows) even if a configured circuit-ID exists.

ANCP allows a remote server to request the list of active flows from the ANCP client (Catalyst 4500 series switch is the ANCP client). This list is very similar to the output from the **show ancp multicast** command except that it follows the ANCP protocol packet format (see IETF.org). Observe that the **show ancp multicast** command provides the flows that have been activated with the **anyp port client identifier** command while the ANCP active flow request only reports the client ID in DHCP option 82 circuit-ID format, regardless of the activation mechanism.

Refer to [Chapter 60, “Configuring DHCP Snooping, IP Source Guard, and IPSG for Static Hosts”](#) for details on the CLI.

ANCP Guidelines and Restrictions

When using (or configuring) ANCP, consider these guidelines and restrictions:

- Entering a **shut** command on a port removes ANCP activated multicast streams from the port. They must be reactivated by the ANCP server.
- Entering a **suspend** or **shut** command on a VLAN removes ANCP-activated multicast streams from the VLAN.
- Deleting a VLAN removes ANCP-activated multicast streams from the VLAN.
- If a port enters the errdisable or blocked state, ANCP-activated multicast streams are removed from the port.
- Disabling IGMP snooping globally or per-VLAN might disrupt ANCP client functionality.
- An ANCP client does not account for the Layer 3 interface state changes (if PIM interface at Layer 3 shuts down, ANCP does not remove the streams). When a PIM interface is running again, multicast streams are received by subscribers.



Configuring Bidirectional Forwarding Detection



Note

Bidirectional Forwarding Detection (BFD) support was introduced Starting with Cisco IOS Release IOS 15.1(1)SG, on Supervisor Engine 6-E, Supervisor Engine 6L-E. Starting with Cisco IOS XE 3.5.0E and IOS 15.2(1)E, on Supervisor Engine 7-E, and 7L-E. Starting with Cisco IOS XE 3.6.0E and IOS 15.2(2)E, on Supervisor Engine 8-E. Starting with Cisco IOS XE 3.10.0E, on Supervisor Engine 9-E

This document describes how to enable the BFD protocol, which is a detection protocol designed to provide fast forwarding path failure detection times for all media types, encapsulations, topologies, and routing protocols.

BFD provides a consistent failure detection method for network administrators in addition to fast forwarding path failure detection. Because the network administrator can use BFD to detect forwarding path failures at a uniform rate, rather than the variable rates for different routing protocol hello mechanisms, network profiling and planning are simplified, and reconvergence time is more consistent and predictable.



Note

For details on all the BFD commands introduced in this chapter, see the URL:

http://www.cisco.com/en/US/docs/ios/iproute_pi/command/reference/iri_book.html

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the [“Technical Assistance” section on page 41-29](#).

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Contents

- [Prerequisites for Bidirectional Forwarding Detection, page 41-2](#)
- [Restrictions for Bidirectional Forwarding Detection, page 41-2](#)
- [Information About Bidirectional Forwarding Detection, page 41-3](#)
- [How to Configure Bidirectional Forwarding Detection, page 41-8](#)
- [Configuration Examples for Bidirectional Forwarding Detection, page 41-17](#)
- [Additional References, page 41-28](#)

Prerequisites for Bidirectional Forwarding Detection

Prerequisites include:

- IP routing must be enabled on all participating switches.
- One of the IP routing protocols supported by BFD must be configured on the switches before BFD is deployed. You should implement fast convergence for the routing protocol that you plan to use. See the IP routing documentation for your version of Cisco IOS software for information on configuring fast convergence. See the [“Restrictions for Bidirectional Forwarding Detection” section on page 41-2](#) for more information on BFD routing protocol support in Cisco IOS software.

Restrictions for Bidirectional Forwarding Detection

Restrictions include:

- BFD Hardware offloading of sessions is not supported with the use of authentication on the 4500 Sup7E and Sup8E. Enable BFD Echo mode to work with authentication.
- BFD works only for directly connected neighbors. BFD neighbors must be no more than one IP hop away. Multihop configurations are not supported.

Cisco IOS Release 15.1(1)SG

- Catalyst 4500 series switches support up to 128 BFD sessions with a minimum hello interval of 100 ms and a multiplier of 3. The multiplier specifies the minimum number of consecutive packets that can be missed before a session is declared down.
- If SSO is enabled on a dual RP system, the following limitations apply:
 - The minimum hello interval is 100 ms with a multiplier of 5 or higher.
 - Smaller values may be configured but may flap during an SSO switchover.
- To enable echo mode the peer system must be configured with the **no ip redirects** command.

Cisco IOS Release XE 3.5.0E and IOS 15.2(1)E

- Catalyst 4500 series switches support up to 100 BFD sessions with a minimum hello interval of 100 ms and a multiplier of 3. The multiplier specifies the minimum number of consecutive packets that can be missed before a session is declared down.
- If SSO is enabled on a dual RP system, the following limitations apply:
 - The minimum hello interval is 100 ms with a multiplier of 5 or higher.

- Smaller values may be configured but may flap during an SSO switchover.
- To enable echo mode the peer system must be configured with the **no ip redirects** command.

**Note**

For the most accurate platform and hardware restrictions, see the Cisco IOS software release notes for your software version.

Information About Bidirectional Forwarding Detection

- [BFD Operation, page 41-3](#)
- [Benefits of Using BFD for Failure Detection, page 41-7](#)

BFD Operation

BFD provides a low-overhead, short-duration method of detecting failures in the forwarding path between two adjacent switches, including the interfaces, data links, and forwarding planes.

BFD is a detection protocol that you enable at the interface and routing protocol levels. Cisco supports the BFD asynchronous mode, which depends on the sending of BFD control packets between two systems to activate and maintain BFD neighbor sessions between switches. Therefore, to create a BFD session, you must configure BFD on both systems (or BFD peers). Once BFD has been enabled on the interfaces and at the router level for the appropriate routing protocols, a BFD session is created, BFD timers are negotiated, and the BFD peers will begin to send BFD control packets to each other at the negotiated interval.

Cisco supports BFD echo mode. Echo packets are sent by the forwarding engine and are forwarded back along the same path to perform detection. The BFD session at the other end does not participate in the actual forwarding of the echo packets. See [Configuring BFD Echo Mode, page 41-15](#) for more information.

This section includes the following subsections:

- [Neighbor Relationships, page 41-3](#)
- [BFD Detection of Failures, page 41-4](#)
- [BFD Version Interoperability, page 41-5](#)
- [BFD Session Limits, page 41-5](#)
- [BFD Support for Nonbroadcast Media Interfaces, page 41-5](#)
- [BFD Support for Nonstop Forwarding with Stateful Switchover, page 41-5](#)
- [BFD Support for Stateful Switchover, page 41-6](#)
- [BFD Support for Static Routing, page 41-6](#)

Neighbor Relationships

BFD provides fast BFD peer failure detection times independently of all media types, encapsulations, topologies, and routing protocols BGP, EIGRP, OSPF, and static routes. By sending rapid failure detection notices to the routing protocols in the local switch to initiate the routing table recalculation process, BFD contributes to greatly reduced overall network convergence time. [Figure 41-1](#) shows a

simple network with two switches running OSPF and BFD. When OSPF discovers a neighbor (1) it sends a request to the local BFD process to initiate a BFD neighbor session with the OSPF neighbor routers (2). The BFD neighbor session with the OSPF neighbor router is established (3).

Figure 41-1 Establishing a BFD Neighbor Relationship

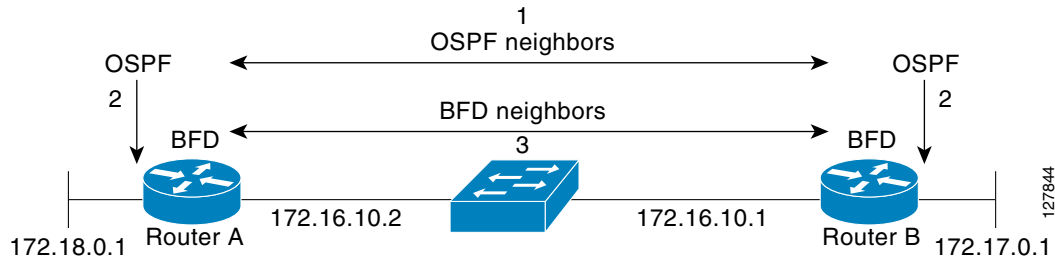
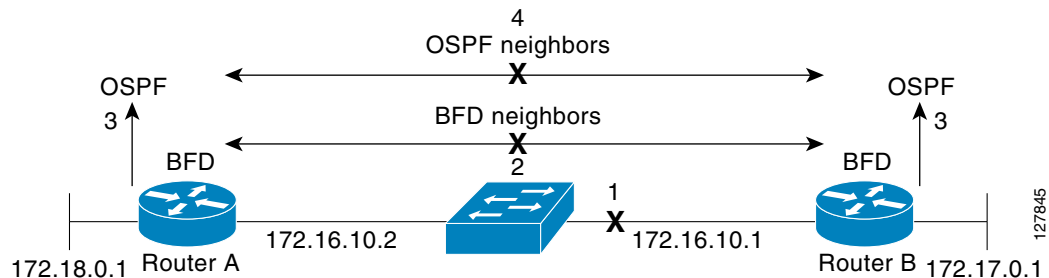


Figure 41-2 shows what happens when a failure occurs in the network (1). The BFD neighbor session with the OSPF neighbor router is torn down (2). BFD notifies the local OSPF process that the BFD neighbor is no longer reachable (3). The local OSPF process tears down the OSPF neighbor relationship (4). If an alternative path is available, the routers will immediately start converging on it.

Figure 41-2 Tearing Down an OSPF Neighbor Relationship



A routing protocol needs to register with BFD for every neighbor it acquires. Once a neighbor is registered, BFD initiates a session with the neighbor if a session does not already exist.

OSPF registers with BFD when:

- A neighbor finite state machine (FSM) transitions to full state.
- Both OSPF BFD and BFD are enabled.

On broadcast interfaces, OSPF establishes a BFD session only with the designated router (DR) and backup designated router (BDR), but not between any two switches (routers) in DROTHER state.

BFD Detection of Failures

Once a BFD session has been established and timer negotiations are complete, BFD peers send BFD control packets that act in the same manner as an IGP hello protocol to detect liveness, except at a more accelerated rate. The following information should be noted:

- BFD is a forwarding path failure detection protocol. BFD detects a failure, but the routing protocol must take action to bypass a failed peer.
- Typically, BFD can be used at any protocol layer. However, the Cisco implementation of BFD supports only Layer 3 clients, in particular, the BGP, EIGRP, and OSPF routing protocols, and static routing.
- Cisco devices will use one BFD session for multiple client protocols in the Cisco implementation of BFD. For example, if a network is running OSPF and EIGRP across the same link to the same peer, only one BFD session will be established, and BFD will share session information with both routing protocols. However, IPv4 and IPv6 clients cannot share a BFD session.

BFD Version Interoperability

Starting with Cisco IOS Release 15.1(1)SG, the Catalyst 4500 series switch supports BFD Version 1 as well as BFD Version 0. All BFD sessions come up as Version 1 by default and will be interoperable with Version 0. The system automatically performs BFD version detection, and BFD sessions between neighbors will run in the highest common BFD version between neighbors. For example, if one BFD neighbor is running BFD Version 0 and the other BFD neighbor is running Version 1, the session will run BFD Version 0. The output from the **show bfd neighbors [details]** command will verify which BFD version a BFD neighbor is running.

See the [“Example: Configuring BFD in an EIGRP Network with Echo Mode Enabled by Default” section on page 41-17](#) for an example of BFD version detection.

BFD Session Limits

The minimum number of BFD sessions that can be created varies with the “hello” interval. With “hello” intervals of 50ms, 128 sessions are permitted. More sessions are permitted at larger hello intervals.

BFD Support for Nonbroadcast Media Interfaces

Starting with Cisco IOS Release 15.1(1)SG, the BFD feature is supported on VLAN interfaces on the Catalyst 4500 series switch.

The **bfd interval** command must be configured on an interface to initiate BFD monitoring.

BFD Support for Nonstop Forwarding with Stateful Switchover

Typically, when a networking device restarts, all routing peers of that device detect that the device went down and then came back up. This transition results in a routing flap, which could spread across multiple routing domains. Routing flaps caused by routing restarts create routing instabilities, which are detrimental to the overall network performance. Nonstop forwarding (NSF) helps to suppress routing flaps in devices that are enabled with stateful switchover (SSO), thereby reducing network instability.

NSF allows for the forwarding of data packets to continue along known routes while the routing protocol information is being restored after a switchover. With NSF, peer networking devices do not experience routing flaps. Data traffic is forwarded while the standby RP assumes control from the failed active RP during a switchover. The ability of line cards and forwarding processors to remain up through a switchover and to remain current with the Forwarding Information Base (FIB) on the active RP is key to NSF operation.

In devices that support dual RPs, SSO establishes one of the RPs as the active processor; the other RP is designated as the standby processor, and then synchronizes information between them. A switchover from the active to the standby processor occurs when the active RP fails, when it is removed from the networking device, or when it is manually taken down for maintenance.

BFD Support for Stateful Switchover

The BFD protocol provides short-duration detection of failures in the path between adjacent forwarding engines. In network deployments that use dual RP switches (to provide redundancy), the switches have a graceful restart mechanism that protects the forwarding state during a switchover between the active RP and the standby RP.

Stateful BFD on the Standby RP

To ensure a successful switchover to the standby RP, the BFD protocol uses checkpoint messages to send session information from the active RP Cisco IOS instance to the standby RP Cisco IOS instance. The session information includes local and remote discriminators, adjacent router timer information, BFD setup information, and session-specific information such as the type of session and the session version. In addition, the BFD protocol sends session creation and deletion checkpoint messages to create or delete a session on the standby RP.

The BFD sessions on the standby RP do not receive or send packets and do not process expired timers. These sessions wait for a switchover to occur and then send packets for any active sessions so that sessions do not time out on adjacent switches.

When the BFD protocol on the standby RP is notified of a switchover it changes its state to active, registers itself with Cisco Express Forwarding so that it can receive packets, and then sends packets for any elements that have expired.

BFD also uses checkpoint messages to ensure that sessions created by clients on the active RP are maintained during a switchover. When a switchover occurs, BFD starts an SSO reclaim timer. Clients must reclaim their sessions within the duration specified by the reclaim timer or else the session is deleted.

Timer values are different based on the number of BFD sessions and the platform.

Table 41-1 describes the timer value on Cisco 4500 series switches.

Table 41-1 BFD Timer Values on a Cisco 4500 Series Switches

Maximum Number of BFD Sessions	BFD Session Type	Minimum Timer Value (ms)	Clients	Comments
100	Async/echo	100 multiplier 3	All	A multiple of 5 is recommended for SSO switches.

BFD Support for Static Routing

Unlike dynamic routing protocols, such as OSPF and BGP, static routing has no method of peer discovery. Therefore, when BFD is configured, the reachability of the gateway is completely dependent on the state of the BFD session to the specified neighbor. Unless the BFD session is up, the gateway for the static route is considered unreachable, and therefore the affected routes will not be installed in the appropriate Routing Information Base (RIB).

For a BFD session to establish successfully, BFD must be configured on the interface on the peer and there must be a BFD client registered on the peer for the address of the BFD neighbor. When an interface is used by dynamic routing protocols, the latter requirement is usually met by configuring the routing protocol instances on each neighbor for BFD. When an interface is used exclusively for static routing, this requirement must be met by configuring static routes on the peers.

BFD is supported on IPv4 and IPv6 static routes.

**Note**

If a BFD configuration is removed from the remote peer while the BFD session is in the up state, the updated state of the BFD session is not signaled to the static route. This will cause the static route to remain in the RIB. The only workaround is to remove the static route BFD neighbor configuration so that the static route no longer tracks BFD session state.

Benefits of Using BFD for Failure Detection

When you deploy any feature, it is important to consider all the alternatives and be aware of any trade-offs.

The closest alternative to BFD in conventional EIGRP, BGP, and OSPF deployments is the use of modified failure detection mechanisms for EIGRP, BGP, and OSPF routing protocols.

If you set EIGRP hello and hold timers to their absolute minimums, the failure detection rate for EIGRP falls to within a one- to two-second range.

If you use fast hellos for either BGP or OSPF, this Interior Gateway Protocol (IGP) protocol reduces its failure detection mechanism to a minimum of one second.

Advantages to implementing BFD over reduced timer mechanisms for routing protocols include the following:

- Although reducing the EIGRP, BGP, and OSPF timers can result in minimum detection timer of one to two seconds, BFD can provide failure detection in less than one second.
- Because BFD is not tied to any particular routing protocol, it can be used as a generic and consistent failure detection mechanism for EIGRP, BGP, and OSPF.
- Because some parts of BFD can be distributed to the data plane, it can be less CPU-intensive than the reduced EIGRP, BGP, and OSPF timers, which exist wholly at the control plane.

Hardware Support for BFD

The Catalyst 4500 supports a limited number of BFD sessions in hardware. Placing a session in BFD hardware is termed *hardware offload*. The advantage of hardware offload is that session keep-alive is handled entirely in hardware, placing no load on the CPU.

Not all BFD sessions can be offloaded to hardware. The requirements for offloaded sessions are:

- BFD version 1
- IPv4
- No echo mode

**Note**

BFD Echo is only supported in software and not hardware, because of a limitation on the Catalyst 4500 series switches.

**Note**

Hardware offload does not work on existing IPv4 static BFD sessions (with BFD Echo enabled). Remove the existing static BFD configuration and reconfigure the static BFD configuration with hardware offload enabled (no BFD Echo) to allow the session to be hosted in hardware.

The number of offloaded sessions varies by supervisor engine:

WS-X45-SUP6-E, WS-X45-SUP6L-E, support 64 sessions in hardware. Further sessions must be supported in software.

WS-X45-SUP7-E, WS-X45-SUP7L-E, WS-X45-SUP8-E, WS-X45-SUP8L-E, and WS-X45-SUP9-E support all 100 sessions in hardware.

The **show bfd neighbor detail** command displays print statistics for software and hardware (offloaded) sessions. Hardware sessions provide a limited set of statistics. In particular, statistics for packet transmit and receive intervals are not available for hardware sessions.

The **holddown** and **hello counts** are zero for all offloaded sessions.

**Note**

Hardware offload is not supported for IPv6 BFD sessions.

How to Configure Bidirectional Forwarding Detection

You start a BFD process by configuring BFD on the interface. When the BFD process is started, no entries are created in the adjacency database; in other words, no BFD control packets are sent or received. BFD echo mode, which is supported in BFD Version 1, starting with Cisco IOS Release 15.1(1)SG, is enabled by default.

BFD echo packets are sent and received, in addition to BFD control packets. The adjacency creation takes place once you have configured BFD support for the applicable routing protocols. This section contains the following procedures:

- [Configuring BFD Session Parameters on the Interface, page 41-8](#) (required)
- [Configuring BFD Support for Dynamic Routing Protocols, page 41-9](#) (required)
- [Configuring BFD Support for Static Routing, page 41-14](#) (optional)
- [Configuring BFD Echo Mode, page 41-15](#) (optional)
- [Monitoring and Troubleshooting BFD, page 41-17](#) (optional)

Configuring BFD Session Parameters on the Interface

The steps in this procedure show how to configure BFD on the interface by setting the baseline BFD session parameters on an interface. Repeat the steps in this procedure for each interface over which you want to run BFD sessions to BFD neighbors.

To configure BFD session parameters, perform this task:

	Command or Action	Purpose
Step 1	enable Switch> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Switch# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Switch(config)# interface gigabitethernet 6/1	Enters interface configuration mode.
Step 4	bfd interval <i>milliseconds</i> min_rx <i>milliseconds</i> multiplier <i>interval-multiplier</i> Switch(config-if)# bfd interval 100 min_rx 100 multiplier 3 Switch(config-if)# no bfd echo	Enables BFD on the interface. Disables BFD echo mode to enable Hardware Off-load.
Step 5	end Switch(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring BFD Support for Dynamic Routing Protocols

You can enable BFD support for dynamic routing protocols at the router level to enable BFD support globally for all interfaces or you can configure BFD on a per-interface basis at the interface level.

This section describes the following procedures:

- [Configuring BFD Support for BGP, page 41-9](#) (optional)
- [Configuring BFD Support for EIGRP, page 41-10](#) (optional)
- [Configuring BFD Support for OSPF, page 41-11](#) (optional)

Configuring BFD Support for BGP

This section describes the procedure for configuring BFD support for BGP so that BGP is a registered protocol with BFD and will receive forwarding path detection failure messages from BFD.

Prerequisites

BGP must be running on all participating switches.

The baseline parameters for BFD sessions on the interfaces over which you want to run BFD sessions to BFD neighbors must be configured. See the [“Configuring BFD Session Parameters on the Interface” section on page 41-8](#) for more information.

To configure BFD support for BGP, perform this task:

	Command or Action	Purpose
Step 1	enable Switch> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Switch# configure terminal	Enters global configuration mode.
Step 3	router bgp as-tag Switch(config)# router bgp tag1	Specifies a BGP process and enters router configuration mode.
Step 4	neighbor ip-address fall-over bfd Switch(config-router)# neighbor 172.16.10.2 fall-over bfd	Enables BFD support for fallover.
Step 5	end Switch(config-router)# end	Exits router configuration mode and returns the switch to privileged EXEC mode.
Step 6	show bfd neighbors [details] Switch# show bfd neighbors detail	(Optional) Verifies that the BFD neighbor is active and displays the routing protocols that BFD has registered.
Step 7	show ip bgp neighbor Switch# show ip bgp neighbor	(Optional) Displays information about BGP and TCP connections to neighbors.

What to Do Next

See the “[Monitoring and Troubleshooting BFD](#)” section on page 41-17 for more information on monitoring and troubleshooting BFD. If you want to configure BFD support for another routing protocol, see the following sections:

- [Configuring BFD Support for EIGRP, page 41-10](#)
- [Configuring BFD Support for OSPF, page 41-11](#)

Configuring BFD Support for EIGRP

This section describes the procedure for configuring BFD support for EIGRP so that EIGRP is a registered protocol with BFD and will receive forwarding path detection failure messages from BFD. There are two methods for enabling BFD support for EIGRP:

- You can enable BFD for all of the interfaces for which EIGRP is routing by using the **bfd all-interfaces** command in router configuration mode.
- You can enable BFD for a subset of the interfaces for which EIGRP is routing by using the **bfd interface type number** command in router configuration mode.

Prerequisites

EIGRP must be running on all participating switches.

The baseline parameters for BFD sessions on the interfaces over which you want to run BFD sessions to BFD neighbors must be configured. See the “[Configuring BFD Session Parameters on the Interface](#)” section on page 41-8 for more information.

To configure BFD support for EIGRP, perform this task:

	Command or Action	Purpose
Step 1	enable Switch> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Switch# configure terminal	Enters global configuration mode.
Step 3	Switch eigrp as-number Switch(config)# router eigrp 123	Configures the EIGRP routing process and enters router configuration mode.
Step 4	bfd all-interfaces or bfd interface type number Switch(config-router)# bfd all-interfaces or Switch(config-router)# bfd interface gigabitethernet 6/1	Enables BFD globally on all interfaces associated with the EIGRP routing process. or Enables BFD on a per-interface basis for one or more interfaces associated with the EIGRP routing process.
Step 5	end Switch(config-router) end	Exits router configuration mode and returns the switch to privileged EXEC mode.
Step 6	show bfd neighbors [details] Switch# show bfd neighbors details	(Optional) Verifies that the BFD neighbor is active and displays the routing protocols that BFD has registered.
Step 7	show ip eigrp interfaces [type number] [as-number] [detail] Switch# show ip eigrp interfaces detail	(Optional) Displays the interfaces for which BFD support for EIGRP has been enabled.

What to Do Next

See the “[Monitoring and Troubleshooting BFD](#)” section on page 41-17 for more information on monitoring and troubleshooting BFD. If you want to configure BFD support for another routing protocol, see the following sections:

- [Configuring BFD Support for OSPF](#), page 41-11

Configuring BFD Support for OSPF

This section describes the procedures for configuring BFD support for OSPF so that OSPF is a registered protocol with BFD and will receive forwarding path detection failure messages from BFD. You can either configure BFD support for OSPF globally on all interfaces or configure it selectively on one or more interfaces.

There are two methods for enabling BFD support for OSPF:

- You can enable BFD on all the interfaces for which OSPF is routing by using the **bfd all-interfaces** command in router configuration mode. You can disable BFD support on individual interfaces using the **ip ospf bfd [disable]** command in interface configuration mode.
- You can enable BFD on a subset of the interfaces for which OSPF is routing by using the **ip ospf bfd** command in interface configuration mode.

See the following sections for tasks for configuring BFD support for OSPF:

- [Configuring BFD Support for OSPF for All Interfaces, page 41-12](#) (optional)
- [Configuring BFD Support for OSPF for One or More Interfaces, page 41-13](#) (optional)

Configuring BFD Support for OSPF for All Interfaces

To configure BFD for all OSPF interfaces, perform the steps in this section.

If you do not want to configure BFD on all OSPF interfaces and would rather configure BFD support specifically for one or more interfaces, see the [“Configuring BFD Support for OSPF for One or More Interfaces”](#) section on page 41-13.

Prerequisites

OSPF must be running on all participating switches.

The baseline parameters for BFD sessions on the interfaces over which you want to run BFD sessions to BFD neighbors must be configured. See the [“Configuring BFD Session Parameters on the Interface”](#) section on page 41-8 for more information.

To configure BFD support for OSPF for all interfaces:

	Command or Action	Purpose
Step 1	enable Switch> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Switch# configure terminal	Enters global configuration mode.
Step 3	Switch ospf process-id Switch(config)# router ospf 4	Specifies an OSPF process and enters router configuration mode.
Step 4	bfd all-interfaces Switch(config-router)# bfd all-interfaces	Enables BFD globally on all interfaces associated with the OSPF routing process.
Step 5	end Switch(config-if)# end	Exits interface configuration mode and returns the switch to privileged EXEC mode.
Step 6	show bfd neighbors [details] Switch# show bfd neighbors detail	(Optional) Displays information that can help verify if the BFD neighbor is active and displays the routing protocols that BFD has registered.
Step 7	show ip ospf Switch# show ip ospf	(Optional) Displays information that can help verify if BFD for OSPF has been enabled.

What to Do Next

See the [“Monitoring and Troubleshooting BFD”](#) section on page 41-17 for more information on monitoring and troubleshooting BFD. If you want to configure BFD support for another routing protocol, see the following sections:

- [Configuring BFD Support for BGP, page 41-9](#)
- [Configuring BFD Support for EIGRP, page 41-10](#)

Configuring BFD Support for OSPF for One or More Interfaces

To configure BFD on one or more OSPF interfaces, perform the steps in this section.

Prerequisites

OSPF must be running on all participating switches.

The baseline parameters for BFD sessions on the interfaces over which you want to run BFD sessions to BFD neighbors must be configured. See the [“Configuring BFD Session Parameters on the Interface” section on page 41-8](#) for more information.

To configure BFD supporter for OSPF for one or more interfaces, perform this task:

	Command or Action	Purpose
Step 1	enable Switch> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	configure terminal Switch# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Switch(config)# interface gigabitethernet 6/1	Enters interface configuration mode.
Step 4	ip ospf bfd [disable] Switch(config-if)# ip ospf bfd	Enables or disables BFD on a per-interface basis for one or more interfaces associated with the OSPF routing process. Note You should use the disable keyword only if you enabled BFD on all of the interfaces that OSPF is associated with using the bfd all-interfaces command in switch configuration mode.
Step 5	end Switch(config-if)# end	Exits interface configuration mode and returns the switch to privileged EXEC mode.
Step 6	show bfd neighbors [details] Switch# show bfd neighbors details	(Optional) Displays information that can help verify if the BFD neighbor is active and displays the routing protocols that BFD has registered. Note If hardware-offloaded BFD sessions are configured with Tx and Rx intervals that are not multiples of 50 ms, the hardware intervals are changed. However, output from the show bfd neighbors details command displays the configured intervals, not the changed ones.
Step 7	show ip ospf Switch# show ip ospf	(Optional) Displays information that can help verify if BFD support for OSPF has been enabled.

What to Do Next

See the [“Monitoring and Troubleshooting BFD” section on page 41-17](#) for more information on monitoring and troubleshooting BFD. If you want to configure BFD support for another routing protocol, see the following sections:

- [Configuring BFD Support for BGP, page 41-9](#)

- [Configuring BFD Support for EIGRP, page 41-10](#)

Configuring BFD Support for Static Routing

Perform this task to configure BFD support for static routing. Repeat the steps in this procedure on each BFD neighbor. For more information, see the [“Example: Configuring BFD Support for Static Routing” section on page 41-27](#).

To configure BFD support for static routing, perform this task:

	Command or Action	Purpose
Step 1	enable Switch> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Switch# configure terminal	Enters global configuration mode.
Step 3	interface type number Switch(config)# interface gigabitethernet6/1	Configures an interface and enters interface configuration mode.
Step 4	no switchport Switch(config-if)# no switchport	Changes the interface to Layer 3.
Step 5	ip address ip-address mask Switch(config-if)# ip address 10.201.201.1 255.255.255.0	Configures an IP address for the interface.
Step 6	bfd interval milliseconds min_rx milliseconds multiplier interval-multiplier Switch(config-if)# bfd interval 500 min_rx 500 multiplier 5	Enables BFD on the interface.
Step 7	exit Switch(config-if)# exit	Exits interface configuration mode and returns to global configuration mode.
Step 8	ip route static bfd interface-type interface-number ip-address [group group-name [passive]] Switch(config)# ip route static bfd Gi6/1 10.1.1.1 group group1 passive	Specifies a static route BFD neighbor. <ul style="list-style-type: none"> • The <i>interface-type</i>, <i>interface-number</i>, and <i>ip-address</i> arguments are required because BFD support exists only for directly connected neighbors.
Step 9	ip route [vrf vrf-name] prefix mask {ip-address interface-type interface-number [ip-address]} [dhcp] [distance] [name next-hop-name] [permanent track number] [tag tag] Example: Switch(config)# ip route 10.0.0.0 255.0.0.0 Gi6/1 10.201.201.2	Specifies a static route BFD neighbor.
Step 10	exit Example: Switch(config)# exit	Exits global configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
Step 11	<code>show ip static route</code> Example: Switch# <code>show ip static route</code>	(Optional) Displays static route database information.
Step 12	<code>show ip static route bfd</code> Example: Switch# <code>show ip static route bfd</code>	(Optional) Displays information about the static BFD configuration from the configured BFD groups and nongroup entries.
Step 13	<code>exit</code> Example: Switch# <code>exit</code>	Exits privileged EXEC mode and returns to user EXEC mode.

Configuring BFD Echo Mode

BFD echo mode is enabled by default, but you can disable it such that it can run independently in each direction.

BFD echo mode works with asynchronous BFD. Echo packets are sent by the forwarding engine and forwarded back along the same path in order to perform detection—the BFD session at the other end does not participate in the actual forwarding of the echo packets. The echo function and the forwarding engine are responsible for the detection process; therefore, the number of BFD control packets that are sent out between two BFD neighbors is reduced. In addition, because the forwarding engine is testing the forwarding path on the remote (neighbor) system without involving the remote system, there is an opportunity to improve the interpacket delay variance, thereby achieving quicker failure detection times than when using BFD Version 0 with BFD control packets for the BFD session.

Echo mode is described as without asymmetry when it is running on both sides (both BFD neighbors are running echo mode).

Prerequisites

BFD must be running on all participating switches.

Before using BFD echo mode, you must disable the sending of Internet Control Message Protocol (ICMP) redirect messages by entering the **no ip redirects** command, to avoid high CPU utilization.

The baseline parameters for BFD sessions on the interfaces over which you want to run BFD sessions to BFD neighbors must be configured. See the [“Configuring BFD Session Parameters on the Interface” section on page 41-8](#) for more information.

Restrictions

BFD echo mode which is supported in BFD Version 1.

This section contains the following configuration tasks for BFD echo mode:

- [Configuring the BFD Slow Timer, page 41-16](#)
- [Disabling BFD Echo Mode Without Asymmetry, page 41-16](#)

**Note**

BFD echo mode does not work in conjunction with Unicast Reverse Path Forwarding (uRPF) configuration. If BFD echo mode and uRPF configurations are enabled, then the sessions will flap.

Configuring the BFD Slow Timer

The steps in this procedure show how to change the value of the BFD slow timer. Repeat the steps in this procedure for each BFD switch.

To configure the BFD slow timer, perform this task:

	Command or Action	Purpose
Step 1	enable Switch> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Switch# configure terminal	Enters global configuration mode.
Step 3	bfd slow-timer <i>milliseconds</i> Switch(config)# bfd slow-timer 12000	Configures the BFD slow timer.
Step 4	end Switch(config)# end	Exits global configuration mode and returns the switch to privileged EXEC mode.

Disabling BFD Echo Mode Without Asymmetry

The steps in this procedure show how to disable BFD echo mode without asymmetry —no echo packets are sent by the switch, and the switch does not forward BFD echo packets that are received from neighboring switches.

Repeat the steps in this procedure for each BFD switch.

To disable BFD echo mode without asymmetry, perform this task:

	Command or Action	Purpose
Step 1	enable Example: Switch> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Switch# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Switch(config)# interface GigabitEthernet 6/1	Enters interface configuration mode.

	Command or Action	Purpose
Step 4	<code>no bfd echo</code> Example: <code>Switch(config-if)# no bfd echo</code>	Disables BFD echo mode.
Step 5	<code>end</code> Example: <code>Switch(config-if)# end</code>	Exits global configuration mode and returns the switch to global configuration mode.

Monitoring and Troubleshooting BFD

This section describes how to retrieve BFD information for maintenance and troubleshooting. The commands in these tasks can be entered as needed, in any order.

For more information about BFD session initiation and failure, refer to the [“BFD Operation” section on page 41-3](#).

To monitor and troubleshoot BFD, perform the following steps:

	Command or Action	Purpose
Step 1	<code>enable</code> <code>Switch> enable</code>	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	<code>show bfd neighbors [details]</code> <code>Switch# show bfd neighbors details</code>	(Optional) Displays the BFD adjacency database. <ul style="list-style-type: none"> The details keyword shows all BFD protocol parameters and timers per neighbor.
Step 3	<code>debug bfd [packet event]</code> <code>Switch# debug bfd packet</code>	(Optional) Displays debugging information about BFD packets.

Configuration Examples for Bidirectional Forwarding Detection

This section provides the following configuration examples:

- [Example: Configuring BFD in an EIGRP Network with Echo Mode Enabled by Default, page 41-17](#)
- [Example: Configuring BFD in an OSPF Network, page 41-22](#)
- [Example: Configuring BFD Hardware-Offload support in a BGP Network Network, page 41-25](#)
- [Example: Configuring BFD Support for Static Routing, page 41-27](#)

Example: Configuring BFD in an EIGRP Network with Echo Mode Enabled by Default

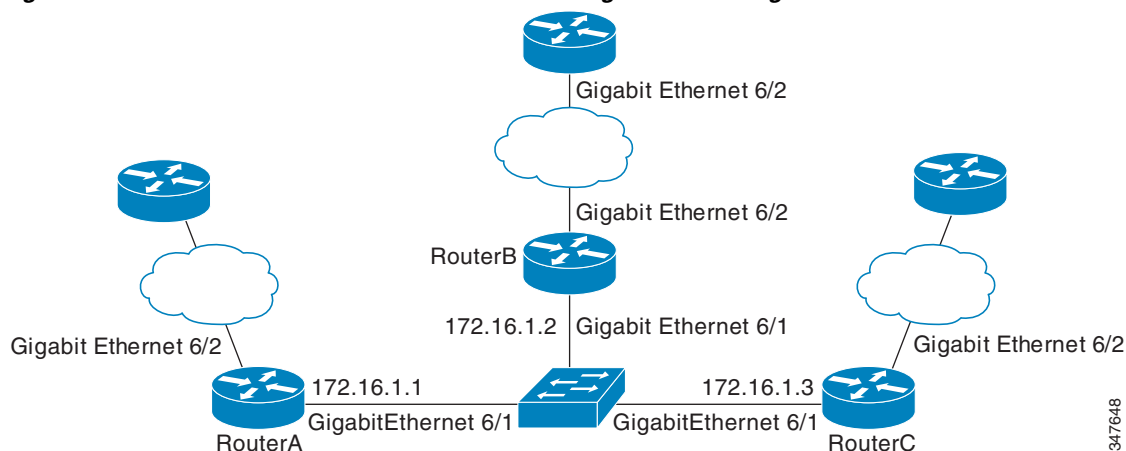
The following example shows how to configure BFD in an EIGRP network with echo mode enabled by default.

In this example, the EIGRP network contains SwitchA, SwitchB, and SwitchC. Gigabit Ethernet interface 6/1 on SwitchA is connected to the same network as Gigabit Ethernet interface 6/1 on SwitchB. Gigabit Ethernet interface 6/1 on SwitchB is connected to the same network as Gigabit Ethernet interface 6/1 on SwitchC.

SwitchA and SwitchB are running BFD Version 1, which supports echo mode, and SwitchC is running BFD Version 0, which does not support echo mode. We would say that the BFD sessions between SwitchC and its BFD neighbors are running echo mode with asymmetry. This is because echo mode will run on the forwarding path for RouterA and SwitchB, and their echo packets will return along the same path for BFD sessions and failure detections, while their BFD neighbor SwitchC runs BFD Version 0 and uses BFD control packets for BFD sessions and failure detections.

Figure 41-3 shows a large EIGRP network with several switches, three of which are BFD neighbors that are running EIGRP as their routing protocol.

Figure 41-3 EIGRP Network with Three BFD Neighbors Running V1 or V0



The example, starting in global configuration mode, shows the configuration of BFD.

Configuration for SwitchA

```
interface GigabitEthernet6/2
  no switch
  ip address 10.4.9.14 255.255.255.0
!
interface GigabitEthernet6/1
  no switchport
  ip address 172.16.1.1 255.255.255.0
  bfd interval 100 min_rx 50 multiplier 3
  no shutdown
!
router eigrp 11
  network 172.16.0.0
  bfd all-interfaces
  auto-summary
!
ip default-gateway 10.4.9.1
ip default-network 0.0.0.0
ip route 0.0.0.0 0.0.0.0 10.4.9.1
ip route 172.16.1.129 255.255.255.255 10.4.9.1
!
```

Configuration for SwitchB

```
!
```



```

interface GigabitEthernet6/2
  no switchport
  ip address 10.4.9.34 255.255.255.0
!
interface GigabitEthernet6/1
  no switchport
  ip address 172.16.1.2 255.255.255.0
  bfd interval 100 min_rx 50 multiplier 3
!
router eigrp 11
  network 172.16.0.0
  bfd all-interfaces
  auto-summary
!
ip default-gateway 10.4.9.1
ip default-network 0.0.0.0
ip route 0.0.0.0 0.0.0.0 10.4.9.1
ip route 172.16.1.129 255.255.255.255 10.4.9.1
!

```

Configuration for SwitchC

```

!
!
interface GigabitEthernet6/2
  no switchport
  no shutdown
  ip address 10.4.9.34 255.255.255.0
!
interface GigabitEthernet6/1
  no switchport
  ip address 172.16.1.3 255.255.255.0
  bfd interval 100 min_rx 50 multiplier 3
  no shutdown
!
router eigrp 11
  network 172.16.0.0
  bfd all-interfaces
  auto-summary
!
ip default-gateway 10.4.9.1
ip default-network 0.0.0.0
ip route 0.0.0.0 0.0.0.0 10.4.9.1
ip route 172.16.1.129 255.255.255.255 10.4.9.1
!
!
end

```

The output from the **show bfd neighbors details** command from SwitchA verifies that BFD sessions have been created among all three switches and that EIGRP is registered for BFD support. The first group of output shows that SwitchC with the IP address 172.16.1.3 runs BFD Version 0 and therefore does not use the echo mode. The second group of output shows that SwitchB with the IP address 172.16.1.2 does run BFD Version 1, and the 50 millisecond BFD interval parameter had been adopted.

The relevant command output is shown in bold.

SwitchA

SwitchA# **show bfd neighbors details**

```

OurAddr      NeighAddr    LD/RD  RH/RS    Holdown(mult)  State    Int
172.16.1.1    172.16.1.3    5/3    1(RH)    150 (3 )       Up      Gi6/1
Session state is UP and not using echo function.
Local Diag: 0, Demand mode: 0, Poll bit: 0

```

```

MinTxInt: 50000, MinRxInt: 50000, Multiplier: 3
Received MinRxInt: 50000, Received Multiplier: 3
Holdown (hits): 150(0), Hello (hits): 50(1364284)
Rx Count: 1351813, Rx Interval (ms) min/max/avg: 28/64/49 last: 4 ms ago
Tx Count: 1364289, Tx Interval (ms) min/max/avg: 40/68/49 last: 32 ms ago
Registered protocols: EIGRP
Uptime: 18:42:45
Last packet: Version: 0           - Diagnostic: 0
                                I Hear You bit: 1   - Demand bit: 0
                                Poll bit: 0         - Final bit: 0
                                Multiplier: 3        - Length: 24
                                My Discr.: 3         - Your Discr.: 5
                                Min tx interval: 50000 - Min rx interval: 50000
                                Min Echo interval: 0

OurAddr      NeighAddr      LD/RD  RH/RS  Holdown(mult)  State  Int
172.16.1.1    172.16.1.2      6/1    Up      0      (3 )  Up      Gi6/1
Session state is UP and using echo function with 50 ms interval.
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 1000000, MinRxInt: 1000000, Multiplier: 3
Received MinRxInt: 1000000, Received Multiplier: 3
Holdown (hits): 3000(0), Hello (hits): 1000(317)
Rx Count: 305, Rx Interval (ms) min/max/avg: 1/1016/887 last: 448 ms ago
Tx Count: 319, Tx Interval (ms) min/max/avg: 1/1008/880 last: 532 ms ago
Registered protocols: EIGRP
Uptime: 00:04:30
Last packet: Version: 1           - Diagnostic: 0
                                State bit: Up       - Demand bit: 0
                                Poll bit: 0         - Final bit: 0
                                Multiplier: 3        - Length: 24
                                My Discr.: 1         - Your Discr.: 6
                                Min tx interval: 1000000 - Min rx interval: 1000000
                                Min Echo interval: 50000

```

The output from the **show bfd neighbors details** command on SwitchB verifies that BFD sessions have been created and that EIGRP is registered for BFD support. As previously noted, SwitchA runs BFD Version 1, therefore echo mode is running, and SwitchC runs BFD Version 0, so echo mode does not run. The relevant command output is shown in bold.

SwitchB

SwitchB# **show bfd neighbors details**

```

OurAddr      NeighAddr      LD/RD  RH/RS  Holdown(mult)  State  Int
172.16.1.2    172.16.1.1      1/6    Up      0      (3 )  Up      Gi6/1
Session state is UP and using echo function with 50 ms interval.
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 1000000, MinRxInt: 1000000, Multiplier: 3
Received MinRxInt: 1000000, Received Multiplier: 3
Holdown (hits): 3000(0), Hello (hits): 1000(337)
Rx Count: 341, Rx Interval (ms) min/max/avg: 1/1008/882 last: 364 ms ago
Tx Count: 339, Tx Interval (ms) min/max/avg: 1/1016/886 last: 632 ms ago
Registered protocols: EIGRP
Uptime: 00:05:00
Last packet: Version: 1           - Diagnostic: 0
                                State bit: Up       - Demand bit: 0
                                Poll bit: 0         - Final bit: 0
                                Multiplier: 3        - Length: 24
                                My Discr.: 6         - Your Discr.: 1
                                Min tx interval: 1000000 - Min rx interval: 1000000
                                Min Echo interval: 50000

OurAddr      NeighAddr      LD/RD  RH/RS  Holdown(mult)  State  Int

```

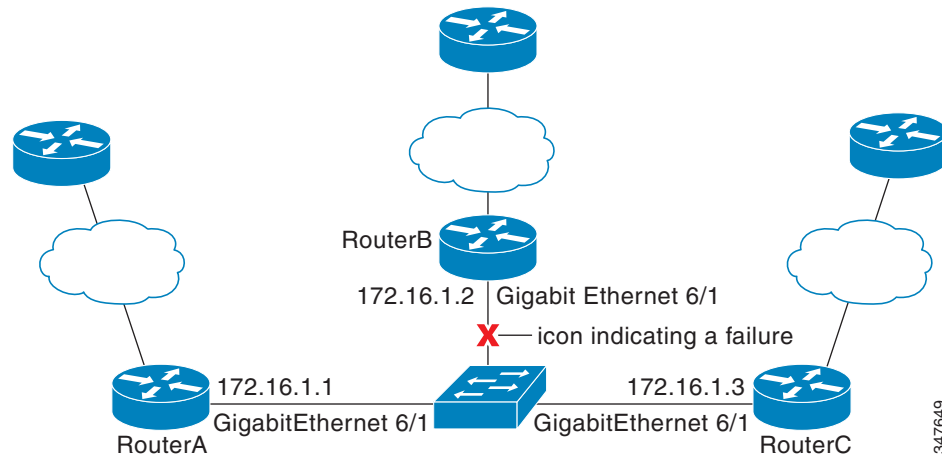
```

172.16.1.2 172.16.1.3 3/6 1(RH) 118 (3 ) Up Gi6/1
Session state is UP and not using echo function.
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 50000, MinRxInt: 50000, Multiplier: 3
Received MinRxInt: 50000, Received Multiplier: 3
Holdown (hits): 150(0), Hello (hits): 50(5735)
Rx Count: 5731, Rx Interval (ms) min/max/avg: 32/72/49 last: 32 ms ago
Tx Count: 5740, Tx Interval (ms) min/max/avg: 40/64/50 last: 44 ms ago
Registered protocols: EIGRP
Uptime: 00:04:45
Last packet: Version: 0 - Diagnostic: 0
I Hear You bit: 1 - Demand bit: 0
Poll bit: 0 - Final bit: 0
Multiplier: 3 - Length: 24
My Discr.: 6 - Your Discr.: 3
Min tx interval: 50000 - Min rx interval: 50000
Min Echo interval: 0

```

Figure 41-4 shows that Gigabit Ethernet interface 6/1 on SwitchB has failed. When Gigabit Ethernet interface 6/1 on SwitchB is shut down, the BFD values of the corresponding BFD sessions on SwitchA and SwitchB are reduced.

Figure 41-4 Gigabit Ethernet Interface 6/1 Failure



When Gigabit Ethernet interface 6/1 on SwitchB fails, BFD will no longer detect SwitchB as a BFD neighbor for SwitchA or for SwitchC. In this example, Gigabit Ethernet interface 6/1 has been administratively shut down on SwitchB.

The following output from the **show bfd neighbors** command on SwitchA now shows only one BFD neighbor for SwitchA in the EIGRP network. The relevant command output is shown in bold.

SwitchA# **show bfd neighbors**

OurAddr	NeighAddr	LD/RD	RH/RS	Holdown(mult)	State	Int
172.16.1.1	172.16.1.3	5/3	1(RH)	134 (3)	Up	Gi6/1

The following output from the **show bfd neighbors** command on SwitchC also now shows only one BFD neighbor for SwitchC in the EIGRP network. The relevant command output is shown in bold.

SwitchC# **show bfd neighbors**

OurAddr	NeighAddr	LD/RD	RH	Holdown(mult)	State	Int
172.16.1.3	172.16.1.1	3/5	1	114 (3)	Up	Gi6/1

Example: Configuring BFD in an OSPF Network

The following example shows how to configure BFD in an OSPF network.

In this example, the “simple” OSPF network consists of SwitchA and SwitchB. Gigabit Ethernet interface 6/1 on SwitchA is connected to the same network as Gigabit Ethernet interface 6/1 in SwitchB. The example, starting in global configuration mode, shows the configuration of BFD. For both SwitchA and SwitchB, BFD is configured globally for all interfaces associated with the OSPF process.

Configuration for SwitchA

```
!
interface GigabitEthernet 6/1
 no switchport
 ip address 172.16.10.1 255.255.255.0
 bfd interval 100 min_rx 100 multiplier 3
!
interface GigabitEthernet 6/2
 no switchport
 ip address 172.17.0.1 255.255.255.0
!
router ospf 123
 log-adjacency-changes detail
 network 172.16.0.0 0.0.0.255 area 0
 network 172.17.0.0 0.0.0.255 area 0
 bfd all-interfaces
```

Configuration for SwitchB

```
!
interface GigabitEthernet 6/1
 no switchport
 ip address 172.16.10.2 255.255.255.0
 bfd interval 100 min_rx 100 multiplier 3
!
interface GigabitEthernet 6/2
 no switchport
 ip address 172.18.0.1 255.255.255.0
!
router ospf 123
 log-adjacency-changes detail
 network 172.16.0.0 0.0.255.255 area 0
 network 172.18.0.0 0.0.255.255 area 0
 bfd all-interfaces
```

The output from the **show bfd neighbors details** command verifies that a BFD session has been created and that OSPF is registered for BFD support. The relevant command output is shown in bold.

SwitchA

SwitchA# **show bfd neighbors details**

```
OurAddr      NeighAddr    LD/RD RH  Holdown(mult)  State    Int
172.16.10.1  172.16.10.2  1/2  1    532 (3 )      Up       Gi6/1
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 200000, MinRxInt: 200000, Multiplier: 5
Received MinRxInt: 1000, Received Multiplier: 3
Holdown (hits): 600(22), Hello (hits): 200(84453)
Rx Count: 49824, Rx Interval (ms) min/max/avg: 208/440/332 last: 68 ms ago
Tx Count: 84488, Tx Interval (ms) min/max/avg: 152/248/196 last: 192 ms ago
Registered protocols: OSPF
Uptime: 02:18:49
Last packet: Version: 0           - Diagnostic: 0
```

```

I Hear You bit: 1      - Demand bit: 0
Poll bit: 0           - Final bit: 0
Multiplier: 3         - Length: 24
My Discr.: 2          - Your Discr.: 1
Min tx interval: 50000 - Min rx interval: 1000
Min Echo interval: 0

```

The output from the **show bfd neighbors details** command on SwitchB verifies that a BFD session has been created:

SwitchB

```
SwitchB# attach 6
```

```
Switch> show bfd neighbors details
```

```
Cleanup timer hits: 0
```

```

OurAddr      NeighAddr    LD/RD RH  Holdown(mult)  State    Int
172.16.10.2  172.16.10.1    8/1  1    1000 (5 )    Up      Gi6/1
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 50000, MinRxInt: 1000, Multiplier: 3
Received MinRxInt: 200000, Received Multiplier: 5
Holdown (hits): 1000(0), Hello (hits): 200(5995)
Rx Count: 10126, Rx Interval (ms) min/max/avg: 152/248/196 last: 0 ms ago
Tx Count: 5998, Tx Interval (ms) min/max/avg: 204/440/332 last: 12 ms ago
Last packet: Version: 0          - Diagnostic: 0
                I Hear You bit: 1      - Demand bit: 0
                Poll bit: 0           - Final bit: 0
                Multiplier: 5         - Length: 24
                My Discr.: 1          - Your Discr.: 8
                Min tx interval: 200000 - Min rx interval: 200000
                Min Echo interval: 0
Uptime: 00:33:13
SSO Cleanup Timer called: 0
SSO Cleanup Action Taken: 0
Pseudo pre-emptive process count: 239103 min/max/avg: 8/16/8 last: 0 ms ago
  IPC Tx Failure Count: 0
  IPC Rx Failure Count: 0
  Total Adjs Found: 1

```

The output of the **show ip ospf** command verifies that BFD has been enabled for OSPF. The relevant command output is shown in bold.

SwitchA

```
SwitchA# show ip ospf
```

```

Routing Process "ospf 123" with ID 172.16.10.1
Supports only single TOS(TOS0) routes
Supports opaque LSA
Supports Link-local Signaling (LLS)
Initial SPF schedule delay 5000 msecs
Minimum hold time between two consecutive SPFs 10000 msecs
Maximum wait time between two consecutive SPFs 10000 msecs
Incremental-SPF disabled
Minimum LSA interval 5 secs
Minimum LSA arrival 1000 msecs
LSA group pacing timer 240 secs
Interface flood pacing timer 33 msecs
Retransmission pacing timer 66 msecs
Number of external LSA 0. Checksum Sum 0x000000
Number of opaque AS LSA 0. Checksum Sum 0x000000

```

```

Number of DCbitless external and opaque AS LSA 0
Number of DoNotAge external and opaque AS LSA 0
Number of areas in this router is 1. 1 normal 0 stub 0 nssa
External flood list length 0
BFD is enabled
  Area BACKBONE(0)
    Number of interfaces in this area is 2 (1 loopback)
    Area has no authentication
    SPF algorithm last executed 00:00:08.828 ago
    SPF algorithm executed 9 times
    Area ranges are
    Number of LSA 3. Checksum Sum 0x028417
    Number of opaque link LSA 0. Checksum Sum 0x000000
    Number of DCbitless LSA 0
    Number of indication LSA 0
    Number of DoNotAge LSA 0
    Flood list length 0

```

SwitchB

SwitchB# **show ip ospf**

```

Routing Process "ospf 123" with ID 172.18.0.1
Supports only single TOS(TOS0) routes
Supports opaque LSA
Supports Link-local Signaling (LLS)
Supports area transit capability
Initial SPF schedule delay 5000 msec
Minimum hold time between two consecutive SPF's 10000 msec
Maximum wait time between two consecutive SPF's 10000 msec
Incremental-SPF disabled
Minimum LSA interval 5 secs
Minimum LSA arrival 1000 msec
LSA group pacing timer 240 secs
Interface flood pacing timer 33 msec
Retransmission pacing timer 66 msec
Number of external LSA 0. Checksum Sum 0x0
Number of opaque AS LSA 0. Checksum Sum 0x0
Number of DCbitless external and opaque AS LSA 0
Number of DoNotAge external and opaque AS LSA 0
Number of areas in this router is 1. 1 normal 0 stub 0 nssa
Number of areas transit capable is 0
External flood list length 0
BFD is enabled
  Area BACKBONE(0)
    Number of interfaces in this area is 2 (1 loopback)
    Area has no authentication
    SPF algorithm last executed 02:07:30.932 ago
    SPF algorithm executed 7 times
    Area ranges are
    Number of LSA 3. Checksum Sum 0x028417
    Number of opaque link LSA 0. Checksum Sum 0x0
    Number of DCbitless LSA 0
    Number of indication LSA 0
    Number of DoNotAge LSA 0
    Flood list length 0

```

The output of the **show ip ospf interface** command verifies that BFD has been enabled for OSPF on the interfaces connecting SwitchA and SwitchB. The relevant command output is shown in bold.

SwitchA

SwitchA# **show ip ospf interface gigabitethernet 6/1**

```

show ip ospf interface gigabitethernet 6/1
Gigabitethernet 6/1 is up, line protocol is up
  Internet Address 172.16.10.1/24, Area 0
  Process ID 123, Router ID 172.16.10.1, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State BDR, Priority 1, BFD enabled
  Designated Router (ID) 172.18.0.1, Interface address 172.16.10.2
  Backup Designated router (ID) 172.16.10.1, Interface address 172.16.10.1
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:03
  Supports Link-local Signaling (LLS)
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 172.18.0.1 (Designated Router)
  Suppress hello for 0 neighbor(s)

```

SwitchB

```
SwitchB# show ip ospf interface gigabitethernet 6/1
```

```

Gigabitethernet 6/1 is up, line protocol is up
  Internet Address 172.18.0.1/24, Area 0
  Process ID 123, Router ID 172.18.0.1, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State DR, Priority 1, BFD enabled
  Designated Router (ID) 172.18.0.1, Interface address 172.18.0.1
  No backup designated router on this network
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:01
  Supports Link-local Signaling (LLS)
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 0, maximum is 0
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 0, Adjacent neighbor count is 0
  Suppress hello for 0 neighbor(s)

```

Example: Configuring BFD Hardware-Offload support in a BGP Network

The following example shows how to configure BFD Hardware-Offload support in a BGP network.

In this example, the “simple” BGP network consists of SwitchA and SwitchB. Gigabit Ethernet interface 6/1 on SwitchA is connected to the same network as Gigabit Ethernet interface 6/1 in SwitchB.

Configuration for SwitchA

```

!
interface GigabitEthernet 6/1
  no switchport
  ip address 10.1.1.1 255.255.255.0
  bfd interval 100 min_rx 100 multiplier 3
  no bfd echo

router bgp 10
  neighbor 10.1.1.2 remote-as 10
  neighbor 10.1.1.2 fall-over bfd

```

```
!
```

Configuration for SwitchB

```
!
interface GigabitEthernet 6/1
 no switchport
 ip address 10.1.1.2 255.255.255.0
 bfd interval 100 min_rx 100 multiplier 3
 no bfd echo

router bgp 10
 neighbor 10.1.1.1 remote-as 10
 neighbor 10.1.1.1 fall-over bfd
!
```

The output from the **show bfd neighbors details** command from SwitchA verifies that a BFD session has been created and that BGP is registered for BFD support. The relevant command output is shown in bold.

SwitchA

```
SwitchA# show bfd neighbors details
```

```
IPv4 Sessions
NeighAddr                      LD/RD          RH/RS          State          Int
10.1.1.1                        1/1           Up             Up             Gi3/2
Session state is UP and not using echo function.
Session Host: Hardware
OurAddr: 10.1.1.2
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 50000, MinRxInt: 50000, Multiplier: 3
Received MinRxInt: 50000, Received Multiplier: 3
Holddown (hits): 0(0), Hello (hits): 50(0)
Rx Count: 8678
Tx Count: 8680
Elapsed time watermarks: 0 0 (last: 0)
Registered protocols:  BGP
Uptime: 00:06:18
Last packet: Version: 1          - Diagnostic: 0
                  State bit: Up    - Demand bit: 0
                  Poll bit: 0      - Final bit: 0
                  Multiplier: 3    - Length: 24
                  My Discr.: 1      - Your Discr.: 1
                  Min tx interval: 50000 - Min rx interval: 50000
                  Min Echo interval: 0
```

The output from the **show bfd neighbors details** command on SwitchB verifies that a BFD session has been created:

SwitchB

```
SwitchB# attach 6
```

```
Switch> show bfd neighbors details
```

```
IPv4 Sessions
NeighAddr                      LD/RD          RH/RS          State          Int
10.1.1.2                        1/1           Up             Up             Gi1/2
Session state is UP and not using echo function.
Session Host: Hardware
OurAddr: 10.1.1.1
```



```

Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 50000, MinRxInt: 50000, Multiplier: 3
Received MinRxInt: 50000, Received Multiplier: 3
Holddown (hits): 0(0), Hello (hits): 50(0)
Rx Count: 10138
Tx Count: 10139
Elapsed time watermarks: 0 0 (last: 0)
Registered protocols: BGP
Uptime: 00:07:22
Last packet: Version: 1
                State bit: Up
                Poll bit: 0
                Multiplier: 3
                My Discr.: 1
                Min tx interval: 50000
                Min Echo interval: 0
- Diagnostic: 0
- Demand bit: 0
- Final bit: 0
- Length: 24
- Your Discr.: 1
- Min rx interval: 50000

```

The output of the **show ip bgp neighbors** command verifies that BFD has been enabled for the BGP neighbors:

SwitchA

```
SwitchA# show ip bgp neighbors
```

```

BGP neighbor is 10.1.1.2, remote AS 45000, external link
  Using BFD to detect fast fallover
..

```

SwitchB

```
SwitchB# show ip bgp neighbors
```

```

BGP neighbor is 10.1.1.1, remote AS 40000, external link
  Using BFD to detect fast fallover
..

```

Example: Configuring BFD Support for Static Routing

In the following example, the network consists of SwitchA and SwitchB. Gigabit Ethernet interface 6/1 on SwitchA is connected to the same network as gigabit ethernet interface 6/1 on SwitchB. For the BFD session to come up, SwitchB must be configured.

SwitchA

```

configure terminal
no switchport
interface Gigabit Ethernet 6/1
ip address 10.201.201.1 255.255.255.0
bfd interval 500 min_rx 500 multiplier 5
ip route static bfd Gigabit Ethernet 6/1 10.201.201.2
ip route 10.0.0.0 255.0.0.0 Gigabit Ethernet 6/1 10.201.201.2

```

SwitchB

```

configure terminal
no switchport
interface Gigabit Ethernet 6/1
ip address 10.201.201.2 255.255.255.0
bfd interval 500 min_rx 500 multiplier 5
ip route static bfd Gigabit Ethernet 6/1 10.201.201.1
ip route 10.1.1.1 255.255.255.255 Gigabit Ethernet 6/1 10.201.201.1

```

**Note**

The static route on SwitchB exists solely to enable the BFD session between 10.201.201.1 and 10.201.201.2. If there is no useful static route to configure, select a prefix that will not affect packet forwarding, for example, the address of a locally configured loopback interface.

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Configuring and monitoring BGP	“Configuring BGP” module of the Cisco IOS IP Routing Protocols Configuration Guide
Configuring and monitoring EIGRP	“Configuring EIGRP” module of the Cisco IOS IP Routing Protocols Configuration Guide
Configuring and monitoring OSPF	“Configuring OSPF” module of the Cisco IOS IP Routing Protocols Configuration Guide
BFD commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS IP Routing: Protocol-Independent Command Reference
BGP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS IP Routing: Protocol-Independent Command Reference
EIGRP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS IP Routing: Protocol-Independent Command Reference
OSPF commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS IP Routing: Protocol-Independent Command Reference

Standards

Standard	Title
IETF Draft	BFD for IPv4 and IPv6 (Single Hop) , February 2009
IETF Draft	Bidirectional Forwarding Detection , February 2009

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html



Configuring Policy-Based Routing

This chapter describes the tasks for configuring policy-based routing (PBR) on Catalyst 4500 series switches and includes these major sections:

- [Policy-Based Routing, page 42-1](#)
- [Policy-Based Routing Configuration Tasks, page 42-7](#)
- [Policy-Based Routing Configuration Examples, page 42-16](#)

A switch running the LAN Base license level supports PBR for IPv4 and IPv6.



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).



Note

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com, or refer to the software release notes for a specific release.

Policy-Based Routing

PBR gives you a flexible method of routing packets by allowing you to define policies for traffic flows, lessening reliance on routes derived from routing protocols. PBR gives you more control over routing by extending and complementing the existing mechanisms provided by routing protocols. PBR allows you to specify paths for certain traffic, such as priority traffic over a high-cost link.

You can set up PBR as a way to route packets based on configured policies. For example, you can implement routing policies to allow or deny paths based on the identity of a particular end system, or an application protocol. Enable PBR to provide the following advantages:

- Equal access
- Protocol-sensitive routing
- Source-sensitive routing
- Routing based on interactive versus batch traffic
- Routing based on dedicated links

Some applications or traffic can benefit from source-specific routing; for example, you can transfer stock records to a corporate office on a higher-bandwidth, higher-cost link for a short time while sending routine application data, such as e-mail, over a lower-bandwidth, lower-cost link

Policies can be based on IP address, port numbers, or protocols. For a simple policy, use any one of these descriptors; for a complicated policy, all of them.

Route Maps

The following topics are discussed in the section:

- [Understanding Route Maps, page 42-2](#)
- [PBR Route-Map Processing Logic, page 42-3](#)
- [Load Balancing with Recursive Next Hop, page 42-4](#)
- [Packet Matching Criteria, page 42-4](#)
- [PBR Route-Map Processing Logic Example, page 42-4](#)

Understanding Route Maps

All packets received on an interface with PBR enabled (except those sent directly to the switch IP) are handled by enhanced packet filters known as route maps. The route maps dictate the policy that determines where the packets are forwarded.

Route maps contain statements that can be marked as permit or deny. They are interpreted in the following ways:

- If a statement is marked as deny, the packets meeting the match criteria are sent back using the normal forwarding channels and destination-based routing is performed.
- If the statement is marked as permit and a packet matches the access-lists, then the first valid set clause is applied to that packet.

You can implement PBR by applying a route map on an incoming interface. A given interface can have only one route-map configured. A route map is configured at the global configuration parser mode. You can then apply this route map on one or more interfaces (in the interface configuration parser sub-mode).

Each route map statement contains **match** and **set** commands. The **match** command denotes the match criteria to be applied on the packet data. The **set** command denotes the PBR action to be taken on the packet.

The following example shows a single route map called rm-test and six route map statements:

```
route-map rm-test permit 21
  match ip address 101
  set ip next-hop 21.1.1.1
!
route-map rm-test permit 22
  match ip address 102
  set ip next-hop 22.2.2.1
!
route-map rm-test permit 23
  match ip address 101 2102
  set interface vlan23
!
route-map rm-test deny 24
  match ip address 104
  set ip next-hop 24.4.4.1
```

```
!  
route-map rm-test deny 25  
  match ip address 105  
  set ip next-hop 25.5.5.1  
!  
route-map rm-test permit 26  
  match ip address 2104  
  set ip next-hop 26.6.6.1
```

The numbers 21, 22, ... 26 are the sequence numbers of the route-map statements.

PBR Route-Map Processing Logic

When a packet is received on an interface configured with a route map, the forwarding logic processes each route map statement according to the sequence number.

If the route map statement encountered is a **route-map... permit** statement:

- The packet is matched against the criteria in the **match** command. This command may refer to an ACL that may itself have one or more permit and/or deny expressions. The packet is matched against the expressions in the ACL, and a permit/deny decision is reached.
- If the decision reached is permit, then the PBR logic executes the action specified by the **set** command on the packet.
- If the decision reached is deny, then the PBR action (specified in the **set** command) is not applied. Instead the processing logic moves forward to look at the next route-map statement in the sequence (the statement with the next higher sequence number). If no next statement exists, PBR processing terminates, and the packet is routed using the default IP routing table.

If the route map statement encountered is a **route-map... deny** statement:

- The packet is matched against the criteria given in the **match** command. This command may refer to an ACL that may itself have one or more permit and/or deny expressions. The packet is matched against the expressions in the ACL, and a permit/deny decision is reached.
- If the criteria decision is permit, then the PBR processing terminates, and the packet is routed using the default IP routing table.
- If the criteria decision is deny, then the PBR processing logic moves forward to look at the next route-map statement in the sequence (the statement with the next higher sequence number). If no next statement exists, PBR processing terminates, and the packet is routed using the default IP routing table.



Note

The **set** command has no effect inside a **route-map... deny** statement.

A route map statement can have multiple **set** commands that are applied in the following priority:

set ip next-hop verify-availability

set ip next-hop

set ip next-hop recursive

set interface

set default ip next-hop

set default interface

If both the **set ip next-hop** and **set ip next-hop recursive** commands are present in the same route-map statement, the **next-hop set** command is applied.

If the **set ip next-hop** command is not available then the **set ip next-hop recursive** command is applied.

If the **set ip recursive-next-hop** and the **set interface** command are not present, then the packet is routed using the default routing table; it is not dropped. If the packet is required to be dropped, use the **set next-hop recursive** command followed by a **set interface null0 configuration** command.

Load Balancing with Recursive Next Hop

If multiple equal-cost routes to the subnet have been configured by the **set ip next-hop recursive** command, load balancing will occur only if all the adjacencies to the routes are resolved. If any of the adjacencies have not been resolved, then load balancing will not happen and only one of the routes whose adjacency is resolved will be used. If none of the adjacencies are resolved, then packets will be processed in software, resulting in at least one of the adjacencies to be resolved and programmed in hardware. PBR relies on routing protocols or other means to resolve all adjacencies and make load balancing happen.

Packet Matching Criteria

Access Control Lists (ACLs) define the allowed match criteria for packets. Each ACL is applied to incoming packets in a certain order, stopping only when the packet characteristics match the ACL being applied. Unlike policy maps, route maps do not support the "match any" match semantics.

IPv6 packets are matched via a **match ipv6 address** statement in the associated PBR route-map. IPv6 PBR requires IPv6 ACL, although the statement may specify either an IPv6 ACL or an IPv6 Prefixlist.

Packets are matched using the following criteria:

- Input interface
- Source IPv4/IPv6 Address (Prefixlist/Standard/Extended ACL)
- Destination IPv4/IPv6 Address (Standard/Extended ACL)
- Protocol (Extended ACL)
- Source Port and Destination Port (Extended ACL)
- DSCP (Extended ACL)
- Flow-label (Extended ACL)
- Fragment (Extended ACL)

PBR Route-Map Processing Logic Example

Consider a route map called `rm-test` defined as follows:

```
access-list 101 permit tcp host 61.1.1.1 host 133.3.3.1 eq 101
access-list 102 deny tcp host 61.1.1.1 host 133.3.3.1 eq 102
access-list 2102 permit tcp host 61.1.1.1 host 133.3.3.1 eq 102
access-list 104 deny tcp host 61.1.1.1 host 133.3.3.1 eq 104
access-list 2104 permit tcp host 61.1.1.1 host 133.3.3.1 eq 104
access-list 105 permit tcp host 61.1.1.1 host 133.3.3.1 eq 105
```

```
route-map rm-test permit 21
  match ip address 101
  set ip next-hop 21.1.1.1
!
route-map rm-test permit 22
  match ip address 102
  set ip next-hop 22.2.2.1
```



```

!
route-map rm-test permit 23
  match ip address 101 2102
  set interface vlan23
!
route-map rm-test deny 24
  match ip address 104
  set ip next-hop 24.4.4.1
!

route-map rm-test deny 25
  match ip address 105
  set ip next-hop 25.5.5.1
!
route-map rm-test permit 26
  match ip address 2104
  set ip next-hop 26.6.6.1

```

- TCP packet from 61.1.1.1 to 133.3.3.1 with destination port 101
 - Matches ACL 101 in sequence #21.
 - PBR is switched through next-hop 21.1.1.1.



Note ACL 101 is also matched in sequence #23, but the processing doesn't reach that point

- TCP packet from 61.1.1.1 to 133.3.3.1 with destination port 102
 - In sequence #21, the ACL 101 action denies this packet (because all ACLs have an implicit deny). Processing advances to sequence #22.
 - In sequence #22, ACL 102 matches TCP port 102, but the ACL action is deny. Processing advances to sequence #23.
 - In sequence #23, ACL 2102 matches TCP port 102, and the ACL action is permit.
 - Packet is switched to output interface VLAN 23.
- TCP packet from 61.1.1.1 to 133.3.3.1 with destination port 105
 - Processing moves from sequence #21 to #24, because all ACLs in these sequence numbers have a deny action for port 105.
 - In sequence #25, ACL 105 has a permit action for TCP port 105.
 - The route-map deny command takes effect, and the packet is routed using the default IP routing table.

The Catalyst 4500 series switch supports matching route map actions with a packet by installing entries in the TCAM that match the set of packets described by the ACLs in the match criteria of the route map. These TCAM entries point at adjacencies that either perform the necessary output actions or forward the packet to software if either hardware does not support the action or its resources are exhausted.

If the route map specifies a **set interface ...** action, packets that match the **match** statement are routed in the software. Some packets may be dropped. Similarly, if the route-map specifies a **set default interface...** action and there is no matching IP route for the packet, the packet is routed in the software.



Note

The scale of hardware-based PBR is determined by the TCAM size and the time required for the CPU to flatten the ACL before programming into the hardware. The time take to flatten the ACL increases when a PBR policy requires a considerable number of route-maps. For example, a PBR policy of 1,200

route-maps (each containing ACLs with permit ACEs only) may require 6-7 minutes of flatten time before programming into hardware. This process may repeat if an adjacency change requires PBR reprogramming.

Policy-Based Routing with Object Tracking

Beginning in Cisco IOS XE Release 3.8.0E and Cisco IOS Release 15.2(4)E, you can configure Policy-Based Routing (PBR) to use object tracking, to verify the most viable next-hop IP address to which to forward packets, using an Internet Control Message Protocol (ICMP) ping as the verification method. PBR used with object tracking is most suitable for devices that have multiple Ethernet connections as the next hop. Normally, Ethernet interfaces connect to digital subscriber line (DSL) modems or cable modems, and do not detect a failure upstream in the ISP broadband network. The Ethernet interface remains up, and any form of static routing points to that interface. Using PBR with object tracking allows you to back-up two Ethernet interfaces, determine the interface that is available by sending ICMP pings to verify if the IP address can be reached, and then route traffic to that interface.

To verify the next-hop IP address for the device, PBR informs the object tracking process that it is interested in tracking a certain object. The tracking process, in turn, informs PBR when the state of the object changes.

Restrictions for Policy-Based Routing with Object Tracking

The **set next-hop verify-availability** command is not supported with the following:

- VRF instances
- Virtual switching system (VSS)
- IPv6 traffic

IPv4 and IPv6 Policy-Based Routing for VRF Instances

Virtual routing and forwarding (VRF) allows multiple routing instances in Cisco software. Beginning in Cisco IOS Release XE 3.7 0E and IOS 15.2(3)E, the Policy-Based Routing (PBR) feature is VRF-aware and works on multiple routing instances, beyond the default or global routing table.

Incoming packets are filtered through the match criteria that are defined in the route map. After a successful match occurs, the **set** command configuration determines the VRF through which outbound packets are policy routed.

Inherit-VRF, Inter-VRF, Global-to-VRF, and VRF-to-Global Routing

The Policy-Based Routing feature supports inherit-VRF, inter-VRF, and VRF-to-global routing.

With inherit-VRF, packets arriving at a virtual routing and forwarding (VRF) interface are routed, by looking-up the same VRF's routing table.

With inter-VRF routing, packets arriving at a VRF interface are routed, by looking-up a different VRF's routing table, as specified by the **set** command.

With VRF-to-global routing, packets arriving at a VRF interface are routed via the global routing table.

With global-to-VRF routing, packets arriving at the global interface (an interface that is not part of a VRF) are routed via a VRF routing table.

Restrictions for VRF-Aware Policy-Based Routing

- VRF is not supported at the LAN Base license level.
- The same route-map cannot be used to configure PBR:
 - on interfaces that belong to different VRFs
 - on one VRF interface and another global interface (an interface that is not part of a VRF).
- The **set vrf** and **set ip global next-hop** commands can be configured with the **set default interface**, **set interface**, **set ip default next-hop**, and **set ip next-hop** commands. But the **set vrf** and **set ip global next-hop** commands take precedence over the **set default interface**, **set interface**, **set ip default next-hop**, and **set ip next-hop** commands. No error message is displayed if you attempt to configure the **set vrf** command with any of these three set commands.
- The **set global** and **set vrf** commands cannot be simultaneously applied to a route map.
- When you use the **set vrf** command you specify the VRF table to be looked-up; this overrides the default or global routing table. If a route is not specified in the VRF routing table, then packets are dropped (even if a route exists in the global routing table).
- The **set next-hop verify-availability** and the **set ip next hop recursive** commands are not supported within VRF instances.

Policy-Based Routing Configuration Tasks

To configure PBR, perform the tasks described in the following sections. The task in the first section is required; the tasks in the remaining sections are optional. For configuration examples, see the [“Policy-Based Routing Configuration Examples” section on page 42-16](#).

- [Enabling IPv4 PBR, page 42-7](#) (Required)
- [Enabling IPv6 PBR, page 42-10](#) (Required)
- [Enabling Local IPv4 and Local IPv6 PBR, page 42-12](#) (Optional)
- [Verifying Next-Hop IP using Object Tracking , page 42-14](#) (Optional)
- [Unsupported Commands, page 42-15](#) (Optional)
- [Configuring IPv4 and IPv6 PBR for VRF Instances, page 42-12](#) (Optional)
- [Unsupported Commands, page 42-15](#)

Enabling IPv4 PBR

To enable PBR, you must create a route map that specifies the match criteria and the resulting action if all of the match clauses are met. Then you must apply that route-map on a particular interface. All packets arriving on the specified interface matching the match clauses are subject to PBR.

To enable IPv4 PBR on an interface, perform this task:

	Command	Purpose
Step 1	Switch(config)# route-map <i>map-tag</i> [permit deny] [<i>sequence-number</i>]	Defines a route map to control where packets are sent. This command puts the switch into route-map configuration mode.
Step 2	Switch(config-route-map)# match ip address { <i>access-list-number</i> <i>name</i> } [... <i>access-list-number</i> <i>name</i>]	Specifies the match criteria. The match criteria take the form of one or more Standard or Extended IP access-lists. The access-lists can specify the source and destination IP addresses, protocol types, and port numbers.
Step 3	Switch(config-route-map)# set ip next-hop <i>ip-address</i> [... <i>ip-address</i>]	Specifies the next-hop IP address to which matching packets are sent. The next-hop IP address specified here must belong to a subnet that is directly connected to this switch. If more than one next-hop IP address is specified, the first usable next-hop is chosen for routing matching packets. If the next-hop is (or becomes) unavailable for some reason, the next one in the list is chosen.
Step 4	Switch(config-route-map)# set ip next-hop verify-availability [<i>next-hop-address-sequence</i> track <i>object</i>]	(Optional) Configures the route map to verify the reachability of the tracked object. Note This option is not supported for IPv6 traffic. For information about defining new tracked object, see Verifying Next-Hop IP using Object Tracking , page 42-14
Step 5	Switch(config-route-map)# set ip next-hop recursive <i>ip-address</i>	Specifies a recursive next-hop IP address. Note The recursive next-hop can be a subnet that is not directly connected. The set ip next-hop recursive command does not ensure that packets are routed through the recursive-next-hop if there is an intermediate node with a shorter route to the destination such that the route does not pass through the recursive-next-hop.
Step 6	Switch(config-route-map)# set interface <i>interface-type interface-number</i> [... <i>type number</i>]	Specifies the output interface from which the packet will be sent. This action specifies that the packet is forwarded out of the local interface. The interface must be a Layer 3 interface (not a switchport). Packets are forwarded on the specified interface only if one of the following conditions is met: <ul style="list-style-type: none"> • The destination IP address in the packet lies within the IP subnet to which the specified interface belongs. • The destination IP address in the packet is reachable through the specified interface (as per the IP routing table). If the destination IP address on the packet does not meet either of these conditions, the packet is dropped. This action forces matching packets to be switched in software.k

	Command	Purpose
Step 7	Switch(config-route-map)# set ip default next-hop <i>ip-address</i> [... <i>ip-address</i>]	Sets next hop to which to route the packet if there is no explicit route for the destination IP address in the packet. Before forwarding the packet to the next hop, the switch looks up the packet's destination address in the unicast routing table. If a match is found, the packet is forwarded by way of the routing table. If no match is found, the packet is forwarded to the specified next hop.
Step 8	Switch(config-route-map)# set default interface <i>interface-type interface-number</i> [... <i>type</i> ... <i>number</i>]	<p>Specifies the output interface from which the packet will be sent if there is no explicit route for this destination. Before forwarding the packet to the next hop, the switch looks up the packet's destination address in the unicast routing table. If a match is found, the packet is forwarded by using the routing table. If no match is found, the packet is forwarded to the specified output interface.</p> <p>Packets are forwarded on the specified interface only if one of the following conditions is met:</p> <ul style="list-style-type: none"> • The destination IP address in the packet lies within the IP subnet to which the specified interface belongs. • The destination IP address in the packet is reachable through the specified interface (as per the IP routing table). <p>If the destination IP address on the packet does not meet either of these conditions, the packet is dropped. This action forces matching packets to be switched in software.</p>
Step 9	Switch(config-route-map)# interface <i>interface-type interface-number</i>	Specifies the interface. This command puts the switch into interface configuration mode.
Step 10	Switch(config-if)# ip policy route-map <i>map-tag</i>	Identifies the route map to use for PBR. One interface can only have one route map tag, but you can have multiple route map entries with different sequence numbers. These entries are evaluated in sequence number order until the first match. If no match exists, packets are routed as usual.

Use the **set** commands with each other. These commands are evaluated in the order shown in Step 3 in the previous task table. A usable next hop implies an interface. Once the local switch finds a next hop and a usable interface, it routes the packet.

Refer to the section [Policy-Based Routing Configuration Examples, page 42-16](#) for examples of IPv4 PBR.

Use the **show route-map map-tag** command to display the existing route map.



Note

Packet and byte counters in the output of the **show route-map map-tag** command are not updated.

Enabling IPv6 PBR



Note

With IOS XE 3.6.0E and IOS 15.2(2)E, IPv6 PBR is not supported on Supervisor Engine 8-E.

To enable PBR, you must create a route map that specifies the match criteria and the resulting action if all of the match clauses are met. Then you must apply that route-map on a particular interface. All packets arriving on the specified interface matching the match clauses are subject to PBR.

To enable IPv6 PBR on an interface, perform this task:

	Command	Purpose
Step 1	Switch(config)# route-map map-tag [permit deny] [sequence-number]	Defines a route map to control where packets are sent. This command puts the switch into route-map configuration mode.
Step 2	Switch(config-route-map)# match ipv6 address {access-list-number name} [...access-list-number name]	Specifies the match criteria. The match criteria take the form of one or more Standard or Extended ipv6 access-lists. The access-lists can specify the source and destination IP addresses, protocol types, and port numbers.
Step 3	Switch(config-route-map)# set ipv6 next-hop ip-address [... ip-address]	Specifies the next-hop IP address to which matching packets are sent. The next-hop IP address specified here must belong to a subnet that is directly connected to this switch. If more than one next-hop IP address is specified, the first usable next-hop is chosen for routing matching packets. If the next-hop is (or becomes) unavailable for some reason, the next one in the list is chosen.
Step 4	Switch(config-route-map)# set interface interface-type interface-number [... type number]	Specifies the output interface from which the packet will be sent. This action specifies that the packet is forwarded out of the local interface. The interface must be a Layer 3 interface (not a switchport). Packets are forwarded on the specified interface only if one of the following conditions is met: <ul style="list-style-type: none"> The destination IP address in the packet lies within the IP subnet to which the specified interface belongs. The destination IP address in the packet is reachable through the specified interface (as per the IP routing table). If the destination IP address on the packet does not meet either of these conditions, the packet is dropped. This action forces matching packets to be switched in software.k
Step 5	Switch(config-route-map)# set ipv6 default next-hop ip-address [... ip-address]	Sets next hop to which to route the packet if there is no explicit route for the destination IP address in the packet. Before forwarding the packet to the next hop, the switch looks up the packet's destination address in the unicast routing table. If a match is found, the packet is forwarded by way of the routing table. If no match is found, the packet is forwarded to the specified next hop.

	Command	Purpose
Step 6	Switch(config-route-map)# set default interface <i>interface-type interface-number</i> [...type ...number]	<p>Specifies the output interface from which the packet will be sent if there is no explicit route for this destination. Before forwarding the packet to the next hop, the switch looks up the packet's destination address in the unicast routing table. If a match is found, the packet is forwarded by using the routing table. If no match is found, the packet is forwarded to the specified output interface.</p> <p>Packets are forwarded on the specified interface only if one of the following conditions is met:</p> <ul style="list-style-type: none"> • The destination IP address in the packet lies within the IP subnet to which the specified interface belongs. • The destination IP address in the packet is reachable through the specified interface (as per the IP routing table). <p>If the destination IP address on the packet does not meet either of these conditions, the packet is dropped. This action forces matching packets to be switched in software.</p>
Step 7	Switch(config-route-map)# interface <i>interface-type interface-number</i>	Specifies the interface. This command puts the switch into interface configuration mode.
Step 8	Switch(config-if)# ipv6 policy route-map <i>map-tag</i>	Identifies the route map to use for PBR. One interface can only have one route map tag, but you can have multiple route map entries with different sequence numbers. These entries are evaluated in sequence number order until the first match. If no match exists, packets are routed as usual.

**Note**

The **recursive** option is supported for IPv4, but not for IPv6. An interface can have either an ipv4 route map or an ipv6 route map. An interface can be bound to only one route map.

Use the **set** commands with each other. These commands are evaluated in the order shown in Step 3 in the previous task table. A usable next hop implies an interface. Once the local switch finds a next hop and a usable interface, it routes the packet.

Refer to the following document for IPv6 PBR configuration examples.

<http://www.cisco.com/c/en/us/support/docs/ip/ip-version-6-ipv6/112218-policy-based-routing-ipv6-configex.html>

**Note**

Packet and byte counters in the output of the **show route-map map-tag** command are updated only for software switched packets. Counters for hardware switched packets are not updated.

Enabling Local IPv4 and Local IPv6 PBR

Packets that are generated by the switch are not normally policy-routed. To enable local PBR for such packets, indicate which route map the switch should use by entering this command:

IPv4

Command	Purpose
Switch(config)# ip local policy route-map <i>map-tag</i>	Identifies the IPv4 route map to use for local PBR.

IPv6

Command	Purpose
Switch(config)# ipv6 local policy route-map <i>map-tag</i>	Identifies the IPv6 route map to use for local PBR.

All packets originating on the switch are then subject to local PBR.
Use the **show ip local policy** command to display the route map used for local PBR, if one exists.

Configuring IPv4 and IPv6 PBR for VRF Instances

To enable PBR for multiple routing instances, configure your device in the following way:

	Command	Purpose
Step 1	Switch(config)# route-map <i>map-tag</i> [permit deny] [<i>sequence-number</i>]	Defines a route map to control where packets are sent. This command puts the switch into route-map configuration mode.
Step 2	For IPv4: Switch(config-route-map)# match ip address { <i>access-list-number</i> <i>name</i> } [... <i>access-list-number</i> <i>name</i>] For IPv6: Switch(config-route-map)# match ipv6 address { <i>access-list-number</i> <i>name</i> } [... <i>access-list-number</i> <i>name</i>]	Specifies the match criteria. The match criteria take the form of one or more Standard or Extended access-lists. The access-lists can specify the source and destination IP addresses, protocol types, and port numbers.

	Command	Purpose
Step 3	Set one of the following	Specifies the next-hop IP address under the VRF, to which the matched packets must be forwarded. The next-hop IP address must exist in the routing table, under the VRF.
	<p>For IPv4:</p> <pre>Switch(config-route-map)# set ip vrf [vrf name] next-hop ip address> [...ip-address]</pre> <p>For IPv6:</p> <pre>Switch(config-route-map)# set ipv6 vrf [vrf name] next-hop ip address> [...ip-address]</pre>	
Step 4	<p>For IPv4:</p> <pre>Switch(config-route-map)# set ip global next-hop ip address> [...ip-address]</pre> <p>For IPv6:</p> <pre>Switch(config-route-map)# set ipv6 global next-hop ip address> [...ip-address]</pre>	Specifies the next-hop IP address, from the global routing table, to which to forward matched packets. The next-hop IP address must exist in the global routing table.
	Set one of the following:	Specifies the next-hop IP address to which the matched packets must be forwarded when there is no explicit packet destination address in the routing table, under the VRF.
	<p>For IPv4:</p> <pre>Switch(config-route-map)# set ip default vrf [vrf name] next-hop ip address> [...ip-address]</pre> <p>For IPv6:</p> <pre>Switch(config-route-map)# set ipv6 default vrf [vrf name] next-hop ip address> [...ip-address]</pre>	
	<p>For IPv4:</p> <pre>Switch(config-route-map)# set ip default global next-hop ip address> [...ip-address]</pre> <p>For IPv6:</p> <pre>Switch(config-route-map)# set ipv6 default global next-hop ip address> [...ip-address]</pre>	Specifies the next-hop IP address to which the matched packets must be forwarded when there is no explicit packet destination address corresponding to the VRF to which the interface belongs, in the routing table. The next-hop address specified must exist in the global routing table.

	Command	Purpose
Step 5	<p>Set one of the following:</p> <p>For global routing:</p> <pre>Switch(config-route-map)# set global</pre> <p>For inter-VRF routing:</p> <pre>Switch(config-route-map)# set vrf [vrf name]</pre>	<p>Specifies that the global routing table should be looked-up to route packets,</p> <p>Use the set global command to configure VRF-to-Global routing.</p> <p>Use the set vrf command to specify the VRF table to be looked-up, to route packets.</p> <p>Use this command to configure Inter-VRF routing and route packets arriving at a particular VRF interface through a different VRF interface, by looking-up a different VRF's routing table. This command overrides the default or global routing table. If a route is not specified in the VRF routing table, then packets are dropped (even if a route exists in the global routing table)</p>
Step 6	<pre>Switch(config-route-map)# interface interface-type interface-number</pre>	Specifies the interface. This command puts the switch into interface configuration mode.
Step 7	<pre>Switch(config-if)# ipv6 policy route-map map-tag</pre>	Identifies the route map to use for PBR. One interface can only have one route map tag, but you can have multiple route map entries with different sequence numbers. These entries are evaluated in sequence number order until the first match. If no match exists, packets are routed as usual.

Verifying the PBR Configuration for VRF Instances

To verify the PBR configuration for VRF instances, enter the following steps in any order:

	Command	Purpose
Step 1	<pre>Switch# show ip access list [access-list-number access-list-name]</pre>	Displays the subnet ranges defined as match criteria in the standard access lists.
Step 2	<pre>Switch# show route-map [map-name]</pre>	Displays the match and set commands in the route map.

The following example shows you how to configure PBR for VRF instances:

```
Switch# enable
Switch# configure terminal
Switch(config)# route-map map1 permit 10
Switch(config-route-map)# set ipv6 vrf myvrf next-hop 2001.DB8:4:1::1/64
Switch(config-route-map)# end
Switch# show route-map map1
```

Verifying Next-Hop IP using Object Tracking

To verify the next-hop IP address using PBR with Object Tracking, perform the following steps:



Note

The **set ip next-hop verify-availability** command is not supported on VRF instances, on a virtual switching system (VSS), and with IPv6 traffic.

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch (config)# track [<i>object-number</i>] ip sla [<i>entry-number</i>]	Tracks the state of the specified IP SLA object.
Step 3	Switch (config)# ip sla [<i>operation-number</i>]	Starts a Cisco IOS IP Service Level Agreement (SLA) operation configuration and enters IP SLA configuration mode.
Step 4	Switch (config-ip-sla)# icmp echo [<i>ip-address</i>] source ip [<i>ip-address</i>]	Configures an IP SLA Internet Control Message Protocol (ICMP) echo probe operation and enters Echo configuration mode.
Step 5	Switch (config-ip-sla-echo)# frequency seconds	(Optional) Sets the rate at which a specified IP SLA operation repeats.
Step 6	Switch (config-ip-sla-echo)# threshold milliseconds	(Optional) Sets the length of time required for a rising threshold event to be declared.
Step 7	Switch (config-ip-sla-echo)# timeout milliseconds	(Optional) Sets the maximum time required for the IP SLA operation to be completed.
Step 8	Switch (config)# ip sla schedule [<i>operation-number</i>] [life { <i>forever</i> <i>seconds</i> }] [start-time { <i>hh</i> : <i>mm</i> } [<i>:ss</i>] [<i>monthday</i> <i>daymonth</i>] <i>pending</i> <i>now</i> <i>after hh</i> : <i>mm</i> : <i>ss</i> }] [<i>ageout seconds</i>]	Configures the scheduling parameters for a single Cisco IOS IP SLA operation.
Step 9	Switch(config)# route-map <i>map-tag</i> [<i>permit</i> <i>deny</i>] [<i>sequence-number</i>]	Specifies a route map and enters route-map configuration mode.
Step 10	Switch(config-route-map)# match ip address [<i>access-list-name</i>]	Distributes routes that have a destination IPv4 network number address that is permitted by a standard access list.
Step 11	Switch(config-route-map)# set ip next-hop verify-availability [<i>next-hop-address</i> <i>sequence</i> track <i>object</i>]	Configures the route map to verify the reachability of the tracked object.

The following example shows you how to verify the next-hop IP address in a route map:

```
Switch# enable
Switch# configure terminal
Switch(config)# track 100 ip sla 100
Switch(config)# ip sla 100
switch(config-ip-sla)# icmp-echo 172.19.255.253 source-ip 172.19.255.47
switch(config-ip-sla-echo)# timeout 1500
switch(config-ip-sla-echo)# threshold 1000
switch(config-ip-sla-echo)# frequency 2
switch(config)# ip sla schedule 100 life forever start-time now
switch(config)# route-map alpha permit 10
switch(config-route-map)# match ip address exlist
switch(config-route-map)# set ip next-hop verify-availability 95.1.1.2 1 track 100
switch# show route-map alpha
switch# show track 100
```

Unsupported Commands

The following PBR commands in config-route-map mode are in the CLI but not supported in Cisco IOS for the Catalyst 4500 series switches. If you attempt to use these commands, an error message displays:

- **match-length**
- **set ip qos** and **set ipv6 qos**
- **set ip tos** and **set ipv6 tos**
- **set ip precedence** and **set ipv6 precedence**
- **set ip df** and **set ipv6 df**
- **set ipv6 next-hop recursive**
- **set ipv6 next-hop verify-availability**

Policy-Based Routing Configuration Examples

The following sections provide PBR configuration examples:

- [Equal Access, page 42-16](#)
- [Differing Next Hops, page 42-17](#)
- [Deny ACE, page 42-17](#)

For information on how to configure policy-based routing, see the section “[Policy-Based Routing Configuration Tasks](#)” in this chapter.

Equal Access

The following example provides two sources with equal access to two different service providers. Packets arriving on interface fastethernet 3/1 from the source 10.1.1.1 are sent to the switch at 6.6.6.6 if the switch has no explicit route for the destination of the packet. Packets arriving from the source 2.2.2.2 are sent to the switch at 7.7.7.7 if the switch has no explicit route for the destination of the packet. All other packets for which the switch has no explicit route to the destination are discarded.

```
Switch (config)# access-list 1 permit ip 10.1.1.1
access-list 2 permit ip 2.2.2.2
!
interface fastethernet 3/1
 ip policy route-map equal-access
!

route-map equal-access permit 10
 match ip address 1
 set ip default next-hop 6.6.6.6
route-map equal-access permit 20
 match ip address 2
 set ip default next-hop 7.7.7.7
route-map equal-access permit 30
 set default interface null0
```



Note

If the packets you want to drop do not match either of the first two route-map clauses, then change **set default interface null0** to **set interface null0**.

Differing Next Hops

The following example illustrates how to route traffic from different sources to different places (next hops). Packets arriving from source 10.1.1.1 are sent to the next hop at 3.3.3.3; packets arriving from source 2.2.2.2 are sent to the next hop at 3.3.3.5.

```
access-list 1 permit ip 10.1.1.1
access-list 2 permit ip 2.2.2.2
!
interface fastethernet 3/1
 ip policy route-map Texas
!
route-map Texas permit 10
 match ip address 1
 set ip next-hop 3.3.3.3
!
route-map Texas permit 20
 match ip address 2
 set ip next-hop 3.3.3.5
```

Deny ACE

The following example illustrates how to stop processing a given route map sequence, and to jump to the next sequence. Packets arriving from source 10.1.1.1 skip sequence 10 and jump to sequence 20. All other packets from subnet 10.1.1.0 follow the set statement in sequence 10.

```
access-list 1 deny ip 10.1.1.1
access-list 1 permit ip 10.1.1.0 0.0.0.255
access-list 2 permit ip 10.1.1.1
access-list 2 permit ip 2.2.2.2
!
interface fastethernet 3/1
 ip policy route-map Texas
!
route-map Texas permit 10
 match ip address 1
 set ip next-hop 3.3.3.3
!
route-map Texas permit 20
 match ip address 2
 set ip next-hop 3.3.3.5
```

Examples of the show Command

The following **show** command illustrates that route map pbrv6-test has only one permit sequence.

In the example policy, IPv6 packets with an address matching the criteria defined by the access control list v6_acl are forwarded to the next hop 2006::2. If next-hop 2006::2 is unreachable, the matching packets are forwarded to 2005::2. If both next-hops are unreachable, the packets are forwarded using the routing table lookup. For packets that do not match the filter criteria, a standard routing table lookup is performed for packet forwarding.

```
Switch# show route-map pbrv6-test
route-map pbrv6-test, permit, sequence 10
  Match clauses:
    ipv6 address v6_acl
  Set clauses:
    ipv6 next-hop 2006::2 2005::2
```

```
Policy routing matches: 0 packets, 0 bytes
```



Configuring VRF-lite

Virtual Private Networks (VPNs) provide a secure way for customers to share bandwidth over an ISP backbone network. A VPN is a collection of sites sharing a common routing table. A customer site is connected to the service provider network by one or more interfaces, and the service provider associates each interface with a VPN routing table. A VPN routing table is called a VPN routing/forwarding (VRF) table.

With the VRF-lite feature, the Catalyst 4500 series switch supports multiple VPN routing/forwarding instances in customer edge devices. (VRF-lite is also termed multi-VRF CE, or multi-VRF Customer Edge Device). VRF-lite allows a service provider to support two or more VPNs with overlapping IP addresses using one interface.

This document addresses both IPv4 and IPv6 VRF-lite.



Note

Starting with Cisco IOS Release 12.2(52)SG, the Catalyst 4500 switch supports VRF-lite NSF support with routing protocols OSPF/EIGRP/BGP.

This chapter includes these topics:

- [About VRF-lite, page 43-2](#)
- [VRF-lite Configuration Guidelines, page 43-3](#)
- [Configuring VRF-lite for IPv4, page 43-5](#)
- [Configuring VRF-lite for IPv6, page 43-15](#)
- [VPN Co-existence Between IPv4 and IPv6, page 43-28](#)
- [Migrating from the Old to New CLI Scheme, page 43-28](#)



Note

The switch does not use Multiprotocol Label Switching (MPLS) to support VPNs. For information about MPLS VRF, refer to the *Cisco IOS Switching Services Configuration Guide* at: http://www.cisco.com/en/US/docs/ios/mps/configuration/guide/mp_vpn_ipv4_ipv6_ps6922_TSD_Products_Configuration_Guide_Chapter.html

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About VRF-lite

VRF-lite is a feature that enables a service provider to support two or more VPNs, where IP addresses can be overlapped among the VPNs. VRF-lite uses input interfaces to distinguish routes for different VPNs and forms virtual packet-forwarding tables by associating one or more Layer 3 interfaces with each VRF. Interfaces in a VRF can be either physical, such as Ethernet ports, or logical, such as VLAN SVIs, but a Layer 3 interface cannot belong to more than one VRF at any time.



Note

VRF-lite interfaces must be Layer 3 interfaces.

VRF-lite includes these devices:

- Customer edge (CE) devices provide customer access to the service provider network over a data link to one or more provider edge routers. The CE device advertises the site's local routes to the provider edge router and learns the remote VPN routes from it. A Catalyst 4500 series switch can be a CE.
- Provider edge (PE) routers exchange routing information with CE devices by using static routing or a routing protocol such as BGP, RIPv1, or RIPv2.

The PE is only required to maintain VPN routes for those VPNs to which it is directly attached, eliminating the need for the PE to maintain all of the service provider VPN routes. Each PE router maintains a VRF for each of its directly connected sites. Multiple interfaces on a PE router can be associated with a single VRF if all of these sites participate in the same VPN. Each VPN is mapped to a specified VRF. After learning local VPN routes from CEs, a PE router exchanges VPN routing information with other PE routers by using internal BGP (iBGP).

- Provider routers (or core routers) are any routers in the service provider network that do not attach to CE devices.

With VRF-lite, multiple customers can share one CE, and only one physical link is used between the CE and the PE. The shared CE maintains separate VRF tables for each customer and switches or routes packets for each customer based on its own routing table. VRF-lite extends limited PE functionality to a CE device, giving it the ability to maintain separate VRF tables to extend the privacy and security of a VPN to the branch office.

Figure 43-1 shows a configuration where each Catalyst 4500 series switches acts as multiple virtual CEs. Because VRF-lite is a Layer 3 feature, each interface in a VRF must be a Layer 3 interface.

Figure 43-1 Catalyst 4500 Series Switches Acting as Multiple Virtual CEs

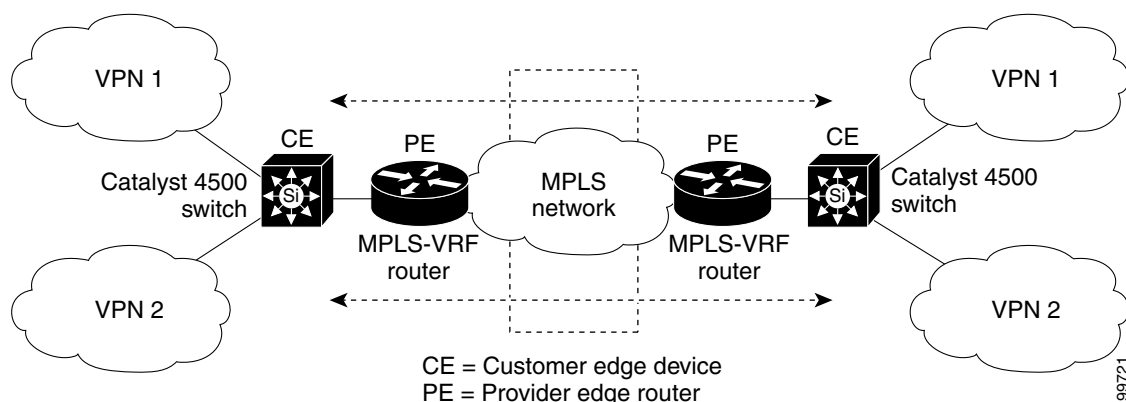


Figure 43-1 illustrates the packet-forwarding process in a VRF-lite CE-enabled network.

- When the CE receives a packet from a VPN, it looks up the routing table based on the input interface. When a route is found, the CE forwards the packet to the PE.
- When the ingress PE receives a packet from the CE, it performs a VRF lookup. When a route is found, the router adds a corresponding MPLS label to the packet and sends it to the MPLS network.
- When an egress PE receives a packet from the network, it strips the label and uses the label to identify the correct VPN routing table. The egress PE then performs the normal route lookup. When a route is found, it forwards the packet to the correct adjacency.
- When a CE receives a packet from an egress PE, it uses the input interface to look up the correct VPN routing table. If a route is found, the CE forwards the packet within the VPN.

To configure VRF, create a VRF table and specify the Layer 3 interface associated with the VRF. You then configure the routing protocols in the VPN and between the CE and the PE. BGP is the preferred routing protocol used to distribute VPN routing information across the providers' backbone. The VRF-lite network has three major components:

- VPN route target communities—Lists all other members of a VPN community. You need to configure VPN route targets for each VPN community member.
- Multiprotocol BGP peering of VPN community PE routers—Propagates VRF reachability information to all members of a VPN community. You need to configure BGP peering in all PE routers within a VPN community.
- VPN forwarding—Transports all traffic between all VPN community members across a VPN service-provider network.

VRF-lite Configuration Guidelines

IPv4 and IPv6

- For the RIP, OSPF, PIM and PBR protocols, VRF is not supported at the LAN Base license level.
- A switch with VRF-lite is shared by multiple customers, and all customers have their own routing tables.
- Because customers use different VRF tables, you can reuse the same IP addresses. Overlapped IP addresses are allowed in different VPNs.
- VRF-lite lets multiple customers share the same physical link between the PE and the CE. Trunk ports with multiple VLANs separate packets among customers. All customers have their own VLANs.
- VRF-lite does not support all MPLS-VRF functionality: label exchange, LDP adjacency, or labeled packets.
- For the PE router, there is no difference between using VRF-lite or using multiple CEs. In Figure 43-1, multiple virtual Layer 3 interfaces are connected to the VRF-lite device.
- The Catalyst 4500 series switch supports configuring VRF by using physical ports, VLAN SVIs, or a combination of both. You can connect SVIs through an access port or a trunk port.
- A customer can use multiple VLANs as long because they do not overlap with those of other customers. A customer's VLANs are mapped to a specific routing table ID that is used to identify the appropriate routing tables stored on the switch.

- The Layer 3 TCAM resource is shared between all VRFs. To ensure that any one VRF has sufficient CAM space, use the **maximum routes** command.
- A Catalyst 4500 series switch using VRF can support one global network and up to 64 VRFs. The total number of routes supported is limited by the size of the TCAM.
- A single VRF can be configured for both IPv4 and IPv6.
- PBR and VRF cannot be configured on the same interface. Similarly, WCCP, Etherchannel and MEC cannot be configured on the same interface with VRF.
- If an incoming packet's destination address is not found in the vrf table, the packet is dropped. Also, if insufficient TCAM space exists for a VRF route, hardware switching for that VRF is disabled and the corresponding data packets are sent to software for processing.

IPv4 Specific

- You can use most routing protocols (BGP, OSPF, EIGRP, RIP and static routing) between the CE and the PE. However, we recommend using external BGP (EBGP) for these reasons:
 - BGP does not require multiple algorithms to communicate with multiple CEs.
 - BGP is designed for passing routing information between systems run by different administrations.
 - BGP makes simplifies passing attributes of the routes to the CE.
- VRF-lite does not support IGRP and ISIS.
- Beginning with Cisco IOS Release 12.2(50)SG, Multicast and VRF can be configured together on a Layer 3 interface.
- The Catalyst 4500 series switch supports all the PIM protocols (PIM-SM, PIM-DM, PIM-SSM, PIM BiDIR).
- The **capability vrf-lite** subcommand under **router ospf** should be used when configuring OSPF as the routing protocol between the PE and the CE.

IPv6 specific

- VRF-aware OSPFv3, BGPv6, EIGRPv6, and IPv6 static routing are supported.
- VRF aware ISISv6, RIPng, IPv6 Multicast Routing(MVRF), and PIMv6 are not supported.
- VRF-aware IPv6 route applications include: ping, telnet, ssh, tftp, ftp and traceroute. (This list does not include the Mgt interface, which is handled differently even though you can configure both IPv4 or IPv6 VRF under it.)

Configuring VRF-lite for IPv4

Configuring VRFs

To configure one or more VRFs, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# ip routing	Enables IP routing.
Step 3	Switch(config)# ip vrf <i>vrf-name</i>	Names the VRF and enters VRF configuration mode.
Step 4	Switch(config-vrf)# rd <i>route-distinguisher</i>	Creates a VRF table by specifying a route distinguisher. Enter either an AS number and an arbitrary number (xxx:y) or an IP address and arbitrary number (A.B.C.D:y).
Step 5	Switch(config-vrf)# route-target { export import both } <i>route-target-ext-community</i>	Creates a list of import, export, or import and export route target communities for the specified VRF. Enter either an AS system number and an arbitrary number (xxx:y) or an IP address and an arbitrary number (A.B.C.D:y). Note This command is effective only if BGP is running.
Step 6	Switch(config-vrf)# import map <i>route-map</i>	(Optional) Associates a route map with the VRF.
Step 7	Switch(config-vrf)# interface <i>interface-id</i>	Enters interface configuration mode and specify the Layer 3 interface to be associated with the VRF. The interface can be a routed port or SVI.
Step 8	Switch(config-if)# ip vrf forwarding <i>vrf-name</i>	Associates the VRF with the Layer 3 interface.
Step 9	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 10	Switch# show ip vrf [brief detail interfaces] [<i>vrf-name</i>]	Verifies the configuration. Displays information about the configured VRFs.
Step 11	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.



Note

For complete syntax and usage information for the following commands, see the switch command reference for this release and see the *Cisco IOS Switching Services Command Reference* at: http://www.cisco.com/en/US/docs/ios/ipswitch/command/reference/isw_book.html

Use the **no ip vrf** *vrf-name* global configuration command to delete a VRF and to remove all interfaces from it. Use the **no ip vrf forwarding** interface configuration command to remove an interface from the VRF.

Configuring VRF-Aware Services

IP services can be configured on global interfaces and within the global routing instance. IP services are enhanced to run on multiple routing instances; they are VRF-aware. Any configured VRF in the system can be specified for a VRF-aware service.

VRF-aware services are implemented in platform-independent modules. VRF provides multiple routing instances in Cisco IOS. Each platform has its own limit on the number of VRFs it supports.

VRF-aware services have the following characteristics:

- The user can ping a host in a user-specified VRF.
- ARP entries are learned in separate VRFs. The user can display Address Resolution Protocol (ARP) entries for specific VRFs.

Configuring the User Interface for ARP

To configure VRF-aware services for ARP, perform this task:

Command	Purpose
Switch# show ip arp vrf <i>vrf-name</i>	Displays the ARP table (static and dynamic entries) in the specified VRF.
Switch(config)# arp vrf <i>vrf-name</i> <i>ip-address mac-address ARPA</i>	Creates a static ARP entry in the specified VRF.

Configuring Per-VRF for TACACS+ Servers

The per-VRF for TACACS+ servers feature enables you to configure per-virtual route forwarding (per-VRF) authentication, authorization, and accounting (AAA) on TACACS+ servers.

Before configuring per-VRF on a TACACS+ server, you must have configured AAA and a server group.

You can create the VRF routing table (shown in Steps 3 and 4) and configure the interface (Steps 6, 7, and 8). The actual configuration of per-VRF on a TACACS+ server is done in Steps 10 through 13.

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	Switch# configure terminal	Enters global configuration mode.
Step 3	Switch(config)# ip vrf <i>vrf-name</i>	Configures a VRF table and enters VRF configuration mode.
Step 4	Switch (config-vrf)# rd <i>route-distinguisher</i>	Creates routing and forwarding tables for a VRF instance.
Step 5	Switch (config-vrf)# exit	Exits VRF configuration mode.
Step 6	Switch (config)# interface <i>interface-name</i>	Configures an interface and enters interface configuration mode.
Step 7	Switch (config-if)# ip vrf forwarding <i>vrf-name</i>	Configures a VRF for the interface.
Step 8	Switch (config-if)# ip address <i>ip-address mask</i> [secondary]	Sets a primary or secondary IP address for an interface.
Step 9	Switch (config-if)# exit	Exits interface configuration mode.

	Command or Action	Purpose
Step 10	aaa group server tacacs+ <i>group-name</i> Example: Switch (config)# aaa group server tacacs+ tacacs1	Groups different TACACS+ server hosts into distinct lists and distinct methods and enters server-group configuration mode.
Step 11	server-private {<i>ip-address</i> <i>name</i>} [<i>nat</i>] [<i>single-connection</i>] [<i>port port-number</i>] [<i>timeout seconds</i>] [<i>key</i> [0 7] <i>string</i>] Example: Switch (config-sg-tacacs+)# server-private 10.1.1.1 port 19 key cisco	Configures the IP address of the private TACACS+ server for the group server.
Step 12	Switch (config-sg-tacacs+)# ip vrf forwarding <i>vrf-name</i>	Configures the VRF reference of a AAA TACACS+ server group.
Step 13	Switch (config-sg-tacacs+)# ip tacacs source-interface <i>subinterface-name</i>	Uses the IP address of a specified interface for all outgoing TACACS+ packets.
Step 14	Switch (config-sg-tacacs)# exit	Exits server-group configuration mode.

The following example lists all the steps to configure per-VRF TACACS+:

```
Switch> enable
Switch# configure terminal
Switch (config)# ip vrf cisco
Switch (config-vrf)# rd 100:1
Switch (config-vrf)# exit
Switch (config)# interface Loopback0
Switch (config-if)# ip vrf forwarding cisco
Switch (config-if)# ip address 10.0.0.2 255.0.0.0
Switch (config-if)# exit
Switch (config-sg-tacacs+)# ip vrf forwarding cisco
Switch (config-sg-tacacs+)# ip tacacs source-interface Loopback0
Switch (config-sg-tacacs)# exit
```

For more information about configuring per-VRF for TACACS+ server,

http://www.cisco.com/en/US/docs/ios/sec_user_services/configuration/guide/sec_vrf_tacas_svrs.pdf

Configuring Multicast VRFs

To configure multicast within a VRF table, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# ip routing	Enables IP routing.
Step 3	Switch(config)# ip vrf <i>vrf-name</i>	Names the VRF and enters VRF configuration mode.
Step 4	Switch(config-vrf)# ip multicast-routing vrf <i>vrf-name</i>	(Optional) Enables global multicast routing for VRF table.

	Command	Purpose
Step 5	Switch(config-vrf) # rd <i>route-distinguisher</i>	Creates a VRF table by specifying a route distinguisher. Enter either an AS number and an arbitrary number (xxx:y) or an IP address and arbitrary number (A.B.C.D:y).
Step 6	Switch(config-vrf) # route-target { export import both } <i>route-target-ext-community</i>	Creates a list of import, export, or import and export route target communities for the specified VRF. Enter either an AS system number and an arbitrary number (xxx:y) or an IP address and an arbitrary number (A.B.C.D:y). The <i>route-target-ext-community</i> value should be the same as the <i>route-distinguisher</i> value entered in Step 4.
Step 7	Switch(config-vrf) # import map <i>route-map</i>	(Optional) Associates a route map with the VRF.
Step 8	Switch(config-vrf) # interface <i>interface-id</i>	Enters interface configuration mode and specifies the Layer 3 interface to be associated with the VRF. The interface can be a routed port or a SVI.
Step 9	Switch(config-if) # ip vrf forwarding <i>vrf-name</i>	Associates the VRF with the Layer 3 interface.
Step 10	Switch(config-if) # ip address <i>ip-address mask</i>	Configures IP address for the Layer 3 interface.
Step 11	Switch(config-if) # ip pim [sparse-dense mode dense-mode sparse-mode]	Enables PIM on the VRF-associated Layer 3 interface.
Step 12	Switch(config-if) # end	Returns to privileged EXEC mode.
Step 13	Switch# show ip vrf [brief detail interfaces] [<i>vrf-name</i>]	Verifies the configuration. Display information about the configured VRFs.
Step 14	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

The following example shows how to configure multicast within a VRF table:

```
Switch(config)# ip routing
Switch(config)# ip vrf multiVrfA
Switch(config-vrf)# ip multicast-routing vrf multiVrfA
Switch(config-vrf)# interface GigabitEthernet3/1/0
Switch(config-if)# ip vrf forwarding multiVrfA
Switch(config-if)# ip address 172.21.200.203 255.255.255.0
Switch(config-if)# ip pim sparse-mode
```

For more information about configuring a multicast within a Multi-VRF CE, see the *Cisco IOS IP Multicast Configuration Guide, Release 12.4*.

Use the **no ip vrf vrf-name** global configuration command to delete a VRF and to remove all interfaces from it. Use the **no ip vrf forwarding** interface configuration command to remove an interface from the VRF.

Configuring a VPN Routing Session

Routing within the VPN can be configured with any supported routing protocol (RIP, OSPF, or BGP) or with static routing. The configuration shown here is for OSPF, but the process is the same for other protocols.

To configure OSPF in the VPN, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# router ospf <i>process-id vrf vrf-name</i>	Enables OSPF routing, specifies a VPN forwarding table, and enters router configuration mode.
Step 3	Switch(config-router)# log-adjacency-changes	(Optional) Logs changes in the adjacency state (the default state).
Step 4	Switch(config-router)# redistribute bgp autonomous-system-number subnets	Sets the switch to redistribute information from the BGP network to the OSPF network.
Step 5	Switch(config-router)# network <i>network-number area area-id</i>	Defines a network address and mask on which OSPF runs and the area ID for that network address.
Step 6	Switch(config-router)# end	Returns to privileged EXEC mode.
Step 7	Switch# show ip ospf process-id	Verifies the configuration of the OSPF network.
Step 8	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Use the **no router ospf process-id vrf vrf-name** global configuration command to disassociate the VPN forwarding table from the OSPF routing process.

The following example shows how to configure a single VRF named VRF-RED:

```
Switch(config)# ip vrf VRF-RED
Switch(config-vrf)# rd 1:1
Switch(config-vrf)# exit
Switch(config)# router eigrp virtual-name
Switch(config-router)# address-family ipv4 vrf VRF-RED autonomous-system 1
Switch(config-router-af)# network 10.0.0.0 0.0.0.255
Switch(config-router-af)# topology base
Switch(config-router-topology)# default-metric 10000 100 255 1 1500
Switch(config-router-topology)# exit-af-topology
Switch(config-router-af)# exit-address-family
```

Configuring BGP PE to CE Routing Sessions

To configure a BGP PE to CE routing session, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# router bgp <i>autonomous-system-number</i>	Configures the BGP routing process with the AS number passed to other BGP routers and enters router configuration mode.
Step 3	Switch(config-router)# network <i>network-number mask network-mask</i>	Specifies a network and mask to announce using BGP.
Step 4	Switch(config-router)# redistribute ospf process-id match internal	Sets the switch to redistribute OSPF internal routes.
Step 5	Switch(config-router)# network <i>network-number area area-id</i>	Defines a network address and mask on which OSPF runs and the area ID for that network address.

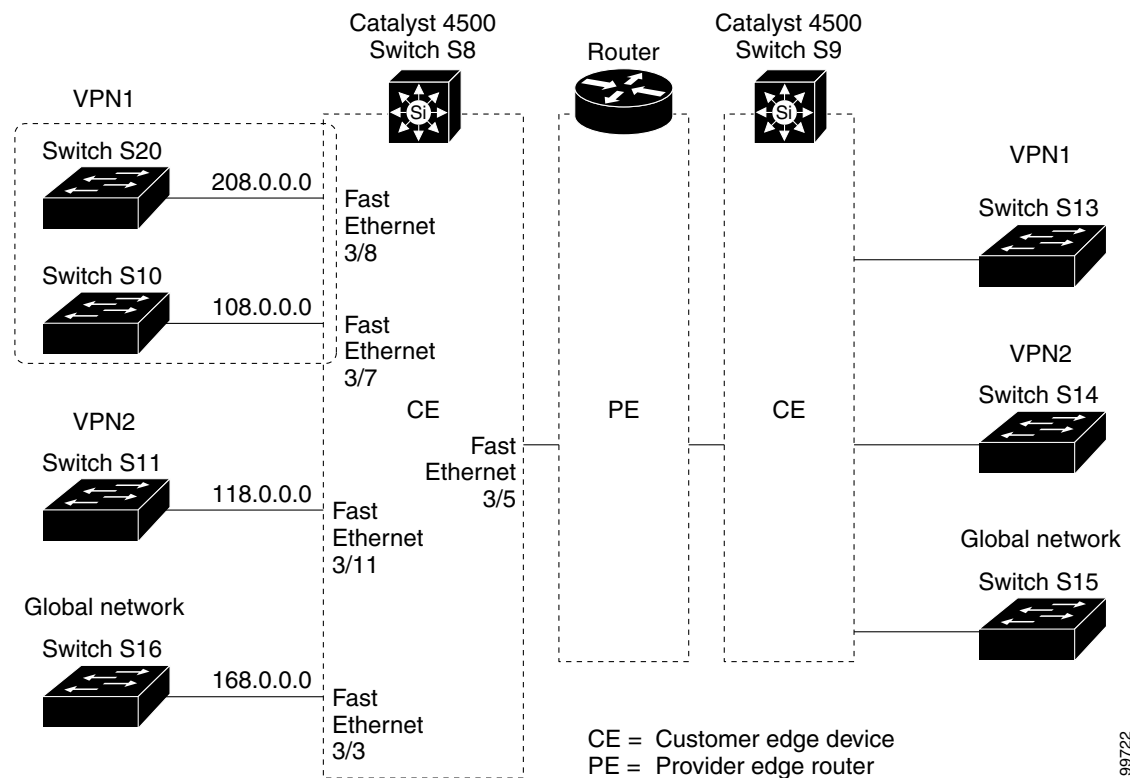
	Command	Purpose
Step 6	Switch(config-router-af)# address-family ipv4 vrf vrf-name	Defines BGP parameters for PE to CE routing sessions and enters VRF address-family mode.
Step 7	Switch(config-router-af)# neighbor <i>address remote-as as-number</i>	Defines a BGP session between PE and CE routers.
Step 8	Switch(config-router-af)# neighbor <i>address activate</i>	Activates the advertisement of the IPv4 address family.
Step 9	Switch(config-router-af)# end	Returns to privileged EXEC mode.
Step 10	Switch# show ip bgp [ipv4] [neighbors]	Verifies BGP configuration.
Step 11	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Use the **no router bgp autonomous-system-number** global configuration command to delete the BGP routing process. Use the command with keywords to delete routing characteristics.

VRF-lite Configuration Example

Figure 43-2 is a simplified example of the physical connections in a network similar to that in Figure 43-1. OSPF is the protocol used in VPN1, VPN2, and the global network. BGP is used in the CE to PE connections. The example commands show how to configure the CE switch S8 and include the VRF configuration for switches S20 and S11 and the PE router commands related to traffic with switch S8. Commands for configuring the other switches are not included but would be similar.

Figure 43-2 VRF-lite Configuration Example



99722

Configuring Switch S8

On switch S8, enable routing and configure VRF.

```
Switch# configure terminal  
Enter configuration commands, one per line. End with CNTL/Z.  
Switch(config)# ip routing  
Switch(config)# ip vrf v11  
Switch(config-vrf)# rd 800:1  
Switch(config-vrf)# route-target export 800:1  
Switch(config-vrf)# route-target import 800:1  
Switch(config-vrf)# exit  
Switch(config)# ip vrf v12  
Switch(config-vrf)# rd 800:2  
Switch(config-vrf)# route-target export 800:2  
Switch(config-vrf)# route-target import 800:2  
Switch(config-vrf)# exit
```

Configure the loopback and physical interfaces on switch S8. Fast Ethernet interface 3/5 is a trunk connection to the PE. Interfaces 3/7 and 3/11 connect to VPNs:

```
Switch(config)# interface loopback1  
Switch(config-if)# ip vrf forwarding v11  
Switch(config-if)# ip address 8.8.1.8 255.255.255.0  
Switch(config-if)# exit  
  
Switch(config)# interface loopback2  
Switch(config-if)# ip vrf forwarding v12  
Switch(config-if)# ip address 8.8.2.8 255.255.255.0  
Switch(config-if)# exit  
  
Switch(config)# interface FastEthernet3/5  
Switch(config-if)# switchport trunk encapsulation dot1q  
Switch(config-if)# switchport mode trunk  
Switch(config-if)# no ip address  
Switch(config-if)# exit  
  
Switch(config)# interface FastEthernet3/8  
Switch(config-if)# switchport access vlan 208  
Switch(config-if)# no ip address  
Switch(config-if)# exit  
  
Switch(config)# interface FastEthernet3/11  
Switch(config-if)# switchport trunk encapsulation dot1q  
Switch(config-if)# switchport mode trunk  
Switch(config-if)# no ip address  
Switch(config-if)# exit
```

Configure the VLANs used on switch S8. VLAN 10 is used by VRF 11 between the CE and the PE. VLAN 20 is used by VRF 12 between the CE and the PE. VLANs 118 and 208 are used for VRF for the VPNs that include switch S11 and switch S20, respectively:

```
Switch(config)# interface Vlan10  
Switch(config-if)# ip vrf forwarding v11  
Switch(config-if)# ip address 38.0.0.8 255.255.255.0  
Switch(config-if)# exit
```

```

Switch(config)# interface Vlan20
Switch(config-if)# ip vrf forwarding v12
Switch(config-if)# ip address 83.0.0.8 255.255.255.0
Switch(config-if)# exit

Switch(config)# interface Vlan118
Switch(config-if)# ip vrf forwarding v12
Switch(config-if)# ip address 118.0.0.8 255.255.255.0
Switch(config-if)# exit

Switch(config)# interface Vlan208
Switch(config-if)# ip vrf forwarding v11
Switch(config-if)# ip address 208.0.0.8 255.255.255.0
Switch(config-if)# exit

```

Configure OSPF routing in VPN1 and VPN2:

```

Switch(config)# router ospf 1 vrf v11
Switch(config-router)# redistribute bgp 800 subnets
Switch(config-router)# network 208.0.0.0 0.0.0.255 area 0
Switch(config-router)# exit
Switch(config)# router ospf 2 vrf v12
Switch(config-router)# redistribute bgp 800 subnets
Switch(config-router)# network 118.0.0.0 0.0.0.255 area 0
Switch(config-router)# exit

```

Configure BGP for CE to PE routing:

```

Switch(config)# router bgp 800
Switch(config-router)# address-family ipv4 vrf v12
Switch(config-router-af)# redistribute ospf 2 match internal
Switch(config-router-af)# neighbor 83.0.0.3 remote-as 100
Switch(config-router-af)# neighbor 83.0.0.3 activate
Switch(config-router-af)# network 8.8.2.0 mask 255.255.255.0
Switch(config-router-af)# exit

Switch(config-router)# address-family ipv4 vrf v11
Switch(config-router-af)# redistribute ospf 1 match internal
Switch(config-router-af)# neighbor 38.0.0.3 remote-as 100
Switch(config-router-af)# neighbor 38.0.0.3 activate
Switch(config-router-af)# network 8.8.1.0 mask 255.255.255.0
Switch(config-router-af)# end

```

Configuring Switch S20

Configure S20 to connect to CE:

```

Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# ip routing
Switch(config)# interface Fast Ethernet 0/7
Switch(config-if)# no switchport
Switch(config-if)# ip address 208.0.0.20 255.255.255.0
Switch(config-if)# exit

Switch(config)# router ospf 101
Switch(config-router)# network 208.0.0.0 0.0.0.255 area 0
Switch(config-router)# end

```

Configuring Switch S11

Configure S11 to connect to CE:

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# ip routing
Switch(config)# interface Gigabit Ethernet 0/3
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# no ip address
Switch(config-if)# exit

Switch(config)# interface Vlan118
Switch(config-if)# ip address 118.0.0.11 255.255.255.0
Switch(config-if)# exit

Switch(config)# router ospf 101
Switch(config-router)# network 118.0.0.0 0.0.0.255 area 0
Switch(config-router)# end
```

Configuring the PE Switch S3

On switch S3 (the router), these commands configure only the connections to switch S8:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip vrf v1
Router(config-vrf)# rd 100:1
Router(config-vrf)# route-target export 100:1
Router(config-vrf)# route-target import 100:1
Router(config-vrf)# exit

Router(config)# ip vrf v2
Router(config-vrf)# rd 100:2
Router(config-vrf)# route-target export 100:2
Router(config-vrf)# route-target import 100:2
Router(config-vrf)# exit

Router(config)# ip cef
Router(config)# interface Loopback1
Router(config-if)# ip vrf forwarding v1
Router(config-if)# ip address 3.3.1.3 255.255.255.0
Router(config-if)# exit

Router(config)# interface Loopback2
Router(config-if)# ip vrf forwarding v2
Router(config-if)# ip address 3.3.2.3 255.255.255.0
Router(config-if)# exit

Router(config)# interface Fast Ethernet3/0.10
Router(config-if)# encapsulation dot1q 10
Router(config-if)# ip vrf forwarding v1
Router(config-if)# ip address 38.0.0.3 255.255.255.0
Router(config-if)# exit

Router(config)# interface Fast Ethernet3/0.20
Router(config-if)# encapsulation dot1q 20
Router(config-if)# ip vrf forwarding v2
Router(config-if)# ip address 83.0.0.3 255.255.255.0
Router(config-if)# exit

Router(config)# router bgp 100
Router(config-router)# address-family ipv4 vrf v2
Router(config-router-af)# neighbor 83.0.0.8 remote-as 800
Router(config-router-af)# neighbor 83.0.0.8 activate
Router(config-router-af)# network 3.3.2.0 mask 255.255.255.0
```

```

Router(config-router-af)# exit
Router(config-router)# address-family ipv4 vrf vl
Router(config-router-af)# neighbor 83.0.0.8 remote-as 800
Router(config-router-af)# neighbor 83.0.0.8 activate
Router(config-router-af)# network 3.3.1.0 mask 255.255.255.0
Router(config-router-af)# end

```

Displaying VRF-lite Status

To display information about VRF-lite configuration and status, perform one of the following tasks:

Command	Purpose
Switch# show ip protocols vrf <i>vrf-name</i>	Displays routing protocol information associated with a VRF.
Switch# show ip route vrf <i>vrf-name</i> [connected] [<i>protocol</i> [<i>as-number</i>]] [list] [mobile] [odr] [profile] [static] [summary] [supernets-only]	Displays IP routing table information associated with a VRF.
Switch# show ip vrf [brief detail interfaces] [<i>vrf-name</i>]	Displays information about the defined VRF instances.
Switch# show ip mroute vrf <i>instance-name</i> <i>a.b.c.d</i> active bidirectional count dense interface proxy pruned sparse ssm static summary	Displays information about the defined VRF instances.

This example shows how to display multicast route table information within a VRF instance:

```

Switch# show ip mroute vrf mcast2 234.34.10.18
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
       U - URD, I - Received Source Specific Host Report,
       Z - Multicast Tunnel, z - MDT-data group sender,
       Y - Joined MDT-data group, y - Sending to MDT-data group
       V - RD & Vector, v - Vector
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 234.34.10.18), 13:39:21/00:02:58, RP 10.1.1.1, flags: BC
Bidir-Upstream: Vlan134, RPF nbr 172.16.34.1
Outgoing interface list:
  Vlan45, Forward/Sparse-Dense, 00:00:02/00:02:57, H
  Vlan134, Bidir-Upstream/Sparse-Dense, 13:35:54/00:00:00, H

```



Note

For more information about the information in the displays, refer to the *Cisco IOS Switching Services Command Reference* at:

http://www.cisco.com/en/US/docs/ios/ipswitch/command/reference/isw_book.html

Configuring VRF-lite for IPv6

Configuring VRF-Aware Services

IP services can be configured on global interfaces and within the global routing instance. IP services are enhanced to run on multiple routing instances; they are VRF-aware. Any configured VRF in the system can be specified for a VRF-aware service.

VRF-aware services are implemented in platform-independent modules. VRF provides multiple routing instances in Cisco IOS. Each platform has its own limit on the number of VRFs it supports.

VRF-aware services have the following characteristics:

- The user can ping a host in a user-specified VRF.
- ARP entries are learned in separate VRFs. The user can display Address Resolution Protocol (ARP) entries for specific VRFs.

These services are VRF-aware:

- Ping
- Unicast Reverse Path Forwarding (uRPF)
- Traceroute
- FTP and TFTP
- Telnet and SSH
- NTP

Configuring the User Interface for ARP

To configure VRF-aware services for ARP, perform this task:

Command	Purpose
Switch# show ip arp vrf <i>vrf-name</i>	Displays the ARP table (static and dynamic entries) in the specified VRF.
Switch(config)# arp vrf <i>vrf-name</i> <i>ip-address mac-address ARPA</i>	Creates a static ARP entry in the specified VRF.

Configuring the User Interface for PING

To perform a VRF-aware ping, perform this task:

Command	Purpose
Switch# ping vrf <i>vrf-name ip-host</i>	Pings an IP host or address in the specified VRF.

Configuring the User Interface for uRPF

You can configure uRPF on an interface assigned to a VRF. Source lookup is performed in the VRF table.

To configure VRF-aware services for uRPF, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Enters interface configuration mode and specifies the Layer 3 interface to configure.
Step 3	Switch(config-if)# no switchport	Removes the interface from Layer 2 configuration mode if it is a physical interface.
Step 4	Switch(config-if)# ip vrf forwarding <i>vrf-name</i>	Configures VRF on the interface.
Step 5	Switch(config-if-vrf)# ip address <i>ip-address subnet-mask</i>	Enters the IP address for the interface.
Step 6	Switch(config-if-vrf)# ip verify unicast source reachable-via rx allow-default	Enables uRPF on the interface.
Step 7	Switch(config-if-vrf)# end	Returns to privileged EXEC mode.

Configuring the User Interface for Traceroute

To configure VRF-aware services for traceroute, perform this task:

Command	Purpose
traceroute vrf <i>vrf-name ipaddress</i>	Specifies the name of a VPN VRF in which to find the destination address.

Configuring the User Interface for FTP and TFTP

You must configure some FTP and TFTP CLIs in order for FTP and TFTP to be VRF-aware. For example, if you want to use a VRF table that is attached to an interface (for example, E1/0), you need to configure the **ip [t]ftp source-interface E1/0** command to inform [t]ftp to use a specific routing table. In this example, the VRF table is used to look up the destination IP address. These changes are backward-compatible and do not affect existing behavior. You can use the source-interface CLI to send packets out a particular interface even if no VRF is configured on that interface.

To specify the source IP address for FTP connections, use the **ip ftp source-interface** show mode command. To use the address of the interface where the connection is made, use the **no** form of this command.

To configure the user interface for FTP and TFTP, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.

	Command	Purpose
Step 2	Switch(config)# ip ftp source-interface <i>interface-type interface-number</i>	Specifies the source IP address for FTP connections.
Step 3	Switch(config)# end	Returns to privileged EXEC mode.

To specify the IP address of an interface as the source address for TFTP connections, use the **ip tftp source-interface** show mode command. To return to the default, use the **no** form of this command.

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# ip tftp source-interface <i>interface-type interface-number</i>	Specifies the source IP address for TFTP connections.
Step 3	Switch(config)# end	Returns to privileged EXEC mode.

Configuring the User Interface for Telnet and SSH

To configure VRF-aware for using Telnet and SSH, perform this task:

Command	Purpose
Switch# telnet <i>ip-address/vrf vrf-name</i>	Connects through Telnet to an IP host or address in the specified VRF.
Switch# ssh -l username -vrf vrf-name ip-host	Connects through SSH to an IP host or address in the specified VRF.

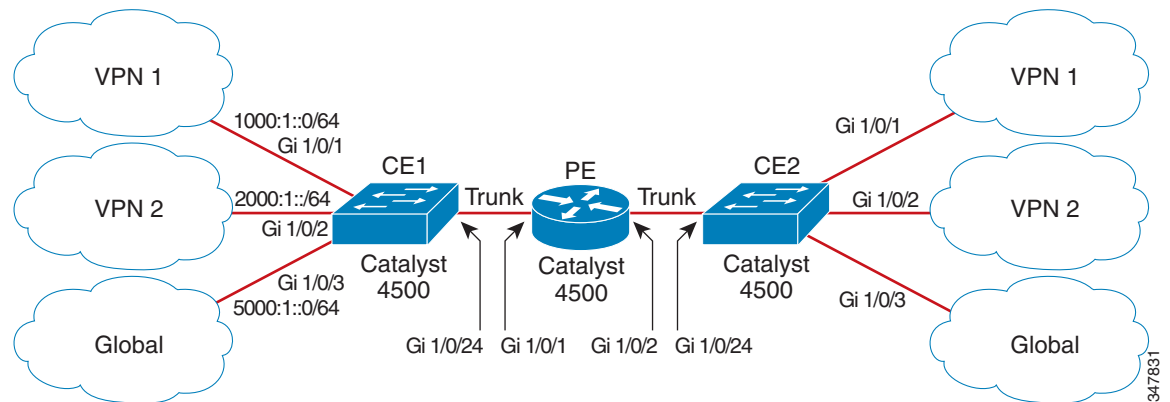
Configuring the User Interface for NTP

To configure VRF-aware for NTP, perform this task:

Command	Purpose
Switch# ntp server vrf vrf-name ip-host	Configure the NTP server in the specified VRF.
Switch# ntp peer vrf vrf-name ip-host	Configure the NTP peer in the specified VRF.

VRF-lite Configuration Example

The following topology illustrates how to use OSPFv3 for CE-PE routing.

Figure 43-3 VRF-lite Configuration Example

Configuring CE1 Switch

```

ipv6 unicast-routing
vrf definition v1
  rd 100:1
  !
  address-family ipv6
    exit-address-family
  !

vrf definition v2
  rd 200:1
  !
  address-family ipv6
    exit-address-family
  !

interface Vlan100
  vrf forwarding v1
  no ip address
  ipv6 address 1000::1::1/64
  ospfv3 100 ipv6 area 0
  !

interface Vlan200
  vrf forwarding v2
  no ip address
  ipv6 address 2000::1::1/64
  ospfv3 200 ipv6 area 0
  !

interface GigabitEthernet 1/0/1
  switchport access vlan 100
  end

interface GigabitEthernet 1/0/2
  switchport access vlan 200
  end

interface GigabitEthernet 1/0/24
  switchport trunk encapsulation dot1q

switchport mode trunk
no ip address

```



```
end

router ospfv3 100
router-id 10.10.10.10
!
address-family ipv6 unicast vrf v1
redistribute connected
area 0 normal
exit-address-family
!

router ospfv3 200
router-id 20.20.20.20
!
address-family ipv6 unicast vrf v2
redistribute connected
area 0 normal
exit-address-family
!
```

Configuring PE Switch

```
ipv6 unicast-routing

vrf definition v1
rd 100:1
!
address-family ipv6
exit-address-family
!

vrf definition v2
rd 200:1
!
address-family ipv6
exit-address-family
!

interface Vlan600
vrf forwarding v1
no ip address
ipv6 address 1000:1::2/64
ospfv3 100 ipv6 area 0
!

interface Vlan700
vrf forwarding v2
no ip address
ipv6 address 2000:1::2/64
ospfv3 200 ipv6 area 0
!

interface Vlan800
vrf forwarding v1
no ip address
ipv6 address 3000:1::7/64
ospfv3 100 ipv6 area 0
!

interface Vlan900
vrf forwarding v2
no ip address
ipv6 address 4000:1::7/64
ospfv3 200 ipv6 area 0
!
```

```

interface GigabitEthernet 1/0/1
  switchport trunk encapsulation dot1q
  switchport mode trunk
  no ip address
  exit

interface GigabitEthernet 1/0/2
  switchport trunk encapsulation dot1q

switchport mode trunk
  no ip address
  exit

router ospfv3 100
  router-id 30.30.30.30
  !
  address-family ipv6 unicast vrf v1
    redistribute connected
    area 0 normal
  exit-address-family
  !
  address-family ipv6 unicast vrf v2
    redistribute connected
    area 0 normal
  exit-address-family
  !

```

Configuring CE2 Switch

```

ipv6 unicast-routing

vrf definition v1
  rd 100:1
  !
  address-family ipv6
    exit-address-family
  !

vrf definition v2
  rd 200:1
  !
  address-family ipv6
    exit-address-family
  !

interface Vlan100
  vrf forwarding v1
  no ip address

ipv6 address 1000:1::3/64
ospfv3 100 ipv6 area 0
!

interface Vlan200
  vrf forwarding v2
  no ip address
  ipv6 address 2000:1::3/64
  ospfv3 200 ipv6 area 0
!

interface GigabitEthernet 1/0/1
  switchport access vlan 100

```

```
end

interface GigabitEthernet 1/0/2
switchport access vlan 200
end

interface GigabitEthernet 1/0/24
switchport trunk encapsulation dot1q
switchport mode trunk
end

router ospfv3 100
router-id 40.40.40.40
!
address-family ipv6 unicast vrf v1
redistribute connected
area 0 normal
exit-address-family
!

router ospfv3 200
router-id 50.50.50.50
!
address-family ipv6 unicast vrf v2
redistribute connected

area 0 normal
exit-address-family
!
```

Displaying VRF-lite Status

To display information about VRF-lite configuration and status, perform one of the following tasks:

**Note**

For more information about the information in the displays, refer to the *Cisco IOS Switching Services Command Reference* at:

http://www.cisco.com/en/US/docs/ios/ipswitch/command/reference/isw_book.html

To display information about VRF-lite configuration and status, perform one of the following tasks:

Command	Purpose
Switch# show ipv6 route vrf a [X:X:X:X::X/<0-128>] [bgp] [connected] [eigrp] [interface] [isis] [local] [nd] [nsf] [ospf] [repair] [rip] [shortcut] [static] [summary] [tag] [updated] [watch]	Displays routing protocol information associated with a VRF. X:X:X:X::X/<0-128> IPv6 prefix bgp BGP routes connected Connected routes eigrp EIGRP routes interface interface specific routes isis IS-IS routes local Local routes nd ND routes nsf non stop forwarding state ospf OSPFv3 routes repair Routes with Repair paths rip RIPng routes shortcut Routes with Shortcut paths static Static routes summary Summary display tag Route Tag updated Show routes with timestamps watch route watchers
Switch# show ipv6 vrf [brief detail interfaces] [vrf-name]	Displays information about the defined VRF instances. brief Brief VPN Routing/Forwarding instance information detail Detailed VPN Routing/Forwarding instance information interfaces Show VPN Routing/Forwarding interface information

When you configure VRF table “a” with the IPv6 address family and attach the VRF to the interface with IPv6 address 1::2/64, the **show ipv6 route vrf a** command displays the following output:

```
Switch# show ipv6 route vrf a
IPv6 Routing Table - a - 3 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
        B - BGP, R - RIP, I1 - ISIS L1, I2 - ISIS L2
        IA - ISIS interarea, IS - ISIS summary, D - EIGRP, EX - EIGRP external
        ND - ND Default, NDp - ND Prefix, DCE - Destination, NDr - Redirect
        O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
        ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
C    1::/64 [0/0]
    via GigabitEthernet7/1, directly connected
L    1::2/128 [0/0]
    via GigabitEthernet7/1, receive
L    FF00::/8 [0/0]
    via Null0, receive
Switch#
```

For further examples, refer to

http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_16.html

Configuring IPv6 VRF-lite

Beginning with Release IOS XE 3.5.0E and IOS 15.2(1)E, to support IPv6 VRF-lite, we transition from the **ip vrf** command to the “new” **vrf definition** command.

Configure VRFs

To configure one or more VRFs, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# ipv6 routing	Enables IPv6 routing.
Step 3	Switch(config)# vrf definition <i>vrf-name</i>	Names the VRF and enters VRF configuration mode.
Step 4	Switch(config-vrf)# address-family <i>ipv4 ipv6</i>	(Optional) IPv4 by default. Configuration MUST for ipv6.
Step 5	Switch(config-vrf)# rd <i>route-distinguisher</i>	(Optional) Creates a VRF table by specifying a route distinguisher. Enter either an Autonomous System number and an arbitrary number (xxx:y) or an IP address and arbitrary number (A.B.C.D:y).
Step 6	Switch(config-vrf)# route-target { export import both } <i>route-target-ext-community</i>	(Optional) Creates a list of import, export, or import and export route target communities for the specified VRF. Enter either an AS system number and an arbitrary number (xxx:y) or an IP address and an arbitrary number (A.B.C.D:y). Note This command is effective only if BGP is running.
Step 7	Switch(config-vrf)# import map <i>route-map</i>	(Optional) Associates a route map with the VRF.
Step 8	Switch(config-vrf)# interface <i>interface-id</i>	Enters interface configuration mode and specify the Layer 3 interface to be associated with the VRF. The interface can be a routed port or SVI.
Step 9	Switch(config-if)# ip vrf forwarding <i>vrf-name</i>	Associates the VRF with the Layer 3 interface.
Step 10	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 11	Switch# show ip vrf [brief detail interfaces] [<i>vrf-name</i>]	Verifies the configuration. Displays information about the configured VRFs.
Step 12	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to configure VRFs:

```
Switch(config)# vrf definition red
Switch(config-vrf)# rd 100:1
Switch(config-vrf)# address family ipv6
Switch(config-vrf-af)# route-target both 200:1
Switch(config-vrf)# exit-address-family
Switch(config-vrf)# interface Ethernet0/1
Switch(config-if)# vrf forwarding red
Switch(config-if)# ipv6 address 5000::72B/64
```

Associate Interfaces to the Defined VRFs

To associate interface to the defined VRFs, perform this task:

	Command	Purpose
Step 1	Switch(config)# vrf configuration	Enters vrf configuration mode.
Step 1	Switch(config-vrf)# interface <i>interface-id</i>	Enters interface configuration mode and specifies the Layer 3 interface to be associated with the VRF. The interface can be a routed port or SVI.
Step 2	Switch(config-if)# vrf forwarding <i>vrf-name</i>	Associates the VRF with the Layer 3 interface.
Step 3	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 4	Switch# show ipv6 vrf [brief detail interfaces] [<i>vrf-name</i>]	Verifies the configuration. Displays information about the configured VRFs.
Step 5	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to associate an interface to VRFs:

```
Switch(config-vrf)# interface ethernet0/1
Switch(config-if)# vrf forwarding red
Switch(config-if)# ipv6 address 5000::72B/64
```

Populate VRF with Routes via Routing Protocols

Static Route

	Command	Purpose
Step 1	ipv6 route [vrf <i>vrf-name</i>] <i>ipv6-prefix/prefix-length</i> { <i>ipv6-address</i> interface-type <i>interface-number</i> [<i>ipv6-address</i>]}	To configure static routes specific to VRF.

This example shows how to populate VRF with a static route:

```
Switch(config)# ipv6 route vrf v6a 7000::/64 TenGigabitEthernet3/2 4000::2
```

Routing Protocols

OSPFv3

To configure the OSPFv3 router process and the IPv6 address family in OSPFv3, perform the following steps:

	Command	Purpose
Step 1	Switch> enable	Enters privileged EXEC mode. Enter your password if prompted.
Step 2	Switch# configure terminal	Enters global configuration mode.
Step 3	Switch(config)# router ospfv3 <i>process-id</i>	Enables OSPFv3 router configuration mode for the IPv4 or IPv6 address family.
Step 4	Switch(config-router)# area <i>area-ID</i> [default-cot nssa stub]	Configures the OSPFv3 area.
Step 5	Switch(config-router)# router-id <i>router-id</i>	Use a fixed router ID.
Step 6	Switch(config-router)# address-family ipv6 unicast vrf <i>vrf-name</i> <i>Or</i> Switch(config-router)# address-family ipv4 unicast	Enters IPv6 address family configuration mode for OSPFv3 in VRF <i>vrf-name</i> or Enters IPv4 address family configuration mode for OSPFv3.
Step 7	Switch(config-router)# redistribute source-protocol [<i>process-id</i>] options	Redistributes IPv6 and IPv4 routes from one routing domain into another routing domain.
Step 8	Switch(config-router)# end	Returns to privileged EXEC mode.

This example shows how configure the OSPFv3 router process:

```
Switch(config-router)# router ospfv3 1
Switch(config-router)# router-id 10.1.1.1
Switch(config-router)# address-family ipv6 unicast
Switch(config-router-af)# exit-address-family
```

To enable OSPFv3 on an interface, do the following:

	Command	Purpose
Step 1	Switch> enable	Enters privileged EXEC mode. Enter your password if prompted.
Step 2	Switch# configure terminal	Enters global configuration mode.
Step 3	Switch(config)# interface <i>type-number</i>	Specifies an interface type and number, and places the switch in interface configuration mode.

	Command	Purpose
Step 4	Switch(config-if)# ospfv3 <i>process-id</i> area <i>area-ID</i> { ipv4 ipv6 } [instance <i>instance-id</i>]	Enables OSPFv3 on an interface with the IPv4 or IPv6 AF.
	Or Switch(config-if)# ipv6 ospf process-id <i>area</i> <i>area-ID</i> [instance <i>instance-id</i>]	or Enables OSPFv3 on an interface.
Step 5	Switch(config-if)# end	Returns to privileged EXEC mode.

This example show how to enable OSPFv3 on an interface:

```
Switch(config)# interface GigabitEthernet2/1
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address 4000::2/64
Switch(config-if)# ipv6 enable
Switch(config-if)# ipv6 ospf 1 area 0
Switch(config-if)# end
```

EIGRP

To configure an EIGRPv6 routing process, perform the following steps:

	Command	Purpose
Step 1	Switch> enable	Enters privileged EXEC mode. Enter your password if prompted.
Step 2	Switch# configure terminal	Enters global configuration mode.
Step 3	Switch(config)# router eigrp <i>virtual-instance-name</i>	Configures the EIGRP routing process and enters router configuration mode.
Step 4	Switch(config-router)# address-family ipv6 vrf <i>vrf-name</i> autonomous-system <i>autonomous-system-number</i>	Enables EIGRP IPv6 VRF-Lite and enters address family configuration mode.
Step 5	Switch(config-router-af)# topology { base topology-name <i>tid</i> <i>number</i> }	Configures an EIGRP process to route IP traffic under the specified topology instance and enters address family topology configuration mode.
Step 6	Switch(config-router-af-topology)# exit-af-topology	Exits address family topology configuration mode.
Step 7	Switch(config-router)# eigrp router-id <i>ip-address</i>	Enables the use of a fixed router-id.
Step 8	Switch(config-router)# end	Exits router configuration mode.

This example shows how to configure an EIGRP routing process:

```
Switch(config)# router eigrp test
Switch(config-router)# address-family ipv6 unicast vrf b1 autonomous-system 10
Switch(config-router-af)# topology base
Switch(config-router-af-topology)# exit-af-topology
Switch(config-router)# eigrp router-id 2.3.4.5
Switch(config-router)# exit-address-family
```


EBGPv6

To configure EBGPv6, do the following:

	Command	Purpose
Step 1	Switch> enable	Enters privileged EXEC mode. Enter your password if prompted.
Step 2	Switch# configure terminal	Enters global configuration mode.
Step 3	Switch(config)# router bgp <i>as-number</i>	Enters router configuration mode for the specified routing process.
Step 4	Switch(config-router)# neighbor <i>peer-group-name peer-group</i>	Creates a multiprotocol BGP peer group.
Step 5	Switch(config-router)# neighbor <i>{ip-address ipv6-address[%] peer-group-name}</i> remote-as <i>autonomous-system-number</i> [alternate-as <i>autonomous-system-number ...</i>]	Adds the IPv6 address of the neighbor in the specified autonomous system to the IPv6 multiprotocol BGP neighbor table of the local router.
Step 6	Switch(config-router)# address-family ipv6 [vrf <i>vrf-name</i>] [unicast multicast vpnv6]	Specifies the IPv6 address family, and enters address family configuration mode. <ul style="list-style-type: none"> The unicast keyword specifies the IPv6 unicast address family. By default, the switch is placed in configuration mode for the IPv6 unicast address family if the unicast keyword is not specified with the address-family ipv6 command. The multicast keyword specifies IPv6 multicast address prefixes.
Step 7	Switch(config-router-af)# neighbor <i>ipv6-address peer-group</i> <i>peer-group-name</i>	Assigns the IPv6 address of a BGP neighbor to a peer group.
Step 8	Switch(config-router-af)# neighbor <i>{ip-address peer-group-name ipv6-address[%]}</i> route-map <i>map-name</i> [in out]	Applies a route map to incoming or outgoing routes. <ul style="list-style-type: none"> Changes to the route map will not take effect for existing peers until the peering is reset or a soft reset is performed. Using the clear bgp ipv6 command with the soft and in keywords will perform a soft reset.
Step 9	Switch(config-router-af)# exit	Exits address family configuration mode, and returns the router to router configuration mode.

This example shows how to configure EBRPv6:

```
Switch(config)# router bgp 2
Switch(config-router)# bgp router-id 2.2.2.2
Switch(config-router)# bgp log-neighbor-changes
Switch(config-router)# no bgp default ipv4-unicast
Switch(config-router)# neighbor 2500::1 remote-as 1
Switch(config-router)# neighbor 4000::2 remote-as 3
Switch(config-router)# address-family ipv6 vrf b1
Switch(config-router-af)# network 2500::/64
Switch(config-router-af)# network 4000::/64
Switch(config-router-af)# neighbor 2500::1 remote-as 1
Switch(config-router-af)# neighbor 2500::1 activate
Switch(config-router-af)# neighbor 4000::2 remote-as 3
```

```
Switch(config-router-af)# neighbor 4000::2 activate
Switch(config-router-af)# exit-address-family
```

VPN Co-existence Between IPv4 and IPv6

With Release IOS XE 3.5.0E and IOS 15.2(1)E, we provide backward compatibility between the “older” CLI for configuring IPv4 and the “new” CLI for IPv6. This means that a configuration might contain both CLI. The IPv4 CLI retains the ability to have on the same interface, an IP address defined within a VRF as well as an IPv6 address defined in the global routing table.

For example:

```
vrf definition red
rd 100:1
address family ipv6
route-target both 200:1
exit-address-family
!
ip vrf blue
rd 200:1
route-target both 200:1
!
interface Ethernet0/0
vrf forwarding red
ip address 50.1.1.2 255.255.255.0
ipv6 address 4000::72B/64
!
interface Ethernet0/1
ip vrf forwarding blue
ip address 60.1.1.2 255.255.255.0
ipv6 address 5000::72B/64
```

In this example, all addresses (v4 and v6) defined for Ethernet0/0 refer to VRF red whereas for Ethernet0/1, the IP address refers to VRF blue but the ipv6 address refers to the global IPv6 routing table.

Migrating from the Old to New CLI Scheme

Prior to Release IOS XE 3.5.0E and IOS 15.2(1)E, you used the **ip vrf** command to configure vrf. With Release IOS XE 3.5.0E and IOS 15.2(1)E, you use the new **vrf definition** command.

Henceforward, to incorporate IPv6 VRF configurations in addition to IPv4 configurations, you must *migrate* from the prior VRF CLI scheme using the following command:

```
Switch(config)# vrf upgrade-cli multi-af-mode {common-policies | non-common-policies} [vrf name]
```

This command forces migration from *old* CLI for IPv4 VRF to the *new* VRF multi-AF CLI. It is not nvgen'd because the effect is “one-time” only (see BGP similar command " bgp upgrade-cli ").



Configuring Quality of Service

This chapter describes how to configure quality of service (QoS) with either automatic QoS (auto-QoS) commands or standard QoS commands on a Catalyst 4500 series switch. It describes how to specify QoS configuration on different types of interfaces (access, Layer 2 trunk, Layer 3 routed, Etherchannel) as well as VLANs. It also describes how to specify different QoS configurations on different VLANs on a given interface (per-port per-VLAN QoS).

A switch supports a QoS configuration model known as *MQC* (Modular QoS CLI). Please refer to the appropriate configuration section for the supervisor engine on which QoS will be configured. For more information about MQC, see the “Modular Quality of Service Command-Line Interface” section of the *Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.3*.

This chapter addresses both VSS and non-VSS environments. Sections include:

- [Overview of QoS, page 44-1](#)
- [Configuring VSS QoS, page 44-14](#)
- [Configuring QoS on a Standalone Supervisor Engine 6-E, 6L-E or Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, and 7-E, page 44-48](#)
- [Configuring VSS Auto-QoS, page 44-82](#)
- [Configuring Auto-QoS on a Standalone Supervisor Engine 6-E/6L-E or Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, and 7-E, page 44-89](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

Overview of QoS

Typically, networks operate on a *best-effort* delivery basis, which means that all traffic has equal priority and an equal chance of being delivered in a timely manner. When congestion occurs, all traffic has an equal chance of being dropped.

QoS selects network traffic (both unicast and multicast), prioritizes it according to its relative importance, and uses congestion avoidance to provide priority-indexed treatment; QoS can also limit the bandwidth used by network traffic. QoS can make network performance more predictable and bandwidth utilization more effective.

This section contains the following subsections:

- [Prioritization, page 44-2](#)
- [QoS Terminology, page 44-3](#)
- [Basic QoS Model, page 44-5](#)
- [Classification, page 44-6](#)
- [Policing and Marking, page 44-8](#)
- [Queueing and Scheduling, page 44-8](#)
- [Packet Modification, page 44-9](#)
- [Per Port Per VLAN QoS, page 44-10](#)
- [Flow-based QoS, page 44-10](#)
- [Using Metadata in QoS Policy, page 44-11](#)
- [Configuring System Queue Limit, page 44-12](#)
- [Configuring QoS with a 40-Gigabit Ethernet Interface, page 44-13](#)

Prioritization

QoS implementation is based on the DiffServ architecture. This architecture specifies that each packet is classified upon entry into the network. The classification is carried in the IP packet header, using 6 bits from the deprecated IP type of service (TOS) field to carry the classification (*class*) information. Classification can also be carried in the Layer 2 frame. These special bits in the Layer 2 frame or a Layer 3 packet are described here and shown in [Figure 44-1](#):

- Prioritization values in Layer 2 frames:

Layer 2 Inter-Switch Link (ISL) frame headers have a 1-byte User field that carries an IEEE 802.1p class of service (CoS) value in the three least-significant bits. On interfaces configured as Layer 2 ISL trunks, all traffic is in ISL frames.

Layer 2 802.1Q frame headers have a 2-byte Tag Control Information field that carries the CoS value in the three most-significant bits, which are called the User Priority bits. On interfaces configured as Layer 2 802.1Q trunks, all traffic is in 802.1Q frames except for traffic in the native VLAN.

Other frame types cannot carry Layer 2 CoS values.

Layer 2 CoS values range from 0 for low priority to 7 for high priority.

- Prioritization bits in Layer 3 packets:

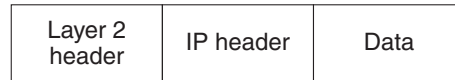
Layer 3 IP packets can carry either an IP precedence value or a Differentiated Services Code Point (DSCP) value. QoS supports the use of either value because DSCP values are backward-compatible with IP precedence values.

IP precedence values range from 0 to 7.

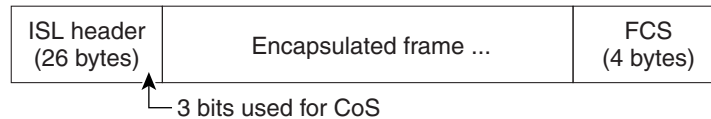
DSCP values range from 0 to 63.

Figure 44-1 QoS Classification Layers in Frames and Packets

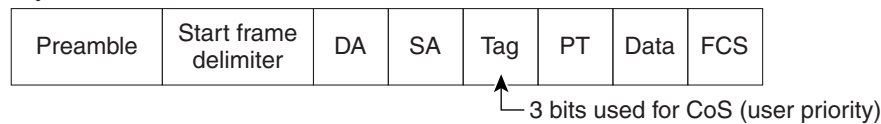
Encapsulated Packet



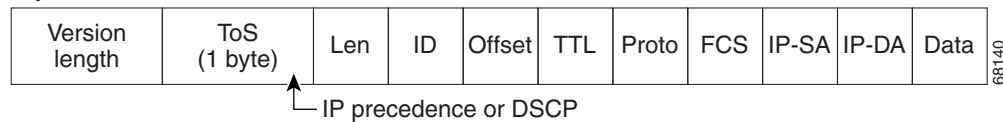
Layer 2 ISL Frame



Layer 2 802.1Q/P Frame



Layer 3 IPv4 Packet



All switches and routers across the Internet rely on the class information to provide the same forwarding treatment to packets with the same class information and different treatment to packets with different class information. The class information in the packet can be assigned by end hosts or by switches or routers along the way, based on a configured policy, detailed examination of the packet, or both. Detailed examination of the packet is expected to happen closer to the edge of the network so that the core switches and routers are not overloaded.

Switches and routers along the path can use the class information to limit the amount of resources allocated per traffic class. The behavior of an individual device when handling traffic in the DiffServ architecture is called per-hop behavior. If all devices along a path provide a consistent per-hop behavior, you can construct an end-to-end QoS solution.

Implementing QoS in your network can be a simple or complex task and depends on the QoS features offered by your internetworking devices, the traffic types and patterns in your network, and the granularity of control you need over incoming and outgoing traffic.

QoS Terminology

The following terms are used when discussing QoS features:

- *Packets* carry traffic at Layer 3.
- *Frames* carry traffic at Layer 2. Layer 2 frames carry Layer 3 packets.
- *Labels* are prioritization values carried in Layer 3 packets and Layer 2 frames:
 - Layer 2 class of service (CoS) values, which range between zero for low priority and seven for high priority:

Layer 2 Inter-Switch Link (ISL) frame headers have a 1-byte User field that carries an IEEE 802.1p CoS value in the three least significant bits.

Layer 2 802.1Q frame headers have a 2-byte Tag Control Information field that carries the CoS value in the three most significant bits, which are called the User Priority bits.

Other frame types cannot carry Layer 2 CoS values.

**Note**

On interfaces configured as Layer 2 ISL trunks, all traffic is in ISL frames. On interfaces configured as Layer 2 802.1Q trunks, all traffic is in 802.1Q frames except for traffic in the native VLAN.

- Layer 3 IP precedence values—The IP version 4 specification defines the three most significant bits of the 1-byte ToS field as IP precedence. IP precedence values range between zero for low priority and seven for high priority.
- Layer 3 differentiated services code point (DSCP) values—The Internet Engineering Task Force (IETF) has defined the six most significant bits of the 1-byte IP ToS field as the DSCP. The per-hop behavior represented by a particular DSCP value is configurable. DSCP values range between 0 and 63.

**Note**

Layer 3 IP packets can carry either an IP precedence value or a DSCP value. QoS supports the use of either value, since DSCP values are backwards compatible with IP precedence values. See [Table 44-1](#).

Table 44-1 IP Precedence and DSCP Values

3-bit IP Precedence	6 MSb ¹ of ToS						6-bit DSCP		3-bit IP Precedence	6 MSb ¹ of ToS						6-bit DSCP
	8	7	6	5	4	3				8	7	6	5	4	3	
0	0	0	0	0	0	0	0		4	1	0	0	0	0	0	32
	0	0	0	0	0	1	1			1	0	0	0	0	1	33
	0	0	0	0	1	0	2			1	0	0	0	1	0	34
	0	0	0	0	1	1	3			1	0	0	0	1	1	35
	0	0	0	1	0	0	4			1	0	0	1	0	0	36
	0	0	0	1	0	1	5			1	0	0	1	0	1	37
	0	0	0	1	1	0	6			1	0	0	1	1	0	38
	0	0	0	1	1	1	7			1	0	0	1	1	1	39
1	0	0	1	0	0	0	8		5	1	0	1	0	0	0	40
	0	0	1	0	0	1	9			1	0	1	0	0	1	41
	0	0	1	0	1	0	10			1	0	1	0	1	0	42
	0	0	1	0	1	1	11			1	0	1	0	1	1	43
	0	0	1	1	0	0	12			1	0	1	1	0	0	44
	0	0	1	1	0	1	13			1	0	1	1	0	1	45
	0	0	1	1	1	0	14			1	0	1	1	1	0	46
	0	0	1	1	1	1	15			1	0	1	1	1	1	47

Table 44-1 IP Precedence and DSCP Values (continued)

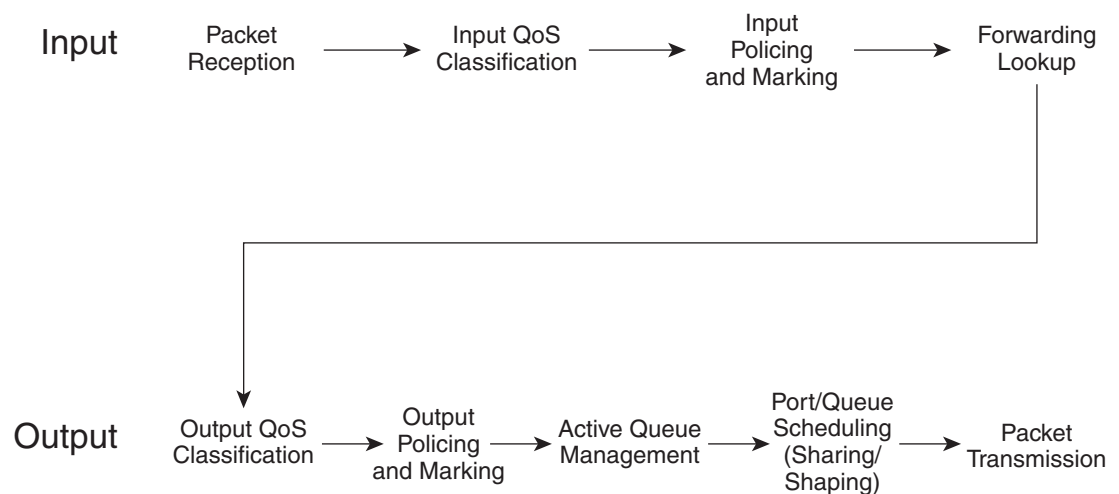
3-bit IP Precedence	6 MSb ¹ of ToS						6-bit DSCP		3-bit IP Precedence	6 MSb ¹ of ToS						6-bit DSCP		
	8	7	6	5	4	3				8	7	6	5	4	3			
2	0	1	0		0	0	0	16		6	1	1	0		0	0	0	48
	0	1	0		0	0	1	17			1	1	0		0	0	1	49
	0	1	0		0	1	0	18			1	1	0		0	1	0	50
	0	1	0		0	1	1	19			1	1	0		0	1	1	51
	0	1	0		1	0	0	20			1	1	0		1	0	0	52
	0	1	0		1	0	1	21			1	1	0		1	0	1	53
	0	1	0		1	1	0	22			1	1	0		1	1	0	54
	0	1	0		1	1	1	23			1	1	0		1	1	1	55
3	0	1	1		0	0	0	24		7	1	1	1		0	0	0	56
	0	1	1		0	0	1	25			1	1	1		0	0	1	57
	0	1	1		0	1	0	26			1	1	1		0	1	0	58
	0	1	1		0	1	1	27			1	1	1		0	1	1	59
	0	1	1		1	0	0	28			1	1	1		1	0	0	60
	0	1	1		1	0	1	29			1	1	1		1	0	1	61
	0	1	1		1	1	0	30			1	1	1		1	1	0	62
	0	1	1		1	1	1	31			1	1	1		1	1	1	63

1. MSb = most significant bit

- *Classification* is the selection of traffic to be marked.
- *Marking*, according to RFC 2475, is the process of setting a Layer 3 DSCP value in a packet; in this publication, the definition of marking is extended to include setting Layer 2 CoS values.
- *Policing* is limiting bandwidth used by a flow of traffic. Policing can mark or drop traffic.

Basic QoS Model

Figure 44-2 illustrates a high-level flow of QoS function.

Figure 44-2 QoS Packet Processing

203973

The QoS model proceeds as follows:

-
- | | |
|---------------|--|
| Step 1 | The incoming packet is classified (based on different packet fields, receive port and/or VLAN) to belong to a traffic class. |
| Step 2 | Depending on the traffic class, the packet is rate-limited/policed and its priority is optionally <i>marked</i> (typically at the edge of the network) so that lower priority packets are dropped or marked with lower priority in the packet fields (DSCP and CoS). |
| Step 3 | After the packet has been marked, it is <i>looked up</i> for forwarding. This action obtains the transmit port and VLAN to transmit the packet. |
| Step 4 | The packet is classified in the output direction based on the transmit port and/or VLAN. The classification takes into account any marking of the packet by input QoS. |
| Step 5 | Depending on the output classification, the packet is policed, its priority is optionally <i>(re-)marked</i> , and the transmit queue for the packet is determined depending on the traffic class. |
| Step 6 | The transmit queue state is dynamically monitored via the AQM (Active Queue Management) algorithm and drop threshold configuration to determine whether the packet should be dropped or enqueued for transmission. |
| Step 7 | If eligible for transmission, the packet is enqueued to a transmit queue. The transmit queue is selected based on output QoS classification criteria. The selected queue provides the desired behavior in terms of latency and bandwidth. |
-

Classification

Classification is the process of distinguishing one kind of traffic from another by examining the fields in the packet. Classification is enabled when a QoS policy-map is attached to an interface.

You specify which fields in the frame or packet that you want to use to classify incoming traffic.

For non-IP traffic, you have the following classification options:

- CoS value in the VLAN tag of the incoming frame is used to classify the packet.
- If the frame does not contain a CoS value, the port's default CoS value ("0") is used for the classification.

Perform the classification based on a configured MAC ACL, which examines the fields in the Layer 2 header.

For IP traffic, you have the following classification options:

- IP DSCP or IP Precedence in the incoming packet is used for classification. DSCP values range from 0 to 63.
- Perform the classification based on a configured IP standard or extended ACL, which examines various fields in the IP header.

Classification Based on QoS ACLs

A packet can be classified for QoS using multiple match criteria, and the classification can specify whether the packet should match all of the specified match criteria or at least one of the match criteria. To define a QoS classifier, you can provide the match criteria using the *match* statements in a class map.

In the 'match' statements, you can specify the fields in the packet to match on, or you can use IP standard or IP extended ACLs or MAC ACLs. For more information, see the [“Classification Based on Class Maps and Policy Maps” section on page 44-7](#).

If the class map is configured to match all the match criteria, then a packet must satisfy all the match statements in the class map before the QoS action is taken. The QoS action for the packet is not taken if the packet does not match even one match criterion in the class map.

If the class map is configured to match at least one match criterion, then a packet must satisfy at least one of the match statements in the class map before the QoS action is taken. The QoS action for the packet is not taken if the packet does not match any match criteria in the class map.

**Note**

When you use the IP standard and IP extended ACLs, the permit and deny ACEs in the ACL have a slightly different meaning in the QoS context.

- If a packet encounters (and satisfies) an ACE with a “permit,” then the packet “matches” the match criterion in the QoS classification.
- If a packet encounters (and satisfies) an ACE with a “deny,” then the packet “does not match” the match criterion in the QoS classification.
- If no match with a permit action is encountered and all the ACEs have been examined, then the packet “does not match” the criterion in the QoS classification.

**Note**

When creating an access list, remember that, by default, the end of the access list contains an implicit deny statement for everything if it did not find a match before reaching the end.

After a traffic class has been defined with the class map, you can create a policy that defines the QoS actions for a traffic class. A policy might contain multiple classes with actions specified for each one of them. A policy might include commands to classify the class as a particular aggregate (for example, assign a DSCP) or rate limit the class. This policy is then attached to a particular port on which it becomes effective.

You implement IP ACLs to classify IP traffic by using the **access-list** global configuration command.

When a class-map is created with the **match-all** keyword, you cannot include both IP and MAC ACLs as match criteria.

Classification Based on Class Maps and Policy Maps

A class map is a mechanism that you use to isolate and name a specific traffic flow (or class) from all other traffic. The class map defines the criterion used to match against a specific traffic flow to further classify it; the criteria can include matching the access group defined by the ACL or matching a specific list of DSCP, IP precedence, or L2 CoS values. If you have more than one type of traffic that you want to classify, you can create another class map and use a different name. After a packet is matched against the class-map criteria, you can specify the QoS actions via a policy map.

A policy map specifies the QoS actions for the traffic classes. Actions can include setting a specific CoS, DSCP, or IP precedence value; policing the traffic to a specified rate; specifying the traffic bandwidth limitations; shaping the traffic to a specified rate. Before a policy map can be effective, you must attach it to an interface.

You create a class map by using the **class-map** global configuration command. When you enter the **class-map** command, the switch enters the class-map configuration mode. In this mode, you define the match criteria for the traffic by using the **match** class-map configuration command.

You create and name a policy map by using the **policy-map** global configuration command. When you enter this command, the switch enters the policy-map configuration mode. In this mode, you specify the actions to take on a specific traffic class by using the **set**, **police**, **bandwidth**, or **shape** policy-map configuration and policy-map class configuration commands. To make the policy map effective, you attach it to an interface by using the **service-policy** interface configuration command.

The policy map can also contain commands that define the policer, (the bandwidth limitations of the traffic) and the action to take if the limits are exceeded. For more information, see the [“Policing and Marking” section on page 44-8](#).

A policy map also has these characteristics:

- A policy map can contain up to 254 class statements.
- You can have different classes within a policy map.

Policing and Marking

Policing involves creating a policer that specifies the bandwidth limits for the traffic. Packets that exceed the limits are *out of profile* or *nonconforming*. Each policer specifies the action to take for packets that are in or out of profile. These actions, carried out by the marker, include passing through the packet without modification, dropping the packet, or marking down the packet with a new DSCP value that is obtained from the configurable policed-DSCP map. You can configure policer within a policy map with the **police** command in policy-map class configuration mode. For information on the policed-DSCP map, see the [“Queueing and Scheduling” section on page 44-8](#).

When configuring policing and policers, keep these items in mind:

- Policers account only for the Layer 2 header length when calculating policer rates. In contrast, shapers account for header length as well as IPG in rate calculations.
- Beginning with Cisco IOS Release 15.0(2)SG (IOS XE 3.2.0), Supervisor Engine 6-E, Supervisor Engine 6L-E, Supervisor Engine 7-E, and Supervisor Engine 7L-E support the **qos account layer-all encapsulation** command, which accounts for Layer 1 headers of 20 bytes (12 bytes preamble + 8 bytes IPG) and Layer 2 headers in policing features.
- Only the average rate and committed burst parameters are configurable.
- After you configure the policy map and policing actions, attach the policy to an ingress or egress interface by using the **service-policy** interface configuration command.
- For 2 rate 3 colors (2r3c) policers, if no explicit violation-action is specified, the exceed-action is used as the violate-action.

Queueing and Scheduling

The Catalyst 4500 series switch supports 8 transmit queues per port. Once the decision has been made to forward a packet out a port, the output QoS classification determines the transmit queue into which the packet must be enqueued.

Queues are assigned when an output policy attached to a port with one or more queuing related actions for one or more classes of traffic. Because there are only eight queues per port, there are at most eight traffic classes (including *class-default*, the reserved class) with queuing action(s). Classes of traffic that do not have any queuing action are referred to as non-queuing classes. Non-queuing class traffic use the queue corresponding to *class-default*.

Active Queue Management

Active queue management (AQM) is the pro-active approach of informing you about congestion before a buffer overflow occurs. AQM is done using Dynamic buffer limiting (DBL). DBL tracks the queue length for each traffic flow in the switch. When the queue length of a flow exceeds its limit, DBL drop packets.

Sharing Link Bandwidth Among Transmit Queues

The eight transmit queues for a transmit port share the available link bandwidth of that transmit port. You can set the link bandwidth to be shared differently among the transmit queues using the **bandwidth** command in the **policy-map class** configuration command in class mode.

With this command, you assign the minimum guaranteed bandwidth for each transmit queue.

By default, all queues are scheduled in a round robin manner.

Strict Priority / Low Latency Queueing

You can only configure one transmit queue on a port as strict priority (termed Low Latency Queue, or LLQ).

LLQ provides strict-priority queuing for a traffic class. It enables delay-sensitive data, such as voice, to be sent before packets in other queues. The priority queue is serviced first until it is empty or until it falls under its shape rate. Only one traffic stream can be destined for the priority queue per class-level policy. You enable the priority queue for a traffic class with the **priority policy-map class** configuration command in class mode.

Traffic Shaping

Traffic Shaping provides the ability to control the rate of outgoing traffic in order to make sure that the traffic conforms to the maximum rate of transmission contracted for it. Traffic that meets certain profile can be shaped to meet the downstream traffic rate requirements to handle any data rate mismatches.

Each transmit queue can be configured to transmit a maximum rate using the **shape** command in the **policy-map class** configuration command in class mode.

The configuration allows you to specify the maximum rate of traffic. Any traffic that exceeds the configured shape rate is queued and transmitted at the configured rate. If the burst of traffic exceeds the size of the queue, packets are dropped to maintain transmission at the configured shape rate.

Packet Modification

A packet is classified, policed, and queued to provide QoS. Packet modifications can occur during this process:

- For IP packets, classification involves assigning a DSCP to the packet. However, the packet is not modified at this stage; only an indication of the assigned DSCP is carried along. The reason for this is that QoS classification and ACL lookup occur in parallel, and it is possible that the ACL specifies that the packet should be denied and logged. In this situation, the packet is forwarded with its original DSCP to the CPU, where it is again processed through ACL software.

- For non-IP packets, classification involves assigning an internal DSCP to the packet, but because there is no DSCP in the non-IP packet, no overwrite occurs. Instead, the internal DSCP is used both for queueing and scheduling decisions and for writing the CoS priority value in the tag if the packet is being transmitted on either an ISL or 802.1Q trunk port.
- During policing, IP and non-IP packets can have another DSCP assigned to them (if they are out of profile and the policer specifies a markdown DSCP). Once again, the DSCP in the packet is not modified, but an indication of the marked-down value is carried along. For IP packets, the packet modification occurs at a later stage.

Per Port Per VLAN QoS

Per-port per-VLAN QoS (PVQoS) offers differentiated quality-of-services to individual VLANs on a trunk port. It enables service providers to rate limit individual VLAN-based services on each trunk port to a business or a residence. In an enterprise Voice-over-IP environment, it can be used to rate limit voice VLAN even if an attacker impersonates an IP phone. A per-port per-VLAN service policy can be separately applied to either ingress or egress traffic. For configuration details see [“Enabling Per-Port Per-VLAN QoS” section on page 44-71](#).

Flow-based QoS



Note

Before reading this section, you should be familiar with implementing Flexible NetFlow ([Chapter 76, “Configuring Flexible NetFlow”](#)) and QoS implementation in this chapter.

Flow based QoS enables microflow policing and marking capability to dynamically learn traffic flows. It also rate limits each unique flow to an individual rate. Flow based QoS is available on a Catalyst 4500 series switch with the built-in NetFlow hardware support. It can be applied to ingress traffic on both switched and routed interfaces with flow masks defined using Flexible NetFlow (FNF). It supports up to 100,000 individual flows in hardware and up to 512 unique policer configuration. Flow based QoS is typically used in environments where per-user, granular rate-limiting required. For example, per-flow outbound and inbound traffic rate might differ. Flow based QoS is also referred to as User Based Rate Limiting (UBRL).

A *flow* is defined as a stream of packets having the same properties as those defined by the key fields in the FNF flow record. A new flow is created when the value of data in packet’s key fields is unique with respect to the flow that already exist.

A flow based QoS policy possesses one or more classmaps matching on a FNF flow record. Such a classmap must be configured as **match-all** to match all the match criteria specified in the classmap. When a flow based QoS policy is attached to a QoS target, ingress traffic on the target is first classified based on the classification rules specified in the class-map. If the classifier has FNF flow record, the key fields specified in the FNF flow record are applied on the classified traffic to create flows provided the flow does not already exist. The corresponding policy actions (policing and marking) are then applied to these individual flows. Flow-based policers (termed *microflow policers*) rate limit each unique flow. Flows are dynamically created and inactive flows are periodically aged out.

Flow based QoS policy can be attached to QoS targets such as port (P), vlan (V), per-port-per-vlan (PV), and EtherChannel but only in the ingress direction.

For details on how to enable FNF, refer to the [“Applying Flow-based QoS Policy” section on page 44-76](#).

Using Metadata in QoS Policy

Beginning with Cisco IOS Release IOS XE 3.3.0SG and IOS 15.1(1)SG, you can configure class-map with metadata filters. A QoS policy that include such classes is termed a *metadata based QoS policy* or *parameterized QoS policy*. It allows you to classify flows based on intuitive and user friendly metadata attributes rather than individual flow 5-tuple and applicable QoS actions.

Software uses mechanisms like MSI and MSI-Proxy to do the following:

- Identify flows
- Glean metadata information from the traffic received at the network edge
- Generate and transport metadata information using RSVP messages hop-by-hop to every network element along the flow path using on-path RSVP signalling mechanism.

For configuration details on Cisco Medianet Metadata, refer to the following URLs:

<http://www.cisco.com/en/US/docs/ios-xml/ios/mdata/configuration/15-mt/metadata-framework.html>

For details on the metadata commands, refer to the following URL:

<http://www.cisco.com/en/US/docs/ios-xml/ios/qos/command/qos-cr-book.html>

For configuration details on Cisco Media Services Proxy, refer to the following URL:

<http://www.cisco.com/en/US/docs/ios-xml/ios/msp/configuration/15-mt/media-ser-prxy.html>

For command details on Cisco Media Services Proxy, refer to the following URL:

<http://www.cisco.com/en/US/docs/ios-xml/ios/msp/command/reference/guide/media-ser-prxy.html>

Restrictions

The following restrictions apply to using a metadata-based QoS policy on a Catalyst 4500 series switch:

- They can only be attached to target in input direction.
- They can only be attached to physical ports and EtherChannel. They cannot be attached to VLANs, port VLANs, and SVI interfaces.
- A policy can have multiple metadata-based classifiers.
- A class-map can have one or more metadata filters with match-any or match-all semantics.
- Policy actions corresponding to metadata class are applied on aggregate traffic. However, if the metadata filter is configured along with Flexible NetFlow record filter, the policy action (like policer) applies on individual flows.
- If there are no flows associated with metadata filter, the software configures an implicit ACL with a deny ACE.
- If the same metadata QoS policy is applied on multiple interfaces, the policy is installed in hardware in separate TCAM entries for each interface; the TCAM entries are not shared by the interfaces.
- When a new flow is associated with a metadata filter, the software installs a new set of TCAM entries that includes the new flow along with other existing previously-discovered flows.

Observing Metadata Filter Statistics

- Although interfaces with the same metadata policy do not share TCAM resources in hardware, the metadata filter statistics observed with the show policy-map interface *ifname* command are reported as though it were shared.
- Only metadata filter statistics are available. The individual flow statistics are not available.

Example

The following example illustrates a metadata-based QoS policy with two classes using metadata filters:

```
class-map c1
  match application telepresence-media

class-map c2
  match access-group name mysubnet

class-map match-any c3
  match application webex-video
  match application webex-audio

policy-map p1
  class c1
    police cir 10m
  class c2
    set dscp cs1
    police cir 2m
  class c3
    police cir 5m
```

Configuring System Queue Limit



Note

This feature is available only from Cisco IOS Release 15.0(2)SG1 and later and Cisco IOS Release XE 3.2.1SG.

With the **hw-module system max-queue-limit** command, the Catalyst 4500 series switch allows you to change the queue limit for all interfaces globally, instead of applying a policy with queue limit to all the interfaces.

To set the queue limit globally, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# hw-module system max-queue-limit max-queue-limit	Sets the queue limit for all interfaces globally. Valid values are from 1024 to 8184. The value must be a multiple of 8.
Step 3	Switch(config)# exit	Returns to privileged EXEC mode.
Step 4	Switch# reload or Switch# redundancy reload shelf Switch# redundancy force-switchover	Reloads standalone supervisor engine. Reloads redundancy supervisor engine in SSO mode. Reloads redundancy supervisor engine in RPR mode. This command must be followed by another redundancy force-switchover.

This is a global configuration command. You can override it with the per port, per class, **queue-limit** command.

For a standalone supervisor engine, you must reboot the engine after applying this command.

For redundant supervisors in SSO mode, you must enter the **redundancy reload shelf** command enforce reboot to both the supervisors. For redundancy supervisors in RPR mode, you must execute two consecutive switchovers to enforce the system queue limit on both the supervisors.

This example shows how to set the queue limit globally to 1024 on a standalone supervisor engine:

```
Switch> enable
Switch# configure terminal
Switch(config)# hw-module system max-queue-limit 1024
Switch(config)# exit
Switch# reload (for standalone supervisors)
Switch# redundancy reload shelf (for redundancy supervisors in SSO mode)
or
Switch# redundancy force-switchover (followed by another redundancy force-switchover, for
redundancy supervisors in RPR mode)
```

Configuring QoS with a 40-Gigabit Ethernet Interface

Starting with Cisco IOS XE 3.10.0E, the 40-GE interface is available with Supervisor Engine 9-E.

After you configure the policy map and policy map actions, you attach the policy to an ingress or egress interface by using the **service-policy** interface configuration command. If the interface you are attaching it to is a 40-Gigabit Ethernet interface, note the following:

- The **percent** keyword is not supported—policing actions such as shaping and bandwidth must be configured only with absolute values and not percent values.
- The reserved class (**class-default**) is required—This class must be present in the policy map, and must be configured with an absolute value.



Note

This 40-GE interface limitation applies to QoS configurations in VSS environments, non-VSS environments, and to auto-QoS configurations.

The following example shows how to configure a class map with match criteria, the class-default configured with a value, and how to attach the corresponding policy map, called “queuing” to a 40-Gigabit Ethernet interface:

```
Switch# configure terminal
Switch(config)# class-map dscp-10
Switch(config-cmap)# match dscp 10
Switch(config-cmap)# exit

Switch(config)# policy-map queuing
Switch(config-pmap)# class dscp-10
Switch(config-pmap-c)# shape average 10m
Switch(config-pmap-c)# exit
Switch(config-pmap)# class class-default
Switch(config-pmap-c)# bandwidth 10000
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit

Switch(config)# interface fortygigabitethernet1/1
Switch(config-if)# service-policy output queuing
Switch(config-if)# end
Switch#
```

Configuring VSS QoS

**Note**

HQoS is not supported on the Catalyst 4500 series switch.

Topics include:

- [MQC-based QoS Configuration, page 44-14](#)
- [Platform-supported Classification Criteria and QoS Features, page 44-15](#)
- [Platform Hardware Capabilities, page 44-50](#)
- [Prerequisites for Applying a QoS Service Policy, page 44-50](#)
- [Restrictions for Applying a QoS Service Policy, page 44-51](#)
- [Classification, page 44-51](#)
- [Policing, page 44-52](#)
- [Marking Network Traffic, page 44-53](#)
- [Shaping, Sharing \(Bandwidth\), Priority Queuing, Queue-limiting and DBL, page 44-60](#)
- [Enabling Per-Port Per-VLAN QoS, page 44-71](#)
- [Applying Flow-based QoS Policy, page 44-76](#)
- [Configuring CoS Mutation, page 44-80](#)
- [Configuring System Queue Limit, page 44-81](#)

MQC-based QoS Configuration

**Note**

Starting with Cisco IOS Release 12.2(40)SG Catalyst 4500 series switch with Supervisor Engine 6-E or Supervisor Engine 6L-E use the MQC model of QoS.

To apply QoS, you use the Modular QoS Command-Line Interface (MQC), which is a CLI structure that allows you to complete the following tasks:

- Specify the matching criteria used to define a traffic class.
- Create a traffic policy (policy map). The traffic policy defines the QoS policy actions to be taken for each traffic class.
- Apply the policy actions specified in the policy map to an interface, VLAN, or port and VLAN.

For more information about the MQC, see the “Modular Quality of Service Command-Line Interface” section of the *Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.3*.

**Note**

The incoming traffic is considered trusted by default. Only when the *trusted boundary* feature is enabled on an interface can the port enter untrusted mode. In this mode, the switch marks the DSCP value of an IP packet and the CoS value of the VLAN tag on the Ethernet frame as “0”.

Platform-supported Classification Criteria and QoS Features

The following table provides a summary of various classification criteria and actions supported on the Catalyst 4500 series switch. For details, refer to the *Catalyst 4500 Series Switch Command Reference*.

Supported classification actions	Descriptions
match access-group	Configures the match criteria for a class map on the basis of the specified ACL.
match any	Configures the match criteria for a class map to be successful match criteria for all packets.
match cos	Matches a packet based on a Layer 2 class of service (CoS) marking.
match [ip] dscp	Identifies a specific IP differentiated service code point (DSCP) value as a match criterion. Up to eight DSCP values can be included in one match statement.
match [ip] precedence	Identifies IP precedence values as match criteria.
match protocol	Configures the match criteria for a class map on the basis of the specified protocol.
Supported Qos Features	Descriptions
police	Configures traffic policing.
police (percent)	Configures traffic policing on the basis of a percentage of bandwidth available on an interface.
police (two rates)	Configures traffic policing using two rates, the committed information rate (CIR) and the peak information rate (PIR).
set cos	Sets the Layer 2 class of service (CoS) value of an outgoing packet.
set dscp	Marks a packet by setting the differentiated services code point (DSCP) value in the type of service (ToS) byte of IPv4 or traffic class byte of IPv6 packet.
set precedence	Sets the precedence value in the packet header.
table map support	Unconditional marking of one packet field based on another packet field.
priority	Gives priority to a class of traffic belonging to a policy map.
shape	Shapes traffic to the indicated bit rate according to the algorithm specified.
bandwidth	Provides a guaranteed minimum bandwidth to each of the eight queues.
dbl	Dynamic buffer limit.
queue-limit	Specifies the maximum number of packets a transmit queue can hold.



Note

The classification action “match qos-group” is not supported in Cisco Release IOS XE 3.4.0SG and IOS 15.1(2)SG.



Note

The QoS feature “set qos-group” is not supported in Cisco Release IOS XE 3.4.0SG and IOS 15.1(2)SG.

Platform Hardware Capabilities

Qos Actions	Numbers of entries supported
Classification	64k input and 64k output classification entries are supported. A given policy can use at most 24k ACLs
Policing	16K policers are supported. Policers are allocated to given direction in blocks of 2k. For example, 2k policers can be used in for input and 14k policers can be used for output. Single rate policers uses one policer entry. Single Rate Three Color Marker (srTCM) (RFC 2697) and Two Rate Three Color Marker (trTCM) (RFC 2698) uses two policer entries
Marking	Marking of Cos and DSCP/Precedence is supported through two marking tables, each capable of supporting 512 entries. There are separate tables for each direction.
Queuing	The queue size is Configurable with the maximum number of entries configurable per port depending on the chassis and line card type.
DBL	You can enable DBL action on all configured class-maps.

Prerequisites for Applying a QoS Service Policy

Unlike the Switch QoS model, there is no prerequisite for enabling QoS on various targets. Just the attachment of a service policy enables QoS and detachment of that policy disables QoS on that target.

Restrictions for Applying a QoS Service Policy

Traffic marking can be configured on an interface, a VLAN, or a port and VLAN. An interface can be a Layer 2 access port, a Layer 2 switch trunk, a Layer 3 routed port, or an EtherChannel. A policy is attached to a VLAN using *vlan configuration* mode.

Attaching QoS service policy to VLANs and EtherChannel is described in the [“Policy Associations” section on page 44-74](#).

Classification

The supervisor engine supports classification of Layer 2, IP, IPv6 packets, and ARP packets marking performed on input can be matched in the output direction. The previous table lists the full set of capabilities. By default, the switch also supports classification resources sharing. Similarly, when the same policy is attached to a port or a VLAN or on per-port per-vlan targets, ACL entries are shared though QoS actions are unique on each target.

For example:

```

class-map c1
  match ip dscp 50
Policy Map p1
  class c1
    police rate 1 m burst 200000

```

If policy-map p1 is applied to interfaces Gig 1/1 and Gig 1/2, 1 CAM entry is used (one ACE that matches IP packets), but 2 policers are allocated (one per target). So, all IP packets with dscp 50 are policed to 1 mbps on interface Gig 1/1 and packets on interface Gig 1/2 are policed to 1 mbps.

**Note**

With Cisco IOS Release 12.2(46)SG, you can issue the **match protocol arp** command. For details, see the *Catalyst 4500 Series Switch Cisco IOS Command Reference*.

Classification Statistics

The supervisor engine supports only packet based classification statistics and TCAM resource sharing. When a policy-map is applied on multiple targets, the command **show policy-map interface** displays the aggregate classification statistics, not those specific to an interface.

**Note**

To obtain per interface policy-map stats, you should configure a unique policy-map name on each interface.

When a policy-map is attached to a port-channel member ports, classification statistics are not displayed.

Configuring a Policy Map

You can attach only one policy map to an interface. Policy maps can contain one or more policy-map classes, each with different match criteria and actions.

Configure a separate policy-map class in the policy map for each type of traffic that an interface receives. Put all commands for each type of traffic in the same policy-map class. QoS does not attempt to apply commands from more than one policy-map class to matched traffic.

Creating a Policy Map

To create a policy map, enter this command:

Command	Purpose
Switch(config)# [no] policy-map <i>policy_name</i>	Creates a policy map with a user-specified name. Use the no keyword to delete the policy map.

Attaching a Policy Map to an Interface

To create a policy map, enter this command:

Command	Purpose
Switch(config)# interface {vlan <i>vlan_ID</i> { fastethernet gigabitethernet } <i>slot/interface</i> Port-channel <i>number</i> }	Selects the interface to configure.
Switch(config-if)# [no] service-policy input <i>policy_map_name</i>	Attaches a policy map to the input direction of the interface. Use the no keyword to detach a policy map from an interface.

Command	Purpose
Switch(config-if)# end	Exits configuration mode.
Switch# show policy-map interface {vlan vlan_ID { fastethernet gigabitethernet } slot/interface}	Verifies the configuration.

Policing

The supervisor engine supports policers in the following operation modes:

- Single Rate Policer Two Color Marker

This kind of policer is configured with just the committed rate (CIR) and normal burst and it has only conform and exceed actions.
- Single Rate Three Color Marker (srTCM) (RFC 2697)
- Two Rate Three Color Marker (trTCM) (RFC 2698)
- Color Blind Mode

Policing accuracy of 0.75% of configured policer rate.

The engine supports 16384 (16 x 1024, 16K) single rate, single burst policers. 16K policers are organized as 8 banks of 2K policers. The policer banks are dynamically assigned (input or output policer bank) by the software depending on the QoS configuration. So, the 16K policers are dynamically partitioned by software as follows:

- 0 Input Policers and 16K Output Policers
- 2K Input Policers and 14K Output Policers
- 4K Input Policers and 12K Output Policers
- 6K Input Policers and 10K Output Policers
- 8K Input Policers and 8K Output Policers
- 10K Input Policers and 6K Output Policers
- 12K Input Policers and 4K Output Policers
- 14K Input Policers and 2K Output Policers
- 16K Input Policers and 0 Output Policers

These numbers represent individual policer entries in the hardware that support a single rate and burst parameter. Based on this, a switch supports the following number of policers:

- 16K Single Rate Policer with Single Burst (Two Color Marker)
- 8K Single Rate Three Color Marker (srTCM)
- 8K Two Rate Three Color Marker (trTCM)

These policers are partitioned between Input and Output in chunks of 2K policer banks. The different types of policers can all co-exist in the system. However, a given type of policer (srTCM, trTCM etc.) is configurable as a block of 128 policers.



Note

Two policers are reserved for internal use.

How to Implement Policing

For details on how to implement the policing features on a Catalyst 4500 series switch, refer to the Cisco IOS documentation at the following link:

http://www.cisco.com/en/US/docs/ios/12_2/qos/configuration/guide/qcfdpolsh.html

Platform Restrictions

Platform restrictions include the following:

- Multi-policer actions can be specified (setting CoS and IP DSCP is supported).
- When unconditional marking and policer based marking exists on the same field(cos or dscp or precedence), policer-based marking is preferred.
- If policer based service-policy is attached to both a port and a VLAN, port-based policed is preferred by default. To over-ride a specific VLAN policy on a given port, then you must configure a per-port per-vlan policy.
- You should not delete a port-channel with a per-port, per-VLAN QoS policy.

Workaround: Before deleting the port-channel, do the following:

1. Remove any per-port per-VLAN QoS policies, if any.
2. Remove the VLAN configuration on the port-channel with the **no vlan-range** command.

Marking Network Traffic

Marking network traffic allows you to set or modify the attributes of traffic (that is, packets) belonging to a specific class or category. When used in conjunction with network traffic classification, marking network traffic is the foundation for enabling many quality of service (QoS) features on your network. This module contains conceptual information and the configuration tasks for marking network traffic.

Contents

- “Information About Marking Network Traffic” section on page 44-54
- “Marking Action Drivers” section on page 44-56
- “Traffic Marking Procedure Flowchart” section on page 44-56
- “Restrictions for Marking Network Traffic” section on page 44-57
- “Multi-attribute Marking Support” section on page 44-57
- “Hardware Capabilities for Marking” section on page 44-58
- “Configuring the Policy Map Marking Action” section on page 44-58
- “Marking Statistics” section on page 44-60

Information About Marking Network Traffic

To mark network traffic, you should understand the following concepts:

- [“Purpose of Marking Network Traffic” section on page 44-54](#)
- [“Benefits of Marking Network Traffic” section on page 44-54](#)
- [“Two Methods for Marking Traffic Attributes” section on page 44-55](#)

Purpose of Marking Network Traffic

Traffic marking is used to identify certain traffic types for unique handling, effectively partitioning network traffic into different categories.

After the network traffic is organized into classes by traffic classification, traffic marking allows you to mark (that is, set or change) a value (attribute) for the traffic belonging to a specific class. For instance, you may want to change the class of service (CoS) value from 2 to 1 in one class, or you may want to change the differentiated services code point (DSCP) value from 3 to 2 in another class. In this module, these values are referred to as attributes or marking fields.

Attributes that can be set and modified include the following:

- CoS value of a tagged Ethernet frame
- DSCP/Precedence value in the Type of Service (ToS) byte of IPv4.
- DSCP /Precedence value in the traffic class byte of IPv6

Benefits of Marking Network Traffic

Traffic marking allows you to fine-tune the attributes for traffic on your network. This increased granularity helps isolate traffic that requires special handling, and thus, helps to achieve optimal application performance.

Traffic marking allows you to determine how traffic will be treated, based on how the attributes for the network traffic are set. It allows you to segment network traffic into multiple priority levels or classes of service based on those attributes, as follows:

- Traffic marking is often used to set the IP precedence or IP DSCP values for traffic entering a network. Networking devices within your network can then use the newly marked IP precedence values to determine how traffic should be treated. For example, voice traffic can be marked with a particular IP precedence or DSCP and strict priority can then be configured to put all packets of that marking into that queue. In this case, the marking was used to identify traffic for strict priority queue.
- Traffic marking can be used to identify traffic for any class-based QoS feature (any feature available in policy map class configuration mode, although some restrictions exist).
- Traffic marking can be used to assign traffic to a QoS group within a switch. The switch can use the QoS groups to determine how to prioritize traffic for transmission. The QoS group value is usually used for one of the two following reasons:
 - To leverage a large range of traffic classes. The QoS group value has 64 different individual markings, similar to DSCP.
 - If changing the Precedence or DSCP value is undesirable.

Two Methods for Marking Traffic Attributes



Note

This section describes *Unconditional* marking, which differs from *Policer-based* marking. Unconditional marking is based solely on classification.

Method One: Unconditional Explicit Marking (using the set command)

You specify the traffic attribute you want to change with a set command configured in a policy map. The following table lists the available set commands and the corresponding attribute. For details on the set command, refer to the *Catalyst 4500 Series Switch Command Reference*.

Table 44-2 **set Commands and Applicable Packet Types**

set Commands	Traffic Attribute	Packet Type
set cos	Layer 2 CoS value of the outgoing traffic	Ethernet IPv4, IPv6
set dscp	DSCP value in the ToS byte	IPv4, IPv6
set precedence	precedence value in the packet header	IPv4, IPv6

If you are using individual **set** commands, those set commands are specified in a policy map. The following is a sample of a policy map configured with one of the set commands listed in [Table 44-4](#).

In this sample configuration, the **set cos** command has been configured in the policy map (policy1) to mark the CoS attribute:

```
enable
configure terminal
policy map p1
  class class1
    set cos 3
end
```

For information on configuring a policy map, see the “[Creating a Policy Map](#)” section on page 44-52.

The final task is to attach the policy map to the interface. For information on attaching the policy map to the interface, see the “[Attaching a Policy Map to an Interface](#)” section on page 44-52.

Method Two: Unconditional Tablemap-based Marking

You can create a table map that can be used to mark traffic attributes. A table map is a kind of two-way conversion chart that lists and maps one traffic attribute to another. A table map supports a many-to-one type of conversion and mapping scheme. The table map establishes a to-from relationship for the traffic attributes and defines the change to be made to the attribute. That is, an attribute is set to one value that is taken from another value. The values are based on the specific attribute being changed. For instance, the Precedence attribute can be a number from 0 to 7, while the DSCP attribute can be a number from 0 to 63.

The following is a sample table map configuration:

```
table-map table-map1
map from 0 to 1
map from 2 to 3
exit
```

The following table lists the traffic attributes for which a to-from relationship can be established using the table map.

Table 44-3 Traffic Attributes for Which a To-From Relationship Can Be Established

The "To" Attribute	The "From" Attribute
Precedence	CoS, DSCP, Precedence
DSCP	COS, DSCP, Precedence
CoS	DSCP, CoS, Precedence

The following is an example of a policy map (policy2) configured to use the table map (table-map1) created earlier:

```
Policy map policy
  class class-default
    set cos dscp table table-map
exit
```

In this example, a mapping relationship was created between the CoS attribute and the DSCP attribute as defined in the table map.

For information on configuring a policy map to use a table map, [“Configuring a Policy Map” section on page 44-51](#).

The final task is to attach the policy map to the interface. For information on attaching the policy map to the interface, see the [“Attaching a Policy Map to an Interface” section on page 44-52](#).

Marking Action Drivers

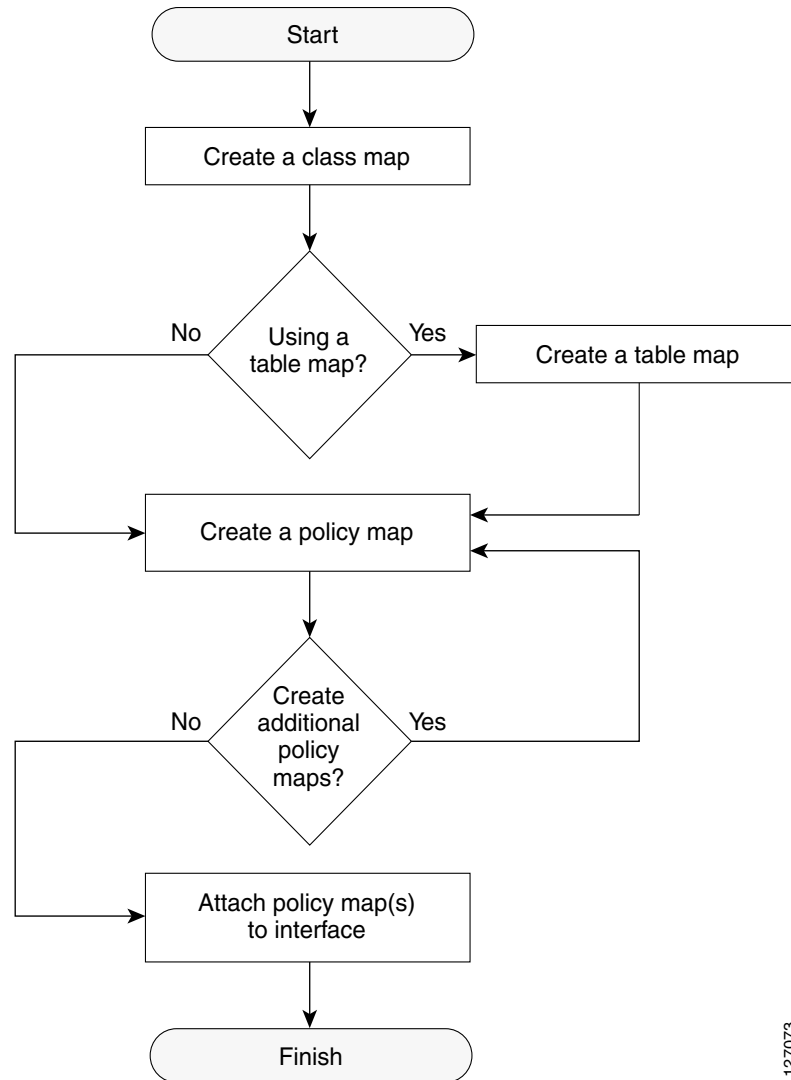
A marking action can be triggered based on one of the two QoS processing steps.

Classification based: In this case, all the traffic matching a class is marked using either explicit or tablemap based method. This method is referred to as *unconditional* marking.

Policer result-based: In this case, a class of traffic is marked differently based on the policer result (conform/exceed/violate) applicable to that packet. This method is referred to as *conditional* marking.

Traffic Marking Procedure Flowchart

[Figure 44-5](#) illustrates the order of the procedures for configuring traffic marking.

Figure 44-3 Traffic marking Procedure Flowchart

127073

Restrictions for Marking Network Traffic

The following restrictions apply to the following packet marking action:

- Only explicit marking is supported for policer-based marking.

Multi-attribute Marking Support

The supervisor engine can mark more than one QoS attribute of a packet matching a class of traffic. For example, DSCP, and CoS can all be set together, using either explicit or tablemap-based marking.



Note

When using unconditional explicit marking of multiple fields or policer-based multi-field, multi-region (conform/exceed/violate) marking the number of table maps that can be setup in TOS or COS marking tables will be less than the maximum supported.

Hardware Capabilities for Marking

Supervisor Engine 6-E, and Supervisor Engine 6L-E provide a 128 entry marking action (Supervisor Engines 9-E, 8L-E, 8-E, 7-LE, and 7-E provide a 256 entry marking action) where each entry specifies the type of marking actions on CoS and DSCP/Precedence fields as well as policer action to transmit/markdown/drop a packet.

One such table is supported for each direction, input and output. This table is used for both unconditional marking as well as policer-based marking. It can be used to support 256 unique marking actions or 64 unique policer-based actions or a combinations of the two.

For each of the marking fields (COS and DSCP), the supervisor engine provides 512 entry marking tables for each direction. These are similar to mapping tables available on supervisor engines that support the switch QoS model. However, these provide an ability to have multiple unique mapping tables that are setup by the user.

For example, the TOS marking table provides marking of DSCP/Precedence fields and can be used as one of the following:

- 64 (32) different tablemaps with each one mapping 8 CoS (16 CoS and CFi) values to DSCP in input (output) direction
- a combination of above two types of tablemaps

Similar mappings are available on the 512 entry COS marking table.

Configuring the Policy Map Marking Action

This section describes how to establish unconditional marking action for network traffic.

As a prerequisites, create a class map (*ipp5*) and a policy map. (Refer to the “[Configuring a Policy Map](#)” section on page 44-51).



Note

The marking action command options have been extended (refer to [Table 44-4 on page 44-55](#) and [Table 44-5 on page 44-56](#)).

Configuring Tablemap-based Unconditional Marking

To configure table-map based unconditional marking, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# table-map name	Configures a tablemap.
Step 3	Switch(config-tablemap)# map from from_value to to_value	Creates a map from a <i>from_value</i> to a <i>to_value</i>
Step 4	Switch(config-tablemap)# exit	Exits table-map configuration mode.
Step 5	Switch(config)# policy-map name	Enters policy-map configuration mode.
Step 6	Switch(config-p)# class name	Selects the class for QoS actions.
Step 7	Switch(config-p-c)# set cos dscp prec cos dscp prec [table name]	Selects the marking action based on an implicit or explicit table-map.
Step 8	Switch(config-p-c)# end	Exits configuration mode.

	Command	Purpose
Step 9	Switch# show policy-map name	Verifies the configuration of the policy-map.
Step 10	Switch# show table-map name	Verifies the configuration of the table-map.

The following example shows how to enable marking action using table-map.

```
Switch(config)# table-map dscp2Cos
Switch(config-tablemap)# map from 8 to 1
Switch(config-tablemap)# exit
Switch(config)# policy-map p1
Switch(config-pmap)# class ipp5
Switch(config-pmap-c)# set cos dscp table dscp2Cos
Switch(config-pmap-c)# end
Switch# show policy-map p1

Policy Map p1
  Class ipp5
    set cos dscp table dscp2Qos

Switch# show table-map dscp2Cos

Table Map dscp2Cos
  from 8 to 1
  default copy
```

Configuring Policer Result-based Conditional Marking

To configure policer result-based conditional marking, setup a single rate or dual rate policer. Refer to the [“How to Implement Policing”](#) section on page 44-53.

This example shows how to configure a two rate three-color policer with explicit actions for each policer region:

```
Switch# configure terminal
Switch(config-pmap-c)# policer cir percent 20 pir percent 30
Switch(config-pmap-c-policer)# conform-action set-cos-transmit 3 set-dscp-transmit 10
Switch(config-pmap-c-policer)# exceed-action set-cos-transmit 4 set-dscp-transmit 20
Switch(config-pmap-c-policer)# violate action drop
Switch# show policy-map p1

Policy Map police
  Class ipp5
    police cir percent 20 pir percent 30
      conform-action set-cos-transmit 3
      conform-action set-dscp-transmit af11
      exceed-action set-cos-transmit 4
      exceed-action set-dscp-transmit af22
      violate-action drop
```

Marking Statistics

The marking statistics indicate the number of packets that are *marked*.

For unconditional marking, the *classification entry* points to an entry in the marking action table that in turn indicates the fields in the packet that are marked. Therefore, the classification statistics by itself indicates the unconditional marking statistics.

For a conditional marking using policer, provided the policer is a packet rate policer, you cannot determine the number packets marked because the policer only provides byte statistics for different policing results.

Shaping, Sharing (Bandwidth), Priority Queuing, Queue-limiting and DBL

Catalyst 4500 series switches support the Classification-based (class-based) mode for transmit queue selection. In this mode, the transmit queue selection is based on the Output QoS classification lookup.



Note

Only output (egress) queuing is supported.

The supervisor engine supports 8 transmit queues per port. Once the forwarding decision has been made to forward a packet out a port, the output QoS classification determines the transmit queue into which the packet needs to be enqueued.

By default, without any service policies associated with a port, there are two queues (a control packet queue and a default queue) with no guarantee as to the bandwidth or kind of prioritization. The only exception is that system generated control packets are enqueued into control packet queue so that control traffic receives some minimum link bandwidth.

Queues are assigned when an output policy attached to a port with one or more queuing related actions for one or more classes of traffic. Because there are only eight queues per port, there can be at most eight classes of traffic (including the reserved class, class-default) with queuing action(s). Classes of traffic that do not have any queuing action are referred to as *non-queuing* classes. Non-queuing class traffic ends up using the queue corresponding to class class-default.

When a queuing policy (a policy with queuing action) is attached, the control packet queue is deleted and the control packets are enqueued into respective queue per their classification. An egress QoS class must be configured to match IP Precedence 6 and 7 traffic, and a bandwidth guarantee must be configured.

Dynamic resizing of queues (queue limit class-map action) is supported through the use of the **queue-limit** command. Based on the chassis and line card type, all eight queues on a port are configured with equal queue size.

Shaping

Shaping enables you to delay out-of-profile packets in queues so that they conform to a specified profile. Shaping is distinct from policing. Policing drops packets that exceed a configured threshold, whereas shaping *buffers* packets so that traffic remains within a given threshold. Shaping offers greater *smoothness* in handling traffic than policing. You enable average-rate traffic shaping on a traffic class with the **policy-map** class configuration command.

The supervisor engine supports a range of 32kbps to 10 gbps for shaping, with a precision of approximately +/- 0.75 per cent.

When a queuing class is configured without any explicit shape configuration, the queue shape is set to the link rate.

To configure class-level shaping in a service policy, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# policy-map <i>policy-map-name</i>	Creates a policy map by entering the policy-map name, and enter policy-map configuration mode. By default, no policy maps are defined.
Step 3	Switch(config-pmap)# class <i>class-name</i>	Specifies the name of the class whose traffic policy you want to create or change, and enter policy-map class configuration mode. By default, no traffic classes are defined.
Step 4	Switch(config-pmap-class)# shape average { <i>cir-bps</i> [<i>optional_postfix</i>] <i>percent percent</i> }	Enables average-rate traffic shaping. You can specify the shaping rate in absolute value or as a percentage: <ul style="list-style-type: none"> For <i>cir-bps</i> [<i>optional_postfix</i>], specify the shaping rate in bps. Range is 32000 to 10000000000 bps. Supply an optional postfix (K, M, G). For <i>percent</i>, specify the percentage of link rate to shape the class of traffic. The range is 1 to 100. By default, average-rate traffic shaping is disabled.
Step 5	Switch(config-pmap-class)# exit	Returns to policy-map configuration mode.
Step 6	Switch(config-pmap)# exit	Returns to global configuration mode.
Step 7	Switch(config)# interface <i>interface-id</i>	Specifies a physical port and enter interface configuration mode.
Step 8	Switch(config-interface)# service-policy output <i>policy-map-name</i>	Specifies the policy-map name, and apply it a physical interface.
Step 9	Switch(config-interface)# end	Returns to privileged EXEC mode.
Step 10	Switch# show policy-map [<i>policy-map-name</i> [<i>class class-map-name</i>]] or Switch# show policy-map interface <i>interface-id</i>	Verifies your entries.
Step 11	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To delete an existing policy map, use the **no policy-map** *policy-map-name* global configuration command. To delete an existing class, use the **no class** *class-name* policy-map configuration command. To disable the average-rate traffic shaping, use the **no shape average** policy-map class configuration command.

This example shows how to configure class-level, average-rate shaping. It limits traffic class class1 to a data transmission rate of 256 kbps:

```
Switch# configure terminal
Switch(config)# policy-map policy1
Switch(config-pmap)# class class1
Switch(config-pmap-c)# shape average 256000
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1
```

```
Switch(config-if)# service-policy output policy1
Switch(config-if)# end
Switch#
```

```
Switch# show policy-map policy1
  Policy Map policy1
    Class class1
      shape average 256000
```

This example shows how to configure class-level, average shape percentage to 32% of link bandwidth for queuing-class traffic:

```
Switch# configure terminal
Switch(config)# policy-map queuing-policy
Switch(config-pmap)# class queuing-class
Switch(config-pmap-c)# shape average percent 32
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1
Switch(config-if)# service-policy output queuing-policy1
Switch(config-if)# end
Switch #
```

```
Switch# show policy-map queuing-policy
  Policy Map queuing-policy
    Class queuing-class
      Average Rate Traffic Shaping
        cir 32%
```

Sharing (bandwidth)

The bandwidth assigned to a class of traffic is the minimum bandwidth that is guaranteed to the class during congestion. Transmit Queue Sharing is the process by which output link bandwidth is shared among multiple queues of a given port.

The supervisor engine supports a range of 32 kbps to 10 gbps for sharing, with a precision of approximately +/- 0.75 per cent. The sum of configured bandwidth across all queuing classes should not exceed the link bandwidth.

To configure class-level bandwidth action in a service policy, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# policy-map <i>policy-map-name</i>	Creates a policy map by entering the policy-map name, and enter policy-map configuration mode. By default, no policy maps are defined.
Step 3	Switch(config-pmap)# class <i>class-name</i>	Specifies the name of the class whose traffic policy you want to create or change, and enter policy-map class configuration mode. By default, no traffic classes are defined.

	Command	Purpose
Step 4	Switch(config-pmap-class)# bandwidth { <i>bandwidth-kbps</i> percent <i>percent</i> }	Specifies the minimum bandwidth provided to a class belonging to the policy map when there is traffic congestion in the switch. If the switch is not congested, the class receives more bandwidth than you specify with the bandwidth command. By default, no bandwidth is specified. You can specify the bandwidth in kbps or as a percentage: <ul style="list-style-type: none"> For <i>bandwidth-kbps</i>, specify the bandwidth amount in kbps assigned to the class. The range is 32 to 10000000. For <i>percent</i>, specify the percentage of available bandwidth assigned to the class. The range is 1 to 100. Specify all the class bandwidths in either kbps or in percentages, but not a mix of both.
Step 5	Switch(config-pmap-class)# exit	Returns to policy-map configuration mode.
Step 6	Switch(config-pmap)# exit	Returns to global configuration mode.
Step 7	Switch(config)# interface <i>interface-id</i>	Specifies a physical port and enter interface configuration mode.
Step 8	Switch(config-interface)# service-policy output <i>policy-map-name</i>	Specifies the policy-map name, and apply it a physical interface.
Step 9	Switch(config-interface)# end	Returns to privileged EXEC mode.
Step 10	Switch# show policy-map [<i>policy-map-name</i> [class <i>class-map-name</i>]] or Switch# show policy-map interface <i>interface-id</i>	Verifies your entries.
Step 11	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To delete an existing policy map, use the **no policy-map** *policy-map-name* global configuration command. To delete an existing class, use the **no class** *class-name* policy-map configuration command. To return to the default bandwidth, use the **no bandwidth** policy-map class configuration command.

This example shows how to create a class-level policy map called policy11 for three classes called prec1, prec2, and prec3. In the policy for these classes, 30 percent of the available bandwidth is assigned to the queue for the first class, 20 percent is assigned to the queue for the second class, and 10 percent is assigned to the queue for the third class.

```
Switch # configure terminal
Switch(config)# policy-map policy11
Switch(config-pmap)# class prec1
Switch(config-pmap-c)# bandwidth percent 30
Switch(config-pmap-c)# exit
Switch(config-pmap)# class prec2
Switch(config-pmap-c)# bandwidth percent 20
Switch(config-pmap-c)# exit
Switch(config-pmap)# class prec3
Switch(config-pmap-c)# bandwidth percent 10
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1
Switch(config-if)# service-policy output policy11
```

```

Switch(config-if)# end
Switch #

Switch# show policy-map policy11
Policy Map policy11
  Class prec1
    bandwidth percent 30
  Class prec2
    bandwidth percent 20
  Class prec3
    bandwidth percent 10

```

This example shows how to create a class-level policy map called policy11 for three classes called prec1, prec2, and prec3. In the policy for these classes, 300 mbps of the available bandwidth is assigned to the queue for the first class, 200 mbps is assigned to the queue for the second class, and 100 mbps is assigned to the queue for the third class.

```

Switch # configure terminal
Switch(config)# policy-map policy11
Switch(config-pmap)# class prec1
Switch(config-pmap-c)# bandwidth 300000
Switch(config-pmap-c)# exit
Switch(config-pmap)# class prec2
Switch(config-pmap-c)# bandwidth 200000
Switch(config-pmap-c)# exit
Switch(config-pmap)# class prec3
Switch(config-pmap-c)# bandwidth 100000
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1
Switch(config-if)# service-policy output policy11
Switch(config-if)# end
Switch #

Switch# show policy-map policy11
Policy Map policy11
  Class prec1
    bandwidth 300000 (kbps)
  Class prec2
    bandwidth 200000 (kbps)
  Class prec3
    bandwidth 100000 (kbps)

```

When a queuing class is configured without any explicit share/bandwidth configuration, because the queue is not guaranteed any minimum bandwidth, the hardware queue is programmed to get a share of any unallocated bandwidth on the port as shown in the following example.

If there is no bandwidth remaining for the new queue or if the unallocated bandwidth is not sufficient to meet the minimum configurable rate (32kbps) for all queues which do not have any explicit share/bandwidth configuration, then the policy association is rejected.

For example, if there are two queues as given below

```

policy-map queue-policy
  class q1
    bandwidth percent 10

  class q2
    bandwidth percent 20

```

then the bandwidth allocation for the queues is as follows

```

q1 = 10%
q2 = 20%

```



```
class-default = 70%
```

Similarly, when another queuing class (say q3) is added without any explicit bandwidth (say, just a shape command), then the bandwidth allocation is

```
q1 = 10%
      q2 = 20%
      q3 = min(35%, q3-shape-rate)
class-default = max(35%, (100 - (q1 + q2 + q3 )))
```

Priority queuing

Only one transmit queue on a port can be configured as *strict priority* (termed Low Latency Queue, or LLQ).

LLQ provides strict-priority queuing for a traffic class. It enables delay-sensitive data, such as voice, to be sent *before* packets in other queues. The priority queue is serviced first until it is empty or until it is under its shape rate. Only one traffic stream can be destined for the priority queue per class-level policy. You enable the priority queue for a traffic class with the **priority policy-map class** configuration command at the class mode.

A LLQ can starve other queues unless it is rate limited. The supervisor engine does not support *conditional policing* where a 2-parameter policer (rate, burst) becomes effective when the queue is *congested* (based on queue length). However, it supports application of an unconditional policer to rate limit packets enqueued to the strict priority queue.

When a priority queue is configured on one class of a policy map, only *bandwidth remaining* is accepted on other classes, guaranteeing a minimum bandwidth for other classes from the remaining bandwidth of what is left after using the priority queue. When a priority queue is configured with a policer, then either *bandwidth* or *bandwidth remaining* is accepted on other classes.



Note

Use *bandwidth* or *bandwidth remaining* on all classes. You cannot apply *bandwidth* on one class and *bandwidth remaining* on another class within a policy map.

To enable class-level priority queuing in a service policy, follow these steps:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# policy-map <i>policy-map-name</i>	Creates a policy map by entering the policy-map name, and enter policy-map configuration mode. By default, no policy maps are defined.
Step 3	Switch(config-pmap)# class <i>class-name</i>	Specifies the name of the class whose traffic policy you want to create or change, and enter policy-map class configuration mode. By default, no traffic classes are defined.
Step 4	Switch(config-pmap-class)# priority	Enables the strict-priority queue, and give priority to a class of traffic. By default, strict-priority queueing is disabled.
Step 5	Switch(config-pmap-class)# exit	Returns to policy-map configuration mode.
Step 6	Switch(config-pmap)# exit	Returns to global configuration mode.
Step 7	Switch(config)# interface <i>interface-id</i>	Specifies a physical port and enter interface configuration mode.

	Command	Purpose
Step 8	Switch(config-interface)# service-policy output <i>policy-map-name</i>	Specifies the policy-map name, and apply it a physical interface.
Step 9	Switch(config-interface)# end	Returns to privileged EXEC mode.
Step 10	Switch# show policy-map [<i>policy-map-name</i> [class <i>class-map-name</i>]] or Switch# show policy-map interface <i>interface-id</i>	Verifies your entries.
Step 11	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To delete an existing policy map, use the **no policy-map** *policy-map-name* global configuration command. To delete an existing class, use the **no class** *class-name* policy-map configuration command. To disable the priority queue, use the **no priority** *policy-map-class* configuration command.

This example shows how to configure a class-level policy called policy1. Class 1 is configured as the priority queue, which is serviced first until it is empty.

```
Switch# configure terminal
Switch(config)# policy-map policy1
Switch(config-pmap)# class class1
Switch(config-pmap-c)# priority
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1
Switch(config-if)# service-policy output policy1
Switch(config-if)# end
Switch #

Switch# show policy-map policy1
  Policy Map policy1
    Class class1
      priority
```

Queue-limiting

When a class-based queue is instantiated on a physical port, it is set up with a default size. This size represents the number of queue entries in which packets belonging to that class of traffic can be queued. The scheduler moves packets from the queue that are ready for transmission, based on the queue shape, bandwidth, and priority configuration.

The queue-limit provides the maximum number of packets that can be in the queue at any given time. When the queue is full, an attempt to enqueue any further packets results in tail drop. However, if dynamic buffer limiting (DBL) is enabled on the queue, packets get a probabilistic drop based on the DBL algorithm, even when the queue is not full.

The **queue-limit** command can be configured under a class only when queue scheduling, such as bandwidth, shape, or priority is already configured. The only exception to this requirement is the support of the stand-alone **queue-limit** command on the class-default class.

Queue Memory

The number of queue entries that can be allocated has to be a multiple of 8 and can range from 16 to 8184. When a class-based queue is instantiated on a physical port, it is given a default number of entries. This default queue size is based on the number of slots in the chassis and the number of front-panel ports in each slot.

Supervisor Engine 6-E, and Supervisor Engine 6L-E have 512 K (524,288) queue entries of which the system sets aside 100 K (102,400) queue entries in a free reserve pool. Of the remaining 412 K (421,88), the drop port is provided 8184 entries and the CPU ports are assigned 11704 entries. Supervisor Engine 9-E, 8-E, and 7-E have 1M (1,048,576) queue entries of which the system sets aside 100K (102,400) queue entries in a free reserve pool. Of the remaining queue entries, the drop port is provided 8184 entries, 24576 entries for recirculation ports and the CPU ports are assigned 8656 entries. Supervisor Engine 7L-E has a 512 K queue entries.

The remaining entries are divided equally among the slots in the chassis. In a redundant chassis the two supervisor slots are treated as one for the purpose of this entries distribution. Within each slot the number of queue entries are equally divided among the front-panel ports present on the line card in that slot.

When the user configuration for queue entries on an interface exceeds its dedicated quota, the system attempts to satisfy the configuration from the free reserve pool. The entries from the free reserve pool are allocated to interfaces on a first-come first-served basis.

Service Policy Association

When a QoS service-policy with queuing actions is configured, but no explicit queue-limit command is attached in the egress direction on a physical interface, each of the class-based queues gets the same number of queue entries from within the dedicated quota for that physical port. When a queue is explicitly given a size using the queue-limit command, the switch tries to allocate all the entries from within the dedicated quota for the interface. If the required number of entries is greater than the dedicated quota for the interface, the switch tries to allocate the entries from the free reserve.

The queue entries associated with a queue always have to be consecutive. This requirement can result in fragmentation of the 512K of the queue entries that are shared across the switch. For example, an interface may not have enough entries for a queue in its dedicated quota and thus have to use the free reserve to set up that queue. In this case, the queue entries from the dedicated quota remain unused because they cannot be shared with any other port or slot.

When the QoS service-policy associated with an interface is removed, any queue entries taken from the free reserve are returned to the free reserve pool. The interface queuing configuration reverts to two queues — class-default and the control-packet queue with default shape, bandwidth, and size. The control-packet queue is set up with size 16 whereas the default queue is set up with the maximum size possible based on the dedicated quota for that interface.

Queue Allocation Failure

The switch might not be able to satisfy the explicit queue size required on one or more queues on an interface because of fragmentation of queue memory or lack of enough free reserve entries. In this scenario, the switch logs an error message to notify you of the failure. The QoS service-policy is left configured on the interface. You can fix the error by removing the QoS service-policy and examining the current usage of the queue entries from the free reserve by other ports on the switch.



Note

Do not attach a QoS policy with the maximum queue-limit (8184) to a large number of targets in a VSS system. This will cause continuous reloads on the standby supervisor engine.

To configure class-level queue-limit in a service policy, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# policy-map <i>policy-map-name</i>	Creates a policy map by entering the policy-map name, and enter policy-map configuration mode. By default, no policy maps are defined.
Step 3	Switch(config-pmap)# class <i>class-name</i>	Specifies the name of the class whose traffic policy you want to create or change, and enter policy-map class configuration mode. By default, no traffic classes are defined.
Step 4	Switch(config-pmap-class)# shape average { <i>cir-bps</i> [<i>optional_postfix</i>] percent <i>percent</i> }	Enables average-rate traffic shaping. You can specify the shaping rate in absolute value or as a percentage: <ul style="list-style-type: none"> For <i>cir-bps</i> [<i>optional_postfix</i>], specify the shaping rate in bps. Range is 32000 to 10000000000 bps. Supply an optional postfix (K, M, G). For <i>percent</i>, specify the percentage of link rate to shape the class of traffic. The range is 1 to 100. By default, average-rate traffic shaping is disabled.
Step 5	Switch(config-pmap-class)# queue-limit <i>number-of-packets</i>	Provides an explicit queue size in packets. The size must be a multiple of 8 and ranging from 16 to 8184.
Step 6	Switch(config-pmap-class)# exit	Returns to policy-map configuration mode.
Step 7	Switch(config-pmap)# exit	Returns to global configuration mode.
Step 8	Switch(config)# interface <i>interface-id</i>	Specifies a physical port and enter interface configuration mode.
Step 9	Switch(config-interface)# service-policy output <i>policy-map-name</i>	Specifies the policy-map name, and apply it a physical interface.
Step 10	Switch(config-interface)# end	Returns to privileged EXEC mode.
Step 11	Switch# show policy-map [<i>policy-map-name</i> [class <i>class-map-name</i>]] or Switch# show policy-map interface <i>interface-id</i>	Verifies your entries.
Step 12	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To remove the explicit queue size use the **no queue-limit** command under the class in a policy-map.

This example shows how to configure a class-based queue with an explicit **queue-limit** command. It limits traffic class class1 to a queue of size 4048:

```
Switch# configure terminal
Switch(config)# policy-map policy1
Switch(config-pmap)# class class1
Switch(config-pmap-c)# shape average 256000
```

```

Switch(config-pmap-c) # queue-limit 4048
Switch(config-pmap-c) # exit
Switch(config-pmap) # exit
Switch(config) # interface gigabitethernet1/1
Switch(config-if) # service-policy output policy1
Switch(config-if) # end
Switch#

Switch# show policy-map policy1
  Policy Map policy1
    Class class1
      shape average 256000
      queue-limit 4048
Switch#

```

Active Queue Management (AQM) via Dynamic Buffer Limiting (DBL)

AQM provides buffering control of traffic flows prior to queuing a packet into a transmit queue of a port. This is of significant interest in a shared memory switch, ensuring that certain flows do not hog the switch packet memory.



Note

The supervisor engine supports active switch buffer management via DBL.

Except for the default class of traffic (class class-default), you can configure DBL action only when at least one of the other queuing action is configured.

To configure class-level dbf action along with shaping in a service policy, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config) # policy-map <i>policy-map-name</i>	Creates a policy map by entering the policy-map name, and enter policy-map configuration mode. By default, no policy maps are defined.
Step 3	Switch(config-pmap) # class <i>class-name</i>	Specifies the name of the class whose traffic policy you want to create or change, and enter policy-map class configuration mode. By default, no traffic classes are defined.
Step 4	Switch(config-pmap-class) # shape average <i>cir-bps</i>	Enables average-rate traffic shaping. Specify the committed information rate, the bit rate that traffic is shaped to, in bps. The range is 32000 to 10000000000 bps. By default, average-rate traffic shaping is disabled.
Step 5	Switch(config-pmap-class) # dbf	Enables DBL on the queue associated with this class of traffic
Step 6	Switch(config-pmap-class) # exit	Returns to policy-map configuration mode.
Step 7	Switch(config-pmap) # exit	Returns to global configuration mode.
Step 8	Switch(config) # interface <i>interface-id</i>	Specifies a physical port and enter interface configuration mode.
Step 9	Switch(config-interface) # service-policy output <i>policy-map-name</i>	Specifies the policy-map name, and apply it a physical interface.
Step 10	Switch(config-interface) # end	Returns to privileged EXEC mode.

	Command	Purpose
Step 11	<pre>Switch# show policy-map [<i>policy-map-name</i> [<i>class</i> <i>class-map-name</i>]] or Switch# show policy-map interface <i>interface-id</i></pre>	Verifies your entries.
Step 12	<pre>Switch# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

To delete an existing policy map, use the **no policy-map** *policy-map-name* global configuration command. To delete an existing class, use the **no class** *class-name* policy-map configuration command. To disable DBL on the associated queue, use the **no db1** policy-map class configuration command.

The following example shows how to configure class-level, DBL action along with average-rate shaping. It enables DBL on the queue associated with traffic-class *class1*.

```
Switch# configure terminal
Switch(config)# policy-map policy1
Switch(config-pmap)# class class1
Switch(config-pmap-c)# shape average 256000
Switch(config-pmap-c)# db1
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitEthernet1/1
Switch(config-if)# service-policy output policy1
Switch(config-if)# end
Switch#

Switch# show policy-map policy1
  Policy Map policy1
    Class class1
      shape average 256000
      db1
```

Transmit Queue Statistics

Transmit queue statistics are visible by entering the **show policy-map interface** command:

```
Switch# show policy-map interface gigabitEthernet 1/1
GigabitEthernet1/1

  Service-policy output: queuing-policy

    Class-map: queuing-class (match-all)
      1833956 packets
      Match: cos 1
      Queueing
        (total drops) 1006239
        (bytes output) 56284756
        shape (average) cir 320000000, bc 1280000, be 1280000
        target shape rate 320000000

    Class-map: class-default (match-any)
      1 packets
      Match: any

        (total drops) 0
        (bytes output) 2104
```

Enabling Per-Port Per-VLAN QoS

The per-port per-VLAN QoS feature enables you to specify different QoS configurations on different VLANs on a given interface. Typically, you use this feature on trunk or voice VLANs (Cisco IP Phone) ports, as they belong to multiple VLANs.

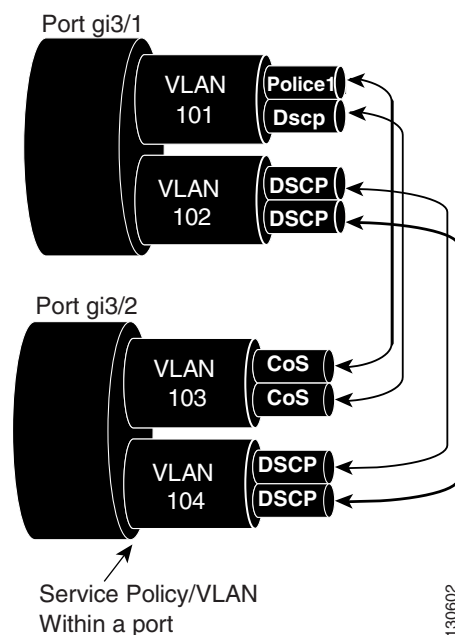
To configure per-port per-VLAN QoS, perform this task:

	Command	Purpose
Step 1	Switch(config)# interface { fastethernet gigabitethernet tengigabitethernet } slot/interface Port-channel <i>number</i>	Selects the interface to configure.
Step 2	Switch(config-if)# vlan-range <i>vlan_range</i>	Specifies the VLANs involved.
Step 3	Switch(config-if-vlan-range)# service-policy { input output } <i>policy-map</i>	Specifies the policy-map and direction.
Step 4	Switch(config-if-vlan-range)# exit	Exits class-map configuration mode.
Step 5	Switch(config-if)# end	Exits configuration interface mode.
Step 6	Switch# show policy-map interface <i>interface_name</i>	Verifies the configuration.

Example 1

Figure 44-6 displays a sample topology for configuring PVQoS. The trunk port gi3/1 is comprised of multiple VLANs (101 and 102). Within a port, you can create your own service policy per VLAN. This policy, performed in hardware, might consist of ingress and egress Policing or giving precedence to voice packet over data.

Figure 44-4 Per-Port Per-VLAN Topology



The following configuration file shows how to perform ingress and egress policing per VLAN using the policy-map P31_QOS applied to port Gigabit Ethernet 3/1:

```
ip access-list 101 permit ip host 1.2.2.2 any
```

```

ip access-list 103 permit ip any any
Class-map match-all RT

match ip access-group 101
Class-map Match all PD

match ip access-group 103
Policy-map P31_QoS

Class RT

Police 200m 16k conform transmit exceed drop

Class PD

Police 100m 16k conform transmit exceed drop

Interface Gigabit 3/1
Switchport
Switchport trunk encapsulation dot1q
Switchport trunk allowed vlan 101-102
  Vlan range 101
    Service-policy input P31_QoS
    Service-policy output P31_QoS
  Vlan range 102
    Service-policy input P32_QoS
    Service-policy output P32_QoS

```

Example 2

Let us assume that interface Gigabit Ethernet 6/1 is a trunk port and belongs to VLANs 20, 300-301, and 400. The following example shows how to apply policy-map p1 for traffic in VLANs 20 and 400 and policy map p2 to traffic in VLANs 300 through 301:

```

Switch# configure terminal
Switch(config)# interface gigabitethernet 6/1
Switch(config-if)# vlan-range 20,400
Switch(config-if-vlan-range)# service-policy input p1
Switch(config-if-vlan-range)# exit
Switch(config-if)# vlan-range 300-301
Switch(config-if-vlan-range)# service-policy output p2
Switch(config-if-vlan-range)# end
Switch#

```

Example 3

The following command shows how to display policy-map statistics on VLAN 20 configured on Gigabit Ethernet interface 6/1:

```

Switch# show policy-map interface gigabitEthernet 6/1 vlan 20

GigabitEthernet6/1 vlan 20

Service-policy input: p1

Class-map: c1 (match-all)
  0 packets
  Match: cos 1
  Match: access-group 100
  police:
    cir 100000000 bps, bc 3125000 bytes

```



```

conformed 0 bytes; actions:
  transmit
exceeded 0 bytes; actions:
  drop
conformed 0000 bps, exceed 0000 bps

Class-map: class-default (match-any)
  0 packets
  Match: any

```

Example 4

The following command shows how to display policy-map statistics on all VLANs configured on Gigabit Ethernet interface 6/1:

```
Switch# show policy-map interface gigabitEthernet 6/1
```

```
GigabitEthernet6/1 vlan 20
```

```
Service-policy input: p1
```

```

Class-map: c1 (match-all)
  0 packets
  Match: cos 1
  Match: access-group 100
  police:
    cir 100000000 bps, bc 3125000 bytes
    conformed 0 bytes; actions:
      transmit
    exceeded 0 bytes; actions:
      drop
    conformed 0000 bps, exceed 0000 bps

```

```

Class-map: class-default (match-any)
  0 packets
  Match: any

```

```
GigabitEthernet6/1 vlan 300
```

```
Service-policy output: p2
```

```

Class-map: c1 (match-all)
  0 packets
  Match: cos 1
  Match: access-group 100
  QoS Set
    dscp 50
  police:
    cir 200000000 bps, bc 6250000 bytes
    conformed 0 bytes; actions:
      transmit
    exceeded 0 bytes; actions:
      drop
    conformed 0000 bps, exceed 0000 bps

```

```

Class-map: class-default (match-any)
  0 packets
  Match: any

```

```
GigabitEthernet6/1 vlan 301
```

```
Service-policy output: p2
```

```

Class-map: c1 (match-all)
  0 packets
  Match: cos 1
  Match: access-group 100
  QoS Set
    dscp 50
  police:
    cir 200000000 bps, bc 6250000 bytes
    conformed 0 bytes; actions:
      transmit
    exceeded 0 bytes; actions:
      drop
    conformed 0000 bps, exceed 0000 bps

```

Policy Associations

The supervisor engine supports per-port, per-VLAN policies. The associated policies are attached to the interface, VLAN, and a specific VLAN on a given port, respectively.

A policy can be associated with a variety of objects. The following table lists the objects and the actions allowed.

Table 44-1 Table QoS Policy Associations

Object	Action
Physical port	Policing, marking, and queuing
VLAN	Policing and marking
Port and VLAN (PV)	Policing and marking
EtherChannel	Policing and marking
EtherChannel member port	Queuing

Qos Action Restrictions

- The same actions cannot be performed multiple times in a given direction on different targets. In other words, it is not possible to police the packets both on port and VLAN in the input direction. However, the user can police on the input port and on the output VLAN.
- Queuing actions are only allowed in the egress direction and only on the physical port.
- Percentage-based actions like policer cannot be configured on a VLAN, Port and VLAN (PV) and EtherChannel.
- Port channel or VLAN configuration can only have a policing or a marking action, not a queueing action.

Qos Policy priorities

- If a policy on a port and a VLAN are configured with conflicting actions (such as policing or marking actions on both a port and VLAN), the port policy is picked.
- If policy on a VLAN on a given port must be over-written, the user can configure PV policy.

Qos Policy merging

Applicable policies are applied to a given packet in given direction. For example, if you configure egress VLAN-based police and marking, followed by selective queuing on the port, then actions from both policies will be applied for this packet.

The following policy-map configuration restrictions are imposed on an EtherChannel:

- only policing and marking actions are supported at the EtherChannel level
- only queuing actions are supported at the physical member port level

A packet can be marked (dscp or cos fields) by the EtherChannel policy. If the physical member port policy uses a classification based on dscp or cos fields, it must be based on the marked (modified) value. To ensure proper operation, the following restriction is placed on the EtherChannel.

The classification criteria for the policy-map on the physical member ports has to be based only on one type of field:

- dscp
- precedence
- cos
- any non marking field (no dscp or cos based classification)

Classification criteria for the policy-map on the physical member ports cannot be based on a combination of fields. This restriction ensures that if the EtherChannel policy is marking down dscp or cos, the marked (modified) value-based classification can be implemented in hardware.

**Note**

Auto-QoS macros with SRND4 generate class-maps with more than one type of match. These class-maps need to be modified to use only with one matching type when applied on EtherChannel member ports.

**Note**

Classification criteria for the policy-map on the physical member ports cannot be modified to add a new type of field.

Auto-QoS is not supported on EtherChannel or its member ports. A physical port configured with Auto-QoS is not allowed to become a member of a physical port.

Software QoS

At the highest level, there are two types of locally sourced traffic (such as control protocol packets, pings, and telnets) from the switch: high priority traffic (typically the control protocol packets like OSPF Hellos and STP) and low priority packets (all other packet types).

The QoS treatment for locally-sourced packets differs for the two types.

The supervisor engine provides a way to apply QoS to packets processed in the software path. The packets that get this QoS treatment in software can be classified into two types: software switched packets and software generated packets.

On reception, software switched packets are sent to the CPU that in turn sends them out of another interface. For such packets, input software QoS provides input marking and output software QoS provides output marking and queue selection.

The software generated packets are the ones locally sourced by the switch. The type of output software QoS processing applied to these packets is the same as the one applied to software switched packets. The only difference in the two is that the software switched packets take input marking of the packet into account for output classification purpose.

High Priority Packets

High priority packets are marked as one of the following:

- internally with PAK_PRIORITY
- with IP Precedence of 6 (for IP packets)
- with CoS of 6 (for VLAN Tagged packets)

These packets behave as follows:

- They are not dropped because of any policing, AQM, drop thresholds (or any feature that can drop a packet) configured as per the egress service policy. However, they might be dropped because of hardware resource constraints (packet buffers, queue full, etc.).
- They are classified and marked as per the marking configuration of the egress service policy that could be a port or VLAN (refer to the [“Policy Associations” section on page 44-74](#)).
- These high priority packets are enqueued to queue on the egress port based on the following criteria:
 - If there is no egress queuing policy on the port, the packet is queued to a control packet queue that is setup separately from the default queue and has 5 percent of the link bandwidth reserved for it.
 - If there is an egress queuing policy on the port, the queue is selected based on the classification criteria applicable to the packet.

Low Priority Packets

Packets that are not considered high priority (as described previously) are considered *unimportant*. These include locally sourced pings, telnet, and other protocol packets. They undergo the same treatment as any other packet that is transiting the given transmit port including egress classification, marking and queuing.

Applying Flow-based QoS Policy

Flow based QoS enables microflow policing and marking capability to dynamically learn traffic flows. It also rate limits each unique flow to an individual rate. Flow based QoS is available with the built-in NetFlow hardware support.

For more overview information, refer to the [“Flow-based QoS” section on page 44-10](#).

The following steps show how to apply Flow based QoS policy to QoS targets:

-
- | | |
|---------------|---|
| Step 1 | Create a FNF flow record by specifying the key fields that identify unique flows. You can use any FNF flow records that are associated with the FNF monitor. |
| Step 2 | Create a class-map to specify the set of match criteria. Include the FNF flow record from Step 1 in the class-map match criteria using the match flow record command. Then, configure the class-map to match all the match criteria with class-map match-all class_name . |
| Step 3 | Create a policy-map and define actions associated with class-map from Step 2. |

Step 4 Attach the policy to one or more QoS targets.

Examples

The following examples illustrate how to configure Flow based QoS policy and apply microflow policers on individual flows.

Example 1

This example assumes there are multiple users (identified by source IP address) on the subnet 192.168.10.*. The configuration below shows how to configure a flow based QoS policy that uses micro policing to limit the per-user traffic with the source address in the range of 192.168.10.*. The microflow policer is configured with a CIR of 1Mbps, “conform action” as transmit, and “exceed action” as drop.

Step 1: Define an ACL to match traffic with specified source address.

```
Switch(config)# ip access-list extended UserGroup1
Switch(config-ext-nacl)# permit ip 192.168.10.0 0.0.0.255 any
Switch(config-ext-nacl)# exit
Switch(config)#
```

Step 2: Define a flow record to create flows with source address as key.

```
Switch(config)# flow record r1
Switch(config-flow-record)# match ipv4 source address
Switch(config-flow-record)# exit
Switch(config)#
```

Step 3: Configure classmap to match on the UserGroup1 and specify flow record definition for flow creation.

```
Switch(config)# class-map match-all c1
Switch(config-cmap)# match access-group name UserGroup1
Switch(config-cmap)# match flow record r1
Switch(config-cmap)# exit
Switch(config)#
```

Step 4: Configure flow based QoS policy-map with microflow policing action for the matching traffic.

```
Switch(config)# policy-map p1
Switch(config-pmap)# class c1
Switch(config-pmap-c)# police cir 1m
Switch(config-pmap-c-police)# conform-action transmit
Switch(config-pmap-c-police)# exceed-action drop
Switch(config-pmap-c-police)# exit
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
```

Step 5: Attach flow QoS policy to the interface.

```
Switch(config)# interface gigabitEthernet3/1
Switch(config-if)# service-policy input p1
Switch(config-if)#
```

Use the **show** commands (described in the policy and marking sections of this chapter) to display the policy-map configuration and interface specific policy-map statistics.

Example 2.

This example assumes there are multiple users (identified by source IP address) on subnets 192.168.10.* and 172.20.55.*. The first requirement is to police with a CIR of 500Kbps and a PIR of 650Kbps on any TCP traffic originating from 192 network to any destination at any given time. The **exceed action** keyword marks down the dscp value to 32. The second requirement is to police per-user traffic originating from 172 network to CIR of 2Mbps and unconditionally mark the traffic with dscp 10.

Step 1: Define an ACL to match traffic with specified source address.

```
Switch(config)# ip access-list extended UserGroup1
Switch(config-ext-nacl)# permit ip 19 2.168.10.0 0.0.0.255 any
Switch(config-ext-nacl)# exit
Switch(config)# ip access-list extended UserGroup2
Switch(config-ext-nacl)# permit ip 172.20.55.0 0.0.0.255 any
Switch(config-ext-nacl)# exit
Switch(config)#
```

Step 2: Define a flow record to create flows with source address as key.

```
Switch(config)# flow record r1
Switch(config-flow-record)# match ipv4 source address
Switch(config-flow-record)# match ipv4 destination address
Switch(config-flow-record)# match transport tcp source-port
Switch(config-flow-record)# match transport tcp destination-port
Switch(config-flow-record)# exit
Switch(config)# flow record r2
Switch(config-flow-record)# match ipv4 source address
Switch(config-flow-record)# exit
Switch(config)#
```

Step 3: Configure classmap to match on the UserGroup1 and specify flow record definition for flow creation.

```
Switch(config)# class-map match-all c1
Switch(config-cmap)# match access-group name UserGroup1
Switch(config-cmap)# match flow record r1
Switch(config-cmap)# exit
Switch(config)# class-map match-all c2
Switch(config-cmap)# match access-group name UserGroup2
Switch(config-cmap)# match flow record r2
Switch(config-cmap)# exit
Switch(config)#
```

Step 4: Configure flow based QoS policy-map with microflow policing action for the matching traffic.

```
Switch(config)# policy-map p1
Switch(config-pmap)# class c1
Switch(config-pmap-c)# police cir 500k pir 650k
Switch(config-pmap-c-police)# conform-action transmit
Switch(config-pmap-c-police)# exceed-action set-dscp-transmit 32
Switch(config-pmap-c-police)# violate-action drop
Switch(config-pmap-c-police)# exit
Switch(config-pmap-c)# exit
Switch(config-pmap)# class c2
Switch(config-pmap-c)# set dscp 10
Switch(config-pmap-c)# police cir 2m
Switch(config-pmap-c-police)# conform-action transmit
Switch(config-pmap-c-police)# exceed-action drop
Switch(config-pmap-c-police)# exit
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
```

Step 5: Attach flow QoS policy to the interface.

```
Switch(config)# interface gigabitEthernet3/1
Switch(config-if)# service-policy input p1
Switch(config-if)# exit
```

Use the show commands described in the QoS section to display the policy-map configuration and interface specific policy-map statistics.

Example 3

Assume that there are two active flows on FastEthernet interface 6/1:

Table 44-2

SrcIp	DStIp	IPProt	SrcL4Port	DstL4Port
192.168.10.10	192.168.20.20	20	6789	81
192.168.10.10	192.168.20.20	20	6789	21

With the following configuration, each flow is policed to 1000000 bps with an allowed 9000 burst value.

```
Switch(config)# flow record r1
Switch(config-flow-record)# match ipv4 source address
Switch(config-flow-record)# match ipv4 destination address
Switch(config-flow-record)# match transport tcp source-port
Switch(config-flow-record)# match transport tcp destination-port
Switch(config-flow-record)# match transport udp source-port
Switch(config-flow-record)# match transport udp destination-port
Switch(config-flow-record)# exit
Switch(config)# class-map match-all c1
Switch(config-cmap)# match flow record r1
Switch(config-cmap)# exit
Switch(config)# policy-map p1
Switch(config-pmap)# class c1
Switch(config-pmap-c)# police 1000000 9000
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface fastEthernet 6/1
Switch(config-if)# service-policy input p1
Switch(config-if)# end
```

Configuration Guidelines

The general guidelines for creating, configuring, modifying, deleting a flow based QoS policy and attaching (and detaching) a flow based QoS policy to a supported target is the same as described in the QoS section. The following description and restriction applies to Flow based QoS policy:

- A classmap can have multiple match statements but only one FNF flow record can be specified in a class-map.
- A flow record must have at least one key field before it can be used in a classmap. Non-key fields can be present in the flow record. However, all the non-key fields are ignored by microflow QoS. Only key-fields are used for flow creation.
- If a FNF flow record is referenced in any class-map, the flow record cannot be modified. Remove the flow record from all classmaps before modifying it.
- A classmap with a FNF flow record must be configured as **match-all**; traffic hitting the class-map must satisfy all match criteria in the class-map.

- A policy can contain multiple classes and each class-map may contain the same or different FNF flow record.
- Flow based QoS policy and FNF monitor both cannot be applied on the same target at the same time.
- When the interface mode changes from switchport to routed port and vice versa, any Flow QoS policy attached to the port remains applied after the mode change.
- There are 3 types of FNF flow records: ipv4, ipv6, and datalink. The datalink flow record is mutually exclusive with the ipv4 and ipv6 flow records; a classmap with the datalink flow record cannot co-exist with classmap having a ipv4 or ipv6 flow record in the same policy and vice-versa.
- Classmap class-default is not editable; it cannot be configured with the match flow record. Instead, you can configure the policy with a class-map that uses a match any filter and the flow record.
- Traffic is classified in the same order in which class-map is defined in a policy. Hence, if a FNF flow record is the only match statement in a class-map, the classifier matches all packets of the type identified by the flow record. This means that any subsequent class-map in the same policy matching on the same traffic type will be redundant and will never be hit.
- Policers associated with classmap having flow record are called *microflow policers*. The CIR and PIR rates for microflow policers cannot be configured using the percent keyword.
- Flow records within the same policy must include appropriate key fields to ensure flows created from different classmaps are unique and distinct. Otherwise, the resulting flows from different classmap cannot be distinguished. In such cases, policy actions corresponding to the classmap which created the first flow in hardware will apply and results will not be always be as expected.
- Flows from traffic received on different QoS targets are distinct even if the same policy is applied to those targets.
- A flow is aged out if it is inactive for more than 5 seconds; there is no traffic matching the flow for a period longer than 5 sec.
- When a flow is aged out, policer state information associated with the flow is also deleted. When a new flow is created, the policer instance for the flow is re-initialized.
- Flows created by flow based QoS policy exist in hardware only and cannot be exported (as with FNF monitor).
- Per-flow statistics are not available for flows created by flow based QoS policy.
- Class-map statistics indicate the number of packets matching the classifier. It does not represent individual flow stats.
- Policer statistics show the aggregate policer statistics of individual flow.
- Information about the flows created by hardware are not available and not displayed in the show commands associated with QoS policy-map. Only class-map and policer statistics are displayed in the output of the **show policy-map** commands.

Configuring CoS Mutation

CoS reflection and CoS mutation are supported on Supervisor Engine 6-E. Below is an example of how to apply CoS reflection.

Let us say that traffic arrives on interface gigabit 2/5 with VLAN 10 and COS 1, 2, We want traffic to egress interface gigabit 2/6 with outer tag VLAN 11 and CoS copied from C-tag, where C-tag is VLAN 10 and COS 1, 2, ...

```
class-map match-all c2
```



```

        match cos 2

class-map match-all c1

    match cos 1

!

policy-map my

class c1

    set cos 1

class c2

    set cos 2

interface GigabitEthernet2/5

switchport mode trunk

switchport vlan mapping 10 dot1q-tunnel 11

spanning-tree bpdupfilter enable

service-policy input my

!

interface GigabitEthernet2/6

switchport mode trunk

```

Configuring System Queue Limit



Note

This feature is available only from Cisco IOS Release 15.0(2)SG1 and later and Cisco IOS Release XE 3.2.1SG.

With the **hw-module system max-queue-limit** command, the Catalyst 4500 series switch allows you to change the queue limit for all interfaces globally, instead of applying a policy with queue limit to all the interfaces.

To set the queue limit globally, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# hw-module system max-queue-limit <i>max-queue-limit</i>	Sets the queue limit for all interfaces globally. Valid values are from 1024 to 8184. The value must be a multiple of 8.

	Command	Purpose
Step 3	Switch(config)# exit	Returns to privileged EXEC mode.
Step 4	Switch# reload	Reloads standalone supervisor engine.
	or	
	Switch# redundancy reload shelf	Reloads redundancy supervisor engine in SSO mode.
	Switch# redundancy force-switchover	Reloads redundancy supervisor engine in RPR mode. This command must be followed by another redundancy force-switchover.

This is a global configuration command. You can override it with the per port, per class, **queue-limit** command.

For a standalone supervisor engine, you must reboot the engine after applying this command.

For redundant supervisors in SSO mode, you must enter the **redundancy reload shelf** command enforce reboot to both the supervisors. For redundancy supervisors in RPR mode, you must execute two consecutive switchovers to enforce the system queue limit on both the supervisors.

This example shows how to set the queue limit globally to 1024 on a standalone supervisor engine:

```
Switch> enable
Switch# configure terminal
Switch(config)# hw-module system max-queue-limit 1024
Switch(config)# exit
Switch# reload (for standalone supervisors)
Switch# redundancy reload shelf (for redundancy supervisors in SSO mode)
or
Switch# redundancy force-switchover (followed by another redundancy force-switchover, for
redundancy supervisors in RPR mode)
```

Configuring QoS on a Standalone Supervisor Engine 6-E, 6L-E or Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, and 7-E



Note

HQoS is not supported on the Catalyst 4500 series switch.

Topics include:

- [MQC-based QoS Configuration, page 44-49](#)
- [Platform-supported Classification Criteria and QoS Features, page 44-15](#)
- [Platform Hardware Capabilities, page 44-50](#)
- [Prerequisites for Applying a QoS Service Policy, page 44-50](#)
- [Restrictions for Applying a QoS Service Policy, page 44-51](#)
- [Classification, page 44-51](#)
- [Policing, page 44-52](#)
- [Marking Network Traffic, page 44-53](#)
- [Shaping, Sharing \(Bandwidth\), Priority Queuing, Queue-limiting and DBL, page 44-60](#)

- [Enabling Per-Port Per-VLAN QoS, page 44-71](#)
- [Applying Flow-based QoS Policy, page 44-76](#)
- [Configuring CoS Mutation, page 44-80](#)
- [Configuring System Queue Limit, page 44-81](#)

MQC-based QoS Configuration



Note

Starting with Cisco IOS Release 12.2(40)SG, Catalyst 4500 series switches with Supervisor Engine 6-E or Supervisor Engine 6L-E use the MQC model of QoS.

To apply QoS, you use the Modular QoS Command-Line Interface (MQC), which is a CLI structure that allows you to complete the following tasks:

- Specify the matching criteria used to define a traffic class.
- Create a traffic policy (policy map). The traffic policy defines the QoS policy actions to be taken for each traffic class.
- Apply the policy actions specified in the policy map to an interface, VLAN, or port and VLAN.

For more information about the MQC, see the “Modular Quality of Service Command-Line Interface” section of the *Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.3*.



Note

The incoming traffic is considered trusted by default. Only when the *trusted boundary* feature is enabled on an interface can the port enter untrusted mode. In this mode, the switch marks the DSCP value of an IP packet and the CoS value of the VLAN tag on the Ethernet frame as “0”.

Platform-supported Classification Criteria and QoS Features

The following table provides a summary of various classification criteria and actions supported on the Catalyst 4500 series switch. For details, refer to the *Catalyst 4500 Series Switch Command Reference*.

Supported classification actions	Descriptions
match access-group	Configures the match criteria for a class map on the basis of the specified ACL.
match any	Configures the match criteria for a class map to be successful match criteria for all packets.
match cos	Matches a packet based on a Layer 2 class of service (CoS) marking.
match [ip] dscp	Identifies a specific IP differentiated service code point (DSCP) value as a match criterion. Up to eight DSCP values can be included in one match statement.
match [ip] precedence	Identifies IP precedence values as match criteria.
match protocol	Configures the match criteria for a class map on the basis of the specified protocol.
match qos-group	Identifies a specific QoS group value as a match criterion. Applies only on the egress direction.
Supported Qos Features	Descriptions

Supported classification actions	Descriptions
police	Configures traffic policing.
police (percent)	Configures traffic policing on the basis of a percentage of bandwidth available on an interface.
police (two rates)	Configures traffic policing using two rates, the committed information rate (CIR) and the peak information rate (PIR).
set cos	Sets the Layer 2 class of service (CoS) value of an outgoing packet.
set dscp	Marks a packet by setting the differentiated services code point (DSCP) value in the type of service (ToS) byte of IPv4 or traffic class byte of IPv6 packet.
set precedence	Sets the precedence value in the packet header.
set qos-group	Sets a QoS group identifier (ID) that can be used later to classify packets.
table map support	Unconditional marking of one packet field based on another packet field.
priority	Gives priority to a class of traffic belonging to a policy map.
shape	Shapes traffic to the indicated bit rate according to the algorithm specified.
bandwidth	Provides a guaranteed minimum bandwidth to each of the eight queues.
dbl	Dynamic buffer limit.
queue-limit	Specifies the maximum number of packets a transmit queue can hold.

Platform Hardware Capabilities

Qos Actions	Numbers of entries supported
Classification	64k input and 64k output classification entries are supported. A given policy can use at most 24k ACLs
Policing	16K policers are supported. Policers are allocated to given direction in blocks of 2k. For example, 2k policers can be used in for input and 14k policers can be used for output. Single rate policers uses one policer entry. Single Rate Three Color Marker (srTCM) (RFC 2697) and Two Rate Three Color Marker (trTCM) (RFC 2698) uses two policer entries
Marking	Marking of Cos and DSCP/Precedence is supported through two marking tables, each capable of supporting 512 entries. There are separate tables for each direction.
Queuing	The queue size is Configurable with the maximum number of entries configurable per port depending on the chassis and line card type.
DBL	You can enable DBL action on all configured class-maps.

Prerequisites for Applying a QoS Service Policy

Unlike the Switch QoS model, there is no prerequisite for enabling QoS on various targets. Just the attachment of a service policy enables QoS and detachment of that policy disables QoS on that target.

Restrictions for Applying a QoS Service Policy

Traffic marking can be configured on an interface, a VLAN, or a port and VLAN. An interface can be a Layer 2 access port, a Layer 2 switch trunk, a Layer 3 routed port, or an EtherChannel. A policy is attached to a VLAN using *vlan configuration* mode.

Attaching QoS service policy to VLANs and EtherChannel is described in the [“Policy Associations” section on page 44-74](#).

Classification

The supervisor engine supports classification of Layer 2, IP, IPv6 packets, and ARP packets marking performed on input can be matched in the output direction. The previous table lists the full set of capabilities. By default, the switch also supports classification resources sharing. Similarly, when the same policy is attached to a port or a VLAN or on per-port per-vlan targets, ACL entries are shared though QoS actions are unique on each target.

For example:

```
class-map c1
  match ip dscp 50

Policy Map p1
  class c1
    police rate 1 m burst 200000
```

If policy-map p1 is applied to interfaces Gig 1/1 and Gig 1/2, 1 CAM entry is used (one ACE that matches IP packets), but 2 policers are allocated (one per target). So, all IP packets with dscp 50 are policed to 1 mbps on interface Gig 1/1 and packets on interface Gig 1/2 are policed to 1 mbps.

**Note**

With Cisco IOS Release 12.2(46)SG, you can issue the **match protocol arp** command. For details, see the *Catalyst 4500 Series Switch Cisco IOS Command Reference*.

Classification Statistics

The supervisor engine supports only packet based classification statistics and TCAM resource sharing. When a policy-map is applied on multiple targets, the command **show policy-map interface** displays the aggregate classification statistics, not those specific to an interface.

**Note**

To obtain per interface policy-map stats, you should configure a unique policy-map name on each interface.

When a policy-map is attached to a port-channel member ports, classification statistics are not displayed.

Configuring a Policy Map

You can attach only one policy map to an interface. Policy maps can contain one or more policy-map classes, each with different match criteria and actions.

Configure a separate policy-map class in the policy map for each type of traffic that an interface receives. Put all commands for each type of traffic in the same policy-map class. QoS does not attempt to apply commands from more than one policy-map class to matched traffic.

Creating a Policy Map

To create a policy map, enter this command:

Command	Purpose
Switch(config)# [no] policy-map <i>policy_name</i>	Creates a policy map with a user-specified name. Use the no keyword to delete the policy map.

Attaching a Policy Map to an Interface

To create a policy map, enter this command:

Command	Purpose
Switch(config)# interface {vlan <i>vlan_ID</i> { fastethernet gigabitethernet } <i>slot/interface</i> Port-channel <i>number</i> }	Selects the interface to configure.
Switch(config-if)# [no] service-policy input <i>policy_map_name</i>	Attaches a policy map to the input direction of the interface. Use the no keyword to detach a policy map from an interface.
Switch(config-if)# end	Exits configuration mode.
Switch# show policy-map interface {vlan <i>vlan_ID</i> { fastethernet gigabitethernet } <i>slot/interface</i> }	Verifies the configuration.

Policing

The supervisor engine supports policers in the following operation modes:

- Single Rate Policer Two Color Marker

This kind of policer is configured with just the committed rate (CIR) and normal burst and it has only conform and exceed actions.

- Single Rate Three Color Marker (srTCM) (RFC 2697)
- Two Rate Three Color Marker (trTCM) (RFC 2698)
- Color Blind Mode

Policing accuracy of 0.75% of configured policer rate.

The engine supports 16384 (16 x 1024, 16K) single rate, single burst policers. 16K policers are organized as 8 banks of 2K policers. The policer banks are dynamically assigned (input or output policer bank) by the software depending on the QoS configuration. So, the 16K policers are dynamically partitioned by software as follows:

- 0 Input Policers and 16K Output Policers
- 2K Input Policers and 14K Output Policers
- 4K Input Policers and 12K Output Policers
- 6K Input Policers and 10K Output Policers
- 8K Input Policers and 8K Output Policers

- 10K Input Policers and 6K Output Policers
- 12K Input Policers and 4K Output Policers
- 14K Input Policers and 2K Output Policers
- 16K Input Policers and 0 Output Policers

These numbers represent individual policer entries in the hardware that support a single rate and burst parameter. Based on this, a switch supports the following number of policers:

- 16K Single Rate Policer with Single Burst (Two Color Marker)
- 8K Single Rate Three Color Marker (srTCM)
- 8K Two Rate Three Color Marker (trTCM)

These policers are partitioned between Input and Output in chunks of 2K policer banks. The different types of policers can all co-exist in the system. However, a given type of policer (srTCM, trTCM etc.) is configurable as a block of 128 policers.

**Note**

Two policers are reserved for internal use.

How to Implement Policing

For details on how to implement the policing features on a Catalyst 4500 Series Switch, refer to the Cisco IOS documentation at the following link:

http://www.cisco.com/en/US/docs/ios/12_2/qos/configuration/guide/qcfpolsh.html

Platform Restrictions

Platform restrictions include the following:

- Multi-policer actions can be specified (setting CoS and IP DSCP is supported).
- When unconditional marking and policer based marking exists on the same field(cos or dscp or precedence), policer-based marking is preferred.
- If policer based service-policy is attached to both a port and a VLAN, port-based policed is preferred by default. To over-ride a specific VLAN policy on a given port, then you must configure a per-port per-vlan policy.
- You should not delete a port-channel with a per-port, per-VLAN QoS policy.

Workaround: Before deleting the port-channel, do the following:

1. Remove any per-port per-VLAN QoS policies, if any.
2. Remove the VLAN configuration on the port-channel with the **no vlan-range** command.

Marking Network Traffic

Marking network traffic allows you to set or modify the attributes of traffic (that is, packets) belonging to a specific class or category. When used in conjunction with network traffic classification, marking network traffic is the foundation for enabling many quality of service (QoS) features on your network. This module contains conceptual information and the configuration tasks for marking network traffic.

Contents

- “Information About Marking Network Traffic” section on page 44-54
- “Marking Action Drivers” section on page 44-56
- “Traffic Marking Procedure Flowchart” section on page 44-56
- “Restrictions for Marking Network Traffic” section on page 44-57
- “Multi-attribute Marking Support” section on page 44-57
- “Hardware Capabilities for Marking” section on page 44-24
- “Configuring the Policy Map Marking Action” section on page 44-58
- “Marking Statistics” section on page 44-60

Information About Marking Network Traffic

To mark network traffic, you should understand the following concepts:

- “Purpose of Marking Network Traffic” section on page 44-54
- “Benefits of Marking Network Traffic” section on page 44-54
- “Two Methods for Marking Traffic Attributes” section on page 44-55

Purpose of Marking Network Traffic

Traffic marking is used to identify certain traffic types for unique handling, effectively partitioning network traffic into different categories.

After the network traffic is organized into classes by traffic classification, traffic marking allows you to mark (that is, set or change) a value (attribute) for the traffic belonging to a specific class. For instance, you may want to change the class of service (CoS) value from 2 to 1 in one class, or you may want to change the differentiated services code point (DSCP) value from 3 to 2 in another class. In this module, these values are referred to as attributes or marking fields.

Attributes that can be set and modified include the following:

- CoS value of a tagged Ethernet frame
- DSCP/Precedence value in the Type of Service (ToS) byte of IPv4.
- QoS group identifier (ID)
- DSCP /Precedence value in the traffic class byte of IPv6

Benefits of Marking Network Traffic

Traffic marking allows you to fine-tune the attributes for traffic on your network. This increased granularity helps isolate traffic that requires special handling, and thus, helps to achieve optimal application performance.

Traffic marking allows you to determine how traffic will be treated, based on how the attributes for the network traffic are set. It allows you to segment network traffic into multiple priority levels or classes of service based on those attributes, as follows:

- Traffic marking is often used to set the IP precedence or IP DSCP values for traffic entering a network. Networking devices within your network can then use the newly marked IP precedence values to determine how traffic should be treated. For example, voice traffic can be marked with a

particular IP precedence or DSCP and strict priority can then be configured to put all packets of that marking into that queue. In this case, the marking was used to identify traffic for strict priority queue.

- Traffic marking can be used to identify traffic for any class-based QoS feature (any feature available in policy map class configuration mode, although some restrictions exist).

Two Methods for Marking Traffic Attributes



Note

This section describes *Unconditional* marking, which differs from *Policer-based* marking. Unconditional marking is based solely on classification.

Method One: Unconditional Explicit Marking (using the set command)

You specify the traffic attribute you want to change with a set command configured in a policy map. The following table lists the available set commands and the corresponding attribute. For details on the set command, refer to the *Catalyst 4500 Series Switch Command Reference*.

Table 44-4 *set Commands and Applicable Packet Types*

set Commands	Traffic Attribute	Packet Type
set cos	Layer 2 CoS value of the outgoing traffic	Ethernet IPv4, IPv6
set dscp	DSCP value in the ToS byte	IPv4, IPv6
set precedence	precedence value in the packet header	IPv4, IPv6
set qos-group	QoS group ID	Ethernet, IPv4, IPv6

If you are using individual **set** commands, those set commands are specified in a policy map. The following is a sample of a policy map configured with one of the set commands listed in [Table 44-4](#).

In this sample configuration, the **set cos** command has been configured in the policy map (policy1) to mark the CoS attribute:

```
enable
configure terminal
policy map p1
  class class1
    set cos 3
end
```

For information on configuring a policy map, see the “[Creating a Policy Map](#)” section on page 44-52.

The final task is to attach the policy map to the interface. For information on attaching the policy map to the interface, see the “[Attaching a Policy Map to an Interface](#)” section on page 44-52.

Method Two: Unconditional Tablemap-based Marking

You can create a table map that can be used to mark traffic attributes. A table map is a kind of two-way conversion chart that lists and maps one traffic attribute to another. A table map supports a many-to-one type of conversion and mapping scheme. The table map establishes a to-from relationship for the traffic attributes and defines the change to be made to the attribute. That is, an attribute is set to one value that is taken from another value. The values are based on the specific attribute being changed. For instance, the Precedence attribute can be a number from 0 to 7, while the DSCP attribute can be a number from 0 to 63.

The following is a sample table map configuration:

```

table-map table-map1
map from 0 to 1
map from 2 to 3
exit

```

The following table lists the traffic attributes for which a to-from relationship can be established using the table map.

Table 44-5 Traffic Attributes for Which a To-From Relationship Can Be Established

The "To" Attribute	The "From" Attribute
Precedence	CoS, QoS group, DSCP, Precedence
DSCP	COS, QoS group, DSCP, Precedence
CoS	DSCP, QoS group, CoS, Precedence

The following is an example of a policy map (policy2) configured to use the table map (table-map1) created earlier:

```

Policy map policy
  class class-default
    set cos dscp table table-map
exit

```

In this example, a mapping relationship was created between the CoS attribute and the DSCP attribute as defined in the table map.

For information on configuring a policy map to use a table map, [“Configuring a Policy Map” section on page 44-51](#).

The final task is to attach the policy map to the interface. For information on attaching the policy map to the interface, see the [“Attaching a Policy Map to an Interface” section on page 44-52](#).

Marking Action Drivers

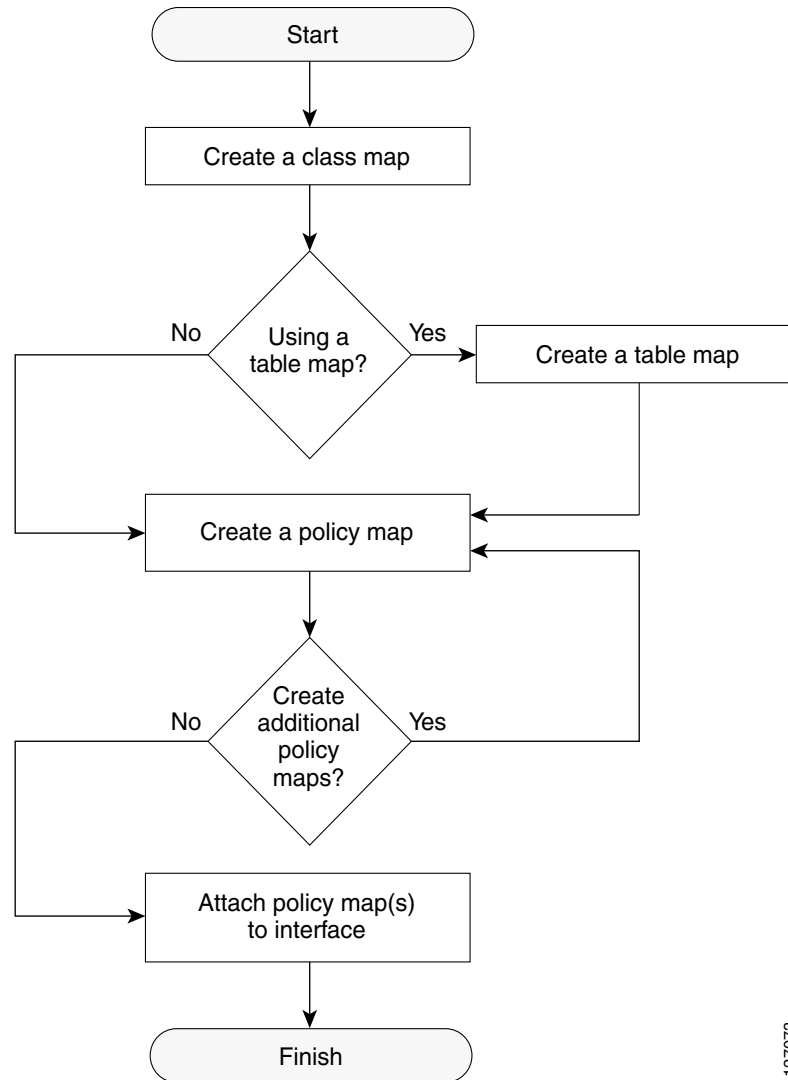
A marking action can be triggered based on one of the two QoS processing steps.

Classification based: In this case, all the traffic matching a class is marked using either explicit or tablemap based method. This method is referred to as *unconditional* marking.

Policer result-based: In this case, a class of traffic is marked differently based on the policer result (conform/exceed/violate) applicable to that packet. This method is referred to as *conditional* marking.

Traffic Marking Procedure Flowchart

[Figure 44-5](#) illustrates the order of the procedures for configuring traffic marking.

Figure 44-5 Traffic marking Procedure Flowchart

127073

Restrictions for Marking Network Traffic

The following restrictions apply to packet marking actions:

- QoS-group can be marked only in the input direction and can only support unconditional explicit marking.
- Only explicit marking is supported for policer-based marking.

Multi-attribute Marking Support

The supervisor engine can mark more than one QoS attribute of a packet matching a class of traffic. For example, DSCP, CoS, and QoS-group can all be set together, using either explicit or tablemap-based marking.

**Note**

When using unconditional explicit marking of multiple fields or policer-based multi-field, multi-region (conform/exceed/violate) marking the number of tablemaps that can be setup in TOS or COS marking tables will be less than the maximum supported.

Hardware Capabilities for Marking

Supervisor Engine 6-E, and Supervisor Engine 6L-E provide a 128 entry marking action (Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, and 7-E provide a 256 entry marking action) where each entry specifies the type of marking actions on CoS and DSCP/Precedence fields as well as policer action to transmit/markdown/drop a packet.

One such table is supported for each direction, input and output. This table is used for both unconditional marking as well as policer-based marking. It can be used to support 256 unique marking actions or 64 unique policer-based actions or a combinations of the two.

For each of the marking fields (COS and DSCP), the supervisor engine provides 512 entry marking tables for each direction. These are similar to mapping tables available on supervisor engines that support the switch QoS model. However, these provide an ability to have multiple unique mapping tables that are setup by the user.

For example, the TOS marking table provides marking of DSCP/Precedence fields and can be used as one of the following:

- 8 different tablemaps with each mapping the 64 DSCP or qos-group values to another DSCP
- 64 (32) different tablemaps with each one mapping 8 CoS (16 CoS and CFi) values to DSCP in input (output) direction
- a combination of above two types of tablemaps

Similar mappings are available on the 512 entry COS marking table.

Configuring the Policy Map Marking Action

This section describes how to establish unconditional marking action for network traffic.

As a prerequisites, create a class map (*ipp5*) and a policy map. (Refer to the [“Configuring a Policy Map” section on page 44-51](#)).

**Note**

The marking action command options have been extended (refer to [Table 44-4 on page 44-55](#) and [Table 44-5 on page 44-56](#)).

Configuring Tablemap-based Unconditional Marking

To configure table-map based unconditional marking, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# table-map name	Configures a tablemap.
Step 3	Switch(config-tablemap)# map from from_value to to_value	Creates a map from a <i>from_value</i> to a <i>to_value</i>

	Command	Purpose
Step 4	Switch(config-tablemap)# exit	Exits table-map configuration mode.
Step 5	Switch(config)# policy-map <i>name</i>	Enters policy-map configuration mode.
Step 6	Switch(config-p)# class <i>name</i>	Selects the class for QoS actions.
Step 7	Switch(config-p-c)# set cos dscp prec cos dscp prec qos-group [table <i>name</i>]	Selects the marking action based on an implicit or explicit table-map.
Step 8	Switch(config-p-c)# end	Exits configuration mode.
Step 9	Switch# show policy-map <i>name</i>	Verifies the configuration of the policy-map.
Step 10	Switch# show table-map <i>name</i>	Verifies the configuration of the table-map.

The following example shows how to enable marking action using table-map.

```
Switch(config)# table-map dscp2Cos
Switch(config-tablemap)# map from 8 to 1
Switch(config-tablemap)# exit
Switch(config)# policy-map p1
Switch(config-pmap)# class ip5
Switch(config-pmap-c)# set cos dscp table dscp2Cos
Switch(config-pmap-c)# end
Switch# show policy-map p1

Policy Map p1
  Class ip5
    set cos dscp table dscp2Qos

Switch# show table-map dscp2Cos

Table Map dscp2Cos
  from 8 to 1
  default copy
```

Configuring Policer Result-based Conditional Marking

To configure policer result-based conditional marking, setup a single rate or dual rate policer. Refer to the [“How to Implement Policing”](#) section on page 44-53.

This example shows how to configure a two rate three-color policer with explicit actions for each policer region:

```
Switch# configure terminal
Switch(config-pmap-c)# policer cir percent 20 pir percent 30
Switch(config-pmap-c-policer)# conform-action set-cos-transmit 3 set-dscp-transmit 10
Switch(config-pmap-c-policer)# exceed-action set-cos-transmit 4 set-dscp-transmit 20
Switch(config-pmap-c-policer)# violate action drop
Switch# show policy-map p1

Policy Map police
  Class ip5
    police cir percent 20 pir percent 30
      conform-action set-cos-transmit 3
      conform-action set-dscp-transmit af11
      exceed-action set-cos-transmit 4
      exceed-action set-dscp-transmit af22
      violate-action drop
```

Marking Statistics

The marking statistics indicate the number of packets that are *marked*.

For unconditional marking, the *classification entry* points to an entry in the marking action table that in turn indicates the fields in the packet that are marked. Therefore, the classification statistics by itself indicates the unconditional marking statistics.

For a conditional marking using policer, provided the policer is a packet rate policer, you cannot determine the number packets marked because the policer only provides byte statistics for different policing results.

Shaping, Sharing (Bandwidth), Priority Queuing, Queue-limiting and DBL

The Catalyst 4500 series switch supports the Classification-based (class-based) mode for transmit queue selection. In this mode, the transmit queue selection is based on the Output QoS classification lookup.



Note

Only output (egress) queuing is supported.

The supervisor engine supports 8 transmit queues per port. Once the forwarding decision has been made to forward a packet out a port, the output QoS classification determines the transmit queue into which the packet needs to be enqueued.

By default, without any service policies associated with a port, there are two queues (a control packet queue and a default queue) with no guarantee as to the bandwidth or kind of prioritization. The only exception is that system generated control packets are enqueued into control packet queue so that control traffic receives some minimum link bandwidth.

Queues are assigned when an output policy attached to a port with one or more queuing related actions for one or more classes of traffic. Because there are only eight queues per port, there can be at most eight classes of traffic (including the reserved class, class-default) with queuing action(s). Classes of traffic that do not have any queuing action are referred to as *non-queuing* classes. Non-queuing class traffic ends up using the queue corresponding to class class-default.

When a queuing policy (a policy with queuing action) is attached, the control packet queue is deleted and the control packets are enqueued into respective queue per their classification. An egress QoS class must be configured to match IP Precedence 6 and 7 traffic, and a bandwidth guarantee must be configured.

Dynamic resizing of queues (queue limit class-map action) is supported through the use of the **queue-limit** command. Based on the chassis and line card type, all eight queues on a port are configured with equal queue size.

Shaping

Shaping enables you to delay out-of-profile packets in queues so that they conform to a specified profile. Shaping is distinct from policing. Policing drops packets that exceed a configured threshold, whereas shaping *buffers* packets so that traffic remains within a given threshold. Shaping offers greater *smoothness* in handling traffic than policing. You enable average-rate traffic shaping on a traffic class with the **policy-map** class configuration command.

The supervisor engine supports a range of 32kbps to 10 gbps for shaping, with a precision of approximately +/- 0.75 per cent.

When a queuing class is configured without any explicit shape configuration, the queue shape is set to the link rate.

To configure class-level shaping in a service policy, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# policy-map <i>policy-map-name</i>	Creates a policy map by entering the policy-map name, and enter policy-map configuration mode. By default, no policy maps are defined.
Step 3	Switch(config-pmap)# class <i>class-name</i>	Specifies the name of the class whose traffic policy you want to create or change, and enter policy-map class configuration mode. By default, no traffic classes are defined.
Step 4	Switch(config-pmap-class)# shape average { <i>cir-bps</i> [<i>optional_postfix</i>] percent <i>percent</i> }	Enables average-rate traffic shaping. You can specify the shaping rate in absolute value or as a percentage: <ul style="list-style-type: none"> For <i>cir-bps</i> [<i>optional_postfix</i>], specify the shaping rate in bps. Range is 32000 to 10000000000 bps. Supply an optional postfix (K, M, G). For <i>percent</i>, specify the percentage of link rate to shape the class of traffic. The range is 1 to 100. By default, average-rate traffic shaping is disabled.
Step 5	Switch(config-pmap-class)# exit	Returns to policy-map configuration mode.
Step 6	Switch(config-pmap)# exit	Returns to global configuration mode.
Step 7	Switch(config)# interface <i>interface-id</i>	Specifies a physical port and enter interface configuration mode.
Step 8	Switch(config-interface)# service-policy output <i>policy-map-name</i>	Specifies the policy-map name, and apply it a physical interface.
Step 9	Switch(config-interface)# end	Returns to privileged EXEC mode.
Step 10	Switch# show policy-map [<i>policy-map-name</i> [class <i>class-map-name</i>]] or Switch# show policy-map interface <i>interface-id</i>	Verifies your entries.
Step 11	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To delete an existing policy map, use the **no policy-map** *policy-map-name* global configuration command. To delete an existing class, use the **no class** *class-name* policy-map configuration command. To disable the average-rate traffic shaping, use the **no shape average** policy-map class configuration command.

This example shows how to configure class-level, average-rate shaping. It limits traffic class class1 to a data transmission rate of 256 kbps:

```
Switch# configure terminal
Switch(config)# policy-map policy1
Switch(config-pmap)# class class1
Switch(config-pmap-c)# shape average 256000
```

```
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1
Switch(config-if)# service-policy output policy1
Switch(config-if)# end
Switch#
```

```
Switch# show policy-map policy1
Policy Map policy1
Class class1
  shape average 256000
```

This example shows how to configure class-level, average shape percentage to 32% of link bandwidth for queuing-class traffic:

```
Switch# configure terminal
Switch(config)# policy-map queuing-policy
Switch(config-pmap)# class queuing-class
Switch(config-pmap-c)# shape average percent 32
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1
Switch(config-if)# service-policy output queuing-policy1
Switch(config-if)# end
Switch #
```

```
Switch# show policy-map queuing-policy
Policy Map queuing-policy
Class queuing-class
  Average Rate Traffic Shaping
  cir 32%
```

Sharing(bandwidth)

The bandwidth assigned to a class of traffic is the minimum bandwidth that is guaranteed to the class during congestion. Transmit Queue Sharing is the process by which output link bandwidth is shared among multiple queues of a given port.

The supervisor engine supports a range of 32 kbps to 10 gbps for sharing, with a precision of approximately +/- 0.75 per cent. The sum of configured bandwidth across all queuing classes should not exceed the link bandwidth.

To configure class-level bandwidth action in a service policy, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# policy-map <i>policy-map-name</i>	Creates a policy map by entering the policy-map name, and enter policy-map configuration mode. By default, no policy maps are defined.
Step 3	Switch(config-pmap)# class <i>class-name</i>	Specifies the name of the class whose traffic policy you want to create or change, and enter policy-map class configuration mode. By default, no traffic classes are defined.

	Command	Purpose
Step 4	Switch(config-pmap-class)# bandwidth { <i>bandwidth-kbps</i> percent <i>percent</i> }	Specifies the minimum bandwidth provided to a class belonging to the policy map when there is traffic congestion in the switch. If the switch is not congested, the class receives more bandwidth than you specify with the bandwidth command. By default, no bandwidth is specified. You can specify the bandwidth in kbps or as a percentage: o For <i>bandwidth-kbps</i> , specify the bandwidth amount in kbps assigned to the class. The range is 32 to 10000000. o For <i>percent</i> , specify the percentage of available bandwidth assigned to the class. The range is 1 to 100. Specify all the class bandwidths in either kbps or in percentages, but not a mix of both.
Step 5	Switch(config-pmap-class)# exit	Returns to policy-map configuration mode.
Step 6	Switch(config-pmap)# exit	Returns to global configuration mode.
Step 7	Switch(config)# interface <i>interface-id</i>	Specifies a physical port and enter interface configuration mode.
Step 8	Switch(config-interface)# service-policy output <i>policy-map-name</i>	Specifies the policy-map name, and apply it a physical interface.
Step 9	Switch(config-interface)# end	Returns to privileged EXEC mode.
Step 10	Switch# show policy-map [<i>policy-map-name</i> [class <i>class-map-name</i>]] or Switch# show policy-map interface <i>interface-id</i>	Verifies your entries.
Step 11	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To delete an existing policy map, use the **no policy-map** *policy-map-name* global configuration command. To delete an existing class, use the **no class** *class-name* policy-map configuration command. To return to the default bandwidth, use the **no bandwidth** policy-map class configuration command.

This example shows how to create a class-level policy map called policy11 for three classes called prec1, prec2, and prec3. In the policy for these classes, 30 percent of the available bandwidth is assigned to the queue for the first class, 20 percent is assigned to the queue for the second class, and 10 percent is assigned to the queue for the third class.

```
Switch # configure terminal
Switch(config)# policy-map policy11
Switch(config-pmap)# class prec1
Switch(config-pmap-c)# bandwidth percent 30
Switch(config-pmap-c)# exit
Switch(config-pmap)# class prec2
Switch(config-pmap-c)# bandwidth percent 20
Switch(config-pmap-c)# exit
Switch(config-pmap)# class prec3
Switch(config-pmap-c)# bandwidth percent 10
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1
Switch(config-if)# service-policy output policy11
```

```

Switch(config-if)# end
Switch #

Switch# show policy-map policy11
Policy Map policy11
  Class prec1
    bandwidth percent 30
  Class prec2
    bandwidth percent 20
  Class prec3
    bandwidth percent 10

```

This example shows how to create a class-level policy map called policy11 for three classes called prec1, prec2, and prec3. In the policy for these classes, 300 mbps of the available bandwidth is assigned to the queue for the first class, 200 mbps is assigned to the queue for the second class, and 100 mbps is assigned to the queue for the third class.

```

Switch # configure terminal
Switch(config)# policy-map policy11
Switch(config-pmap)# class prec1
Switch(config-pmap-c)# bandwidth 300000
Switch(config-pmap-c)# exit
Switch(config-pmap)# class prec2
Switch(config-pmap-c)# bandwidth 200000
Switch(config-pmap-c)# exit
Switch(config-pmap)# class prec3
Switch(config-pmap-c)# bandwidth 100000
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1
Switch(config-if)# service-policy output policy11
Switch(config-if)# end
Switch #

Switch# show policy-map policy11
Policy Map policy11
  Class prec1
    bandwidth 300000 (kbps)
  Class prec2
    bandwidth 200000 (kbps)
  Class prec3
    bandwidth 100000 (kbps)

```

When a queuing class is configured without any explicit share/bandwidth configuration, because the queue is not guaranteed any minimum bandwidth, the hardware queue is programmed to get a share of any unallocated bandwidth on the port as shown in the following example.

If there is no bandwidth remaining for the new queue or if the unallocated bandwidth is not sufficient to meet the minimum configurable rate (32kbps) for all queues which do not have any explicit share/bandwidth configuration, then the policy association is rejected.

For example, if there are two queues as given below

```

policy-map queue-policy
  class q1
    bandwidth percent 10

  class q2
    bandwidth percent 20

```

then the bandwidth allocation for the queues is as follows

```

q1 = 10%
q2 = 20%

```

```
class-default = 70%
```

Similarly, when another queuing class (say q3) is added without any explicit bandwidth (say, just a shape command), then the bandwidth allocation is

```
q1 = 10%
q2 = 20%
q3 = min(35%, q3-shape-rate)
class-default = max(35%, (100 - (q1 + q2 + q3)))
```

Priority queuing

Only one transmit queue on a port can be configured as *strict priority* (termed Low Latency Queue, or LLQ).

LLQ provides strict-priority queuing for a traffic class. It enables delay-sensitive data, such as voice, to be sent *before* packets in other queues. The priority queue is serviced first until it is empty or until it is under its shape rate. Only one traffic stream can be destined for the priority queue per class-level policy. You enable the priority queue for a traffic class with the **priority policy-map class** configuration command at the class mode.

A LLQ can starve other queues unless it is rate limited. The supervisor engine does not support *conditional policing* where a 2-parameter policer (rate, burst) becomes effective when the queue is *congested* (based on queue length). However, it supports application of an unconditional policer to rate limit packets enqueued to the strict priority queue.

When a priority queue is configured on one class of a policy map, only *bandwidth remaining* is accepted on other classes, guaranteeing a minimum bandwidth for other classes from the remaining bandwidth of what is left after using the priority queue. When a priority queue is configured with a policer, then either *bandwidth* or *bandwidth remaining* is accepted on other classes.



Note

Use *bandwidth* or *bandwidth remaining* on all classes. You cannot apply *bandwidth* on one class and *bandwidth remaining* on another class within a policy map.

To enable class-level priority queuing in a service policy, follow these steps:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# policy-map <i>policy-map-name</i>	Creates a policy map by entering the policy-map name, and enter policy-map configuration mode. By default, no policy maps are defined.
Step 3	Switch(config-pmap)# class <i>class-name</i>	Specifies the name of the class whose traffic policy you want to create or change, and enter policy-map class configuration mode. By default, no traffic classes are defined.
Step 4	Switch(config-pmap-class)# priority	Enables the strict-priority queue, and give priority to a class of traffic. By default, strict-priority queueing is disabled.
Step 5	Switch(config-pmap-class)# exit	Returns to policy-map configuration mode.
Step 6	Switch(config-pmap)# exit	Returns to global configuration mode.
Step 7	Switch(config)# interface <i>interface-id</i>	Specifies a physical port and enter interface configuration mode.

	Command	Purpose
Step 8	Switch(config-interface)# service-policy output <i>policy-map-name</i>	Specifies the policy-map name, and apply it a physical interface.
Step 9	Switch(config-interface)# end	Returns to privileged EXEC mode.
Step 10	Switch# show policy-map [<i>policy-map-name</i> [class <i>class-map-name</i>]] or Switch# show policy-map interface <i>interface-id</i>	Verifies your entries.
Step 11	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To delete an existing policy map, use the **no policy-map** *policy-map-name* global configuration command. To delete an existing class, use the **no class** *class-name* policy-map configuration command. To disable the priority queue, use the **no priority** *policy-map-class* configuration command.

This example shows how to configure a class-level policy called policy1. Class 1 is configured as the priority queue, which is serviced first until it is empty.

```
Switch# configure terminal
Switch(config)# policy-map policy1
Switch(config-pmap)# class class1
Switch(config-pmap-c)# priority
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1
Switch(config-if)# service-policy output policy1
Switch(config-if)# end
Switch #

Switch# show policy-map policy1
  Policy Map policy1
    Class class1
      priority
```

Queue-limiting

When a class-based queue is instantiated on a physical port, it is set up with a default size. This size represents the number of queue entries in which packets belonging to that class of traffic can be queued. The scheduler moves packets from the queue that are ready for transmission, based on the queue shape, bandwidth, and priority configuration.

The queue-limit provides the maximum number of packets that can be in the queue at any given time. When the queue is full, an attempt to enqueue any further packets results in tail drop. However, if dynamic buffer limiting (DBL) is enabled on the queue, packets get a probabilistic drop based on the DBL algorithm, even when the queue is not full.

The **queue-limit** command can be configured under a class only when queue scheduling, such as bandwidth, shape, or priority is already configured. The only exception to this requirement is the support of the stand-alone **queue-limit** command on the class-default class.

Queue Memory

The number of queue entries that can be allocated has to be a multiple of 8 and can range from 16 to 8184. When a class-based queue is instantiated on a physical port, it is given a default number of entries. This default queue size is based on the number of slots in the chassis and the number of front-panel ports in each slot.

Supervisor Engine 6-E, and Supervisor Engine 6L-E have 512 K (524,288) queue entries of which the system sets aside 100 K (102,400) queue entries in a free reserve pool. Of the remaining 412 K (421,88), the drop port is provided 8184 entries and the CPU ports are assigned 11704 entries. Supervisor Engine 7-E and Supervisor Engine 8-E have 1M (1,048,576) queue entries of which the system sets aside 100K (102,400) queue entries in a free reserve pool. Of the remaining queue entries, the drop port is provided 8184 entries, 24576 entries for recirculation ports and the CPU ports are assigned 8656 entries.

Supervisor Engine 7L-E has a 512 K queue entries.

The remaining entries are divided equally among the slots in the chassis. In a redundant chassis the two supervisor slots are treated as one for the purpose of this entries distribution. Within each slot the number of queue entries are equally divided among the front-panel ports present on the line card in that slot.

When the user configuration for queue entries on an interface exceeds its dedicated quota, the system attempts to satisfy the configuration from the free reserve pool. The entries from the free reserve pool are allocated to interfaces on a first-come first-served basis.

Service Policy Association

When a QoS service-policy with queuing actions is configured, but no explicit queue-limit command is attached in the egress direction on a physical interface, each of the class-based queues gets the same number of queue entries from within the dedicated quota for that physical port. When a queue is explicitly given a size using the queue-limit command, the switch tries to allocate all the entries from within the dedicated quota for the interface. If the required number of entries is greater than the dedicated quota for the interface, the switch tries to allocate the entries from the free reserve.

The queue entries associated with a queue always have to be consecutive. This requirement can result in fragmentation of the 512K of the queue entries that are shared across the switch. For example, an interface may not have enough entries for a queue in its dedicated quota and thus have to use the free reserve to set up that queue. In this case, the queue entries from the dedicated quota remain unused because they cannot be shared with any other port or slot.

When the QoS service-policy associated with an interface is removed, any queue entries taken from the free reserve are returned to the free reserve pool. The interface queuing configuration reverts to two queues — class-default and the control-packet queue with default shape, bandwidth, and size. The control-packet queue is set up with size 16 whereas the default queue is set up with the maximum size possible based on the dedicated quota for that interface.

Queue Allocation Failure

The switch might not be able to satisfy the explicit queue size required on one or more queues on an interface because of fragmentation of queue memory or lack of enough free reserve entries. In this scenario, the switch logs an error message to notify you of the failure. The QoS service-policy is left configured on the interface. You can fix the error by removing the QoS service-policy and examining the current usage of the queue entries from the free reserve by other ports on the switch.

To configure class-level queue-limit in a service policy, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# policy-map <i>policy-map-name</i>	Creates a policy map by entering the policy-map name, and enter policy-map configuration mode. By default, no policy maps are defined.
Step 3	Switch(config-pmap)# class <i>class-name</i>	Specifies the name of the class whose traffic policy you want to create or change, and enter policy-map class configuration mode. By default, no traffic classes are defined.
Step 4	Switch(config-pmap-class)# shape average { <i>cir-bps</i> [<i>optional_postfix</i>] percent <i>percent</i> }	Enables average-rate traffic shaping. You can specify the shaping rate in absolute value or as a percentage: <ul style="list-style-type: none"> For <i>cir-bps</i> [<i>optional_postfix</i>], specify the shaping rate in bps. Range is 32000 to 10000000000 bps. Supply an optional postfix (K, M, G). For <i>percent</i>, specify the percentage of link rate to shape the class of traffic. The range is 1 to 100. By default, average-rate traffic shaping is disabled.
Step 5	Switch(config-pmap-class)# queue-limit <i>number-of-packets</i>	Provides an explicit queue size in packets. The size must be a multiple of 8 and ranging from 16 to 8184.
Step 6	Switch(config-pmap-class)# exit	Returns to policy-map configuration mode.
Step 7	Switch(config-pmap)# exit	Returns to global configuration mode.
Step 8	Switch(config)# interface <i>interface-id</i>	Specifies a physical port and enter interface configuration mode.
Step 9	Switch(config-interface)# service-policy output <i>policy-map-name</i>	Specifies the policy-map name, and apply it a physical interface.
Step 10	Switch(config-interface)# end	Returns to privileged EXEC mode.
Step 11	Switch# show policy-map [<i>policy-map-name</i> [class <i>class-map-name</i>]] or Switch# show policy-map interface <i>interface-id</i>	Verifies your entries.
Step 12	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To remove the explicit queue size use the **no queue-limit** command under the class in a policy-map.

This example shows how to configure a class-based queue with an explicit **queue-limit** command. It limits traffic class class1 to a queue of size 4048:

```
Switch# configure terminal
Switch(config)# policy-map policy1
Switch(config-pmap)# class class1
Switch(config-pmap-c)# shape average 256000
Switch(config-pmap-c)# queue-limit 4048
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1
Switch(config-if)# service-policy output policy1
```

```

Switch(config-if)# end
Switch#

Switch# show policy-map policy1
  Policy Map policy1
    Class class1
      shape average 256000
      queue-limit 4048
Switch#

```

Active Queue Management (AQM) via Dynamic Buffer Limiting (DBL)

AQM provides buffering control of traffic flows prior to queuing a packet into a transmit queue of a port. This is of significant interest in a shared memory switch, ensuring that certain flows do not hog the switch packet memory.



Note

The supervisor engine supports active switch buffer management via DBL.

Except for the default class of traffic (class class-default), you can configure DBL action only when at least one of the other queuing action is configured.

To configure class-level dbf action along with shaping in a service policy, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# policy-map <i>policy-map-name</i>	Creates a policy map by entering the policy-map name, and enter policy-map configuration mode. By default, no policy maps are defined.
Step 3	Switch(config-pmap)# class <i>class-name</i>	Specifies the name of the class whose traffic policy you want to create or change, and enter policy-map class configuration mode. By default, no traffic classes are defined.
Step 4	Switch(config-pmap-class)# shape average <i>cir-bps</i>	Enables average-rate traffic shaping. Specify the committed information rate, the bit rate that traffic is shaped to, in bps. The range is 32000 to 10000000000 bps. By default, average-rate traffic shaping is disabled.
Step 5	Switch(config-pmap-class)# dbl	Enables DBL on the queue associated with this class of traffic
Step 6	Switch(config-pmap-class)# exit	Returns to policy-map configuration mode.
Step 7	Switch(config-pmap)# exit	Returns to global configuration mode.
Step 8	Switch(config)# interface <i>interface-id</i>	Specifies a physical port and enter interface configuration mode.
Step 9	Switch(config-interface)# service-policy output <i>policy-map-name</i>	Specifies the policy-map name, and apply it a physical interface.
Step 10	Switch(config-interface)# end	Returns to privileged EXEC mode.

	Command	Purpose
Step 11	<pre>Switch# show policy-map [<i>policy-map-name</i> [<i>class</i> <i>class-map-name</i>]] or Switch# show policy-map interface <i>interface-id</i></pre>	Verifies your entries.
Step 12	<pre>Switch# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

To delete an existing policy map, use the **no policy-map** *policy-map-name* global configuration command. To delete an existing class, use the **no class** *class-name* policy-map configuration command. To disable DBL on the associated queue, use the **no db1** policy-map class configuration command.

The following example shows how to configure class-level, DBL action along with average-rate shaping. It enables DBL on the queue associated with traffic-class *class1*.

```
Switch# configure terminal
Switch(config)# policy-map policy1
Switch(config-pmap)# class class1
Switch(config-pmap-c)# shape average 256000
Switch(config-pmap-c)# db1
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitEthernet1/1
Switch(config-if)# service-policy output policy1
Switch(config-if)# end
Switch#

Switch# show policy-map policy1
Policy Map policy1
  Class class1
    shape average 256000
    db1
```

Transmit Queue Statistics

Transmit queue statistics are visible by entering the **show policy-map interface** command:

```
Switch# show policy-map interface gigabitEthernet 1/1
GigabitEthernet1/1

Service-policy output: queuing-policy

Class-map: queuing-class (match-all)
  1833956 packets
  Match: cos 1
  Queueing
    (total drops) 1006239
    (bytes output) 56284756
  shape (average) cir 320000000, bc 1280000, be 1280000
  target shape rate 320000000

Class-map: class-default (match-any)
  1 packets
  Match: any

    (total drops) 0
    (bytes output) 2104
```


Enabling Per-Port Per-VLAN QoS

The per-port per-VLAN QoS feature enables you to specify different QoS configurations on different VLANs on a given interface. Typically, you use this feature on trunk or voice VLANs (Cisco IP Phone) ports, as they belong to multiple VLANs.

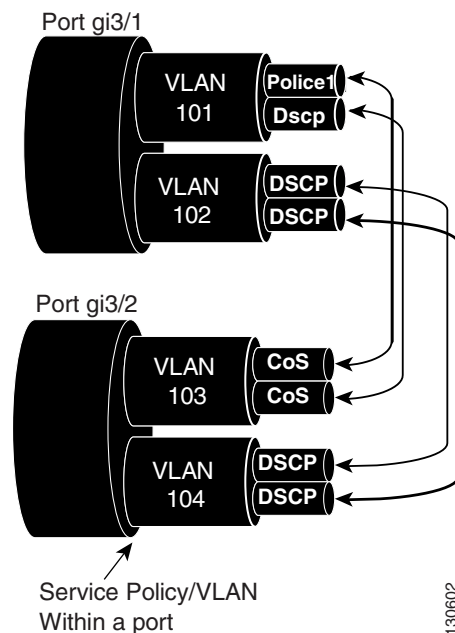
To configure per-port per-VLAN QoS, perform this task:

	Command	Purpose
Step 1	Switch(config)# interface { fastethernet gigabitethernet tengigabitethernet } slot/interface Port-channel <i>number</i>	Selects the interface to configure.
Step 2	Switch(config-if)# vlan-range <i>vlan_range</i>	Specifies the VLANs involved.
Step 3	Switch(config-if-vlan-range)# service-policy { input output } <i>policy-map</i>	Specifies the policy-map and direction.
Step 4	Switch(config-if-vlan-range)# exit	Exits class-map configuration mode.
Step 5	Switch(config-if)# end	Exits configuration interface mode.
Step 6	Switch# show policy-map interface <i>interface_name</i>	Verifies the configuration.

Example 1

Figure 44-6 displays a sample topology for configuring PVQoS. The trunk port gi3/1 is comprised of multiple VLANs (101 and 102). Within a port, you can create your own service policy per VLAN. This policy, performed in hardware, might consist of ingress and egress Policing or giving precedence to voice packet over data.

Figure 44-6 Per-Port Per-VLAN Topology



The following configuration file shows how to perform ingress and egress policing per VLAN using the policy-map P31_QOS applied to port Gigabit Ethernet 3/1:

```
ip access-list 101 permit ip host 1.2.2.2 any
```

```

ip access-list 103 permit ip any any
Class-map match-all RT

match ip access-group 101
Class-map Match all PD

match ip access-group 103
Policy-map P31_QoS

Class RT

Police 200m 16k conform transmit exceed drop

Class PD

Police 100m 16k conform transmit exceed drop

Interface Gigabit 3/1
Switchport
Switchport trunk encapsulation dot1q
Switchport trunk allowed vlan 101-102
  Vlan range 101
    Service-policy input P31_QoS
    Service-policy output P31_QoS
  Vlan range 102
    Service-policy input P32_QoS
    Service-policy output P32_QoS

```

Example 2

Let us assume that interface Gigabit Ethernet 6/1 is a trunk port and belongs to VLANs 20, 300-301, and 400. The following example shows how to apply policy-map p1 for traffic in VLANs 20 and 400 and policy map p2 to traffic in VLANs 300 through 301:

```

Switch# configure terminal
Switch(config)# interface gigabitEthernet 6/1
Switch(config-if)# vlan-range 20,400
Switch(config-if-vlan-range)# service-policy input p1
Switch(config-if-vlan-range)# exit
Switch(config-if)# vlan-range 300-301
Switch(config-if-vlan-range)# service-policy output p2
Switch(config-if-vlan-range)# end
Switch#

```

Example 3

The following command shows how to display policy-map statistics on VLAN 20 configured on Gigabit Ethernet interface 6/1:

```

Switch# show policy-map interface gigabitEthernet 6/1 vlan 20

GigabitEthernet6/1 vlan 20

Service-policy input: p1

Class-map: c1 (match-all)
  0 packets
  Match: cos 1
  Match: access-group 100
  police:
    cir 100000000 bps, bc 3125000 bytes

```

```

conformed 0 bytes; actions:
  transmit
exceeded 0 bytes; actions:
  drop
conformed 0000 bps, exceed 0000 bps

Class-map: class-default (match-any)
  0 packets
  Match: any

```

Example 4

The following command shows how to display policy-map statistics on all VLANs configured on Gigabit Ethernet interface 6/1:

```
Switch# show policy-map interface gigabitEthernet 6/1
```

```
GigabitEthernet6/1 vlan 20
```

```
Service-policy input: p1
```

```

Class-map: c1 (match-all)
  0 packets
  Match: cos 1
  Match: access-group 100
  police:
    cir 100000000 bps, bc 3125000 bytes
    conformed 0 bytes; actions:
      transmit
    exceeded 0 bytes; actions:
      drop
    conformed 0000 bps, exceed 0000 bps

```

```

Class-map: class-default (match-any)
  0 packets
  Match: any

```

```
GigabitEthernet6/1 vlan 300
```

```
Service-policy output: p2
```

```

Class-map: c1 (match-all)
  0 packets
  Match: cos 1
  Match: access-group 100
  QoS Set
    dscp 50
  police:
    cir 200000000 bps, bc 6250000 bytes
    conformed 0 bytes; actions:
      transmit
    exceeded 0 bytes; actions:
      drop
    conformed 0000 bps, exceed 0000 bps

```

```

Class-map: class-default (match-any)
  0 packets
  Match: any

```

```
GigabitEthernet6/1 vlan 301
```

```
Service-policy output: p2
```

```

Class-map: c1 (match-all)
  0 packets
  Match: cos 1
  Match: access-group 100
  QoS Set
    dscp 50
  police:
    cir 200000000 bps, bc 6250000 bytes
    conformed 0 bytes; actions:
      transmit
    exceeded 0 bytes; actions:
      drop
    conformed 0000 bps, exceed 0000 bps

```

Policy Associations

The supervisor engine supports per-port, per-VLAN policies. The associated policies are attached to the interface, VLAN, and a specific VLAN on a given port, respectively.

A policy can be associated with a variety of objects. The following table lists the objects and the actions allowed.

Table 44-3 Table QoS Policy Associations

Object	Action
Physical port	Policing, marking, and queuing
VLAN	Policing and marking
Port and VLAN (PV)	Policing and marking
EtherChannel	Policing and marking
EtherChannel member port	Queuing

Qos Action Restrictions

- The same actions cannot be performed multiple times in a given direction on different targets. In other words, it is not possible to police the packets both on port and VLAN in the input direction. However, the user can police on the input port and on the output VLAN.
- Queuing actions are only allowed in the egress direction and only on the physical port.
- Percentage-based actions like policer cannot be configured on a VLAN, Port and VLAN (PV) and EtherChannel.
- Port channel or VLAN configuration can only have a policing or a marking action, not a queueing action.

Qos Policy priorities

- If a policy on a port and a VLAN are configured with conflicting actions (such as policing or marking actions on both a port and VLAN), the port policy is picked.
- If policy on a VLAN on a given port must be over-written, the user can configure PV policy.

Qos Policy merging

Applicable policies are applied to a given packet in given direction. For example, if you configure egress VLAN-based police and marking, followed by selective queuing on the port, then actions from both policies will be applied for this packet.

The following policy-map configuration restrictions are imposed on an EtherChannel:

- only policing and marking actions are supported at the EtherChannel level
- only queuing actions are supported at the physical member port level

A packet can be marked (dscp or cos fields) by the EtherChannel policy. If the physical member port policy uses a classification based on dscp or cos fields, it must be based on the marked (modified) value. To ensure proper operation, the following restriction is placed on the EtherChannel.

The classification criteria for the policy-map on the physical member ports has to be based only on one type of field:

- dscp
- precedence
- cos
- any non marking field (no dscp or cos based classification)

Classification criteria for the policy-map on the physical member ports cannot be based on a combination of fields. This restriction ensures that if the EtherChannel policy is marking down dscp or cos, the marked (modified) value-based classification can be implemented in hardware.



Note

Classification criteria for the policy-map on the physical member ports cannot be modified to add a new type of field.

Auto-QoS is not supported on EtherChannel or its member ports. A physical port configured with Auto-QoS is not allowed to become a member of a physical port.

Software QoS

At the highest level, there are two types of locally sourced traffic (such as control protocol packets, pings, and telnets) from the switch: high priority traffic (typically the control protocol packets like OSPF Hellos and STP) and low priority packets (all other packet types).

The QoS treatment for locally-sourced packets differs for the two types.

The supervisor engine provides a way to apply QoS to packets processed in the software path. The packets that get this QoS treatment in software can be classified into two types: software switched packets and software generated packets.

On reception, software switched packets are sent to the CPU that in turn sends them out of another interface. For such packets, input software QoS provides input marking and output software QoS provides output marking and queue selection.

The software generated packets are the ones locally sourced by the switch. The type of output software QoS processing applied to these packets is the same as the one applied to software switched packets. The only difference in the two is that the software switched packets take input marking of the packet into account for output classification purpose.

High Priority Packets

High priority packets are marked as one of the following:

- internally with PAK_PRIORITY
- with IP Precedence of 6 (for IP packets)
- with CoS of 6 (for VLAN Tagged packets)

These packets behave as follows:

- They are not dropped because of any policing, AQM, drop thresholds (or any feature that can drop a packet) configured as per the egress service policy. However, they might be dropped because of hardware resource constraints (packet buffers, queue full, etc.).
- They are classified and marked as per the marking configuration of the egress service policy that could be a port or VLAN (refer to the [“Policy Associations” section on page 44-74](#)).
- These high priority packets are enqueued to queue on the egress port based on the following criteria:
 - If there is no egress queuing policy on the port, the packet is queued to a control packet queue that is setup separately from the default queue and has 5 percent of the link bandwidth reserved for it.
 - If there is an egress queuing policy on the port, the queue is selected based on the classification criteria applicable to the packet.

Low Priority Packets

Packets that are not considered high priority (as described previously) are considered *unimportant*. These include locally sourced pings, telnet, and other protocol packets. They undergo the same treatment as any other packet that is transiting the given transmit port including egress classification, marking and queuing.

Applying Flow-based QoS Policy

Flow based QoS enables microflow policing and marking capability to dynamically learn traffic flows. It also rate limits each unique flow to an individual rate. Flow based QoS is available with the built-in NetFlow hardware support.

For more overview information, refer to the [“Flow-based QoS” section on page 44-10](#).

The following steps show how to apply Flow based QoS policy to QoS targets:

-
- | | |
|---------------|--|
| Step 1 | Create a FNF flow record by specifying the key fields that identify unique flows. You can use any FNF flow records that are associated with the FNF monitor. |
| Step 2 | Create a class-map to specify the set of match criteria. Include the FNF flow record from Step 1 in the class-map match criteria using the match flow record command. Then, configure the class-map to match all the match criteria with class-map match-all <i>class_name</i> . |
| Step 3 | Create a policy-map and define actions associated with class-map from Step 2. |
| Step 4 | Attach the policy to one or more QoS targets. |
-

Examples

The following examples illustrate how to configure Flow based QoS policy and apply microflow policers on individual flows.

Example 1

This example assumes there are multiple users (identified by source IP address) on the subnet 192.168.10.*. The configuration below shows how to configure a flow based QoS policy that uses micro policing to limit the per-user traffic with the source address in the range of 192.168.10.*. The microflow policer is configured with a CIR of 1Mbps, “conform action” as transmit, and “exceed action” as drop.

Step 1: Define an ACL to match traffic with specified source address.

```
Switch(config)# ip access-list extended UserGroup1
Switch(config-ext-nacl)# permit ip 192.168.10.0 0.0.0.255 any
Switch(config-ext-nacl)# exit
Switch(config)#
```

Step 2: Define a flow record to create flows with source address as key.

```
Switch(config)# flow record r1
Switch(config-flow-record)# match ipv4 source address
Switch(config-flow-record)# exit
Switch(config)#
```

Step 3: Configure classmap to match on the UserGroup1 and specify flow record definition for flow creation.

```
Switch(config)# class-map match-all c1
Switch(config-cmap)# match access-group name UserGroup1
Switch(config-cmap)# match flow record r1
Switch(config-cmap)# exit
Switch(config)#
```

Step 4: Configure flow based QoS policy-map with microflow policing action for the matching traffic.

```
Switch(config)# policy-map p1
Switch(config-pmap)# class c1
Switch(config-pmap-c)# police cir 1m
Switch(config-pmap-c-police)# conform-action transmit
Switch(config-pmap-c-police)# exceed-action drop
Switch(config-pmap-c-police)# exit
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
```

Step 5: Attach flow QoS policy to the interface.

```
Switch(config)# interface gigabitEthernet3/1
Switch(config-if)# service-policy input p1
Switch(config-if)#
```

Use the **show** commands (described in the policy and marking sections of this chapter) to display the policy-map configuration and interface specific policy-map statistics.

Example 2.

This example assumes there are multiple users (identified by source IP address) on subnets 192.168.10.* and 172.20.55.*. The first requirement is to police with a CIR of 500Kbps and a PIR of 650Kbps on any TCP traffic originating from 192 network to any destination at any given time. The **exceed action** keyword marks down the dscp value to 32. The second requirement is to police per-user traffic originating from 172 network to CIR of 2Mbps and unconditionally mark the traffic with dscp 10.

Step 1: Define an ACL to match traffic with specified source address.

```
Switch(config)# ip access-list extended UserGroup1
Switch(config-ext-nacl)# permit ip 19 2.168.10.0 0.0.0.255 any
Switch(config-ext-nacl)# exit
Switch(config)# ip access-list extended UserGroup2
Switch(config-ext-nacl)# permit ip 172.20.55.0 0.0.0.255 any
Switch(config-ext-nacl)# exit
Switch(config)#
```

Step 2: Define a flow record to create flows with source address as key.

```
Switch(config)# flow record r1
Switch(config-flow-record)# match ipv4 source address
Switch(config-flow-record)# match ipv4 destination address
Switch(config-flow-record)# match transport tcp source-port
Switch(config-flow-record)# match transport tcp destination-port
Switch(config-flow-record)# exit
Switch(config)# flow record r2
Switch(config-flow-record)# match ipv4 source address
Switch(config-flow-record)# exit
Switch(config)#
```

Step 3: Configure classmap to match on the UserGroup1 and specify flow record definition for flow creation.

```
Switch(config)# class-map match-all c1
Switch(config-cmap)# match access-group name UserGroup1
Switch(config-cmap)# match flow record r1
Switch(config-cmap)# exit
Switch(config)# class-map match-all c2
Switch(config-cmap)# match access-group name UserGroup2
Switch(config-cmap)# match flow record r2
Switch(config-cmap)# exit
Switch(config)#
```

Step 4: Configure flow based QoS policy-map with microflow policing action for the matching traffic.

```
Switch(config)# policy-map p1
Switch(config-pmap)# class c1
Switch(config-pmap-c)# police cir 500k pir 650k
Switch(config-pmap-c-police)# conform-action transmit
Switch(config-pmap-c-police)# exceed-action set-dscp-transmit 32
Switch(config-pmap-c-police)# violate-action drop
Switch(config-pmap-c-police)# exit
Switch(config-pmap-c)# exit
Switch(config-pmap)# class c2
Switch(config-pmap-c)# set dscp 10
Switch(config-pmap-c)# police cir 2m
Switch(config-pmap-c-police)# conform-action transmit
Switch(config-pmap-c-police)# exceed-action drop
Switch(config-pmap-c-police)# exit
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
```

Step 5: Attach flow QoS policy to the interface.


```
Switch(config)# interface gigabitEthernet3/1
Switch(config-if)# service-policy input p1
Switch(config-if)# exit
```

Use the show commands described in the QoS section to display the policy-map configuration and interface specific policy-map statistics.

Example 3

Assume that there are two active flows on FastEthernet interface 6/1:

Table 44-4

SrcIp	DStIp	IPProt	SrcL4Port	DstL4Port
192.168.10.10	192.168.20.20	20	6789	81
192.168.10.10	192.168.20.20	20	6789	21

With the following configuration, each flow is policed to 1000000 bps with an allowed 9000 burst value.

```
Switch(config)# flow record r1
Switch(config-flow-record)# match ipv4 source address
Switch(config-flow-record)# match ipv4 destination address
Switch(config-flow-record)# match transport tcp source-port
Switch(config-flow-record)# match transport tcp destination-port
Switch(config-flow-record)# match transport udp source-port
Switch(config-flow-record)# match transport udp destination-port
Switch(config-flow-record)# exit
Switch(config)# class-map match-all c1
Switch(config-cmap)# match flow record r1
Switch(config-cmap)# exit
Switch(config)# policy-map p1
Switch(config-pmap)# class c1
Switch(config-pmap-c)# police 1000000 9000
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface fastEthernet 6/1
Switch(config-if)# service-policy input p1
Switch(config-if)# end
```

Configuration Guidelines

The general guidelines for creating, configuring, modifying, deleting a flow based QoS policy and attaching (and detaching) a flow based QoS policy to a supported target is the same as described in the QoS section. The following description and restriction applies to Flow based QoS policy:

- A classmap can have multiple match statements but only one FNF flow record can be specified in a class-map.
- A flow record must have at least one key field before it can be used in a classmap. Non-key fields can be present in the flow record. However, all the non-key fields are ignored by microflow QoS. Only key-fields are used for flow creation.
- If a FNF flow record is referenced in any class-map, the flow record cannot be modified. Remove the flow record from all classmaps before modifying it.
- A classmap with a FNF flow record must be configured as **match-all**; traffic hitting the class-map must satisfy all match criteria in the class-map.

- A policy can contain multiple classes and each class-map may contain the same or different FNF flow record.
- Flow based QoS policy and FNF monitor both cannot be applied on the same target at the same time.
- When the interface mode changes from switchport to routed port and vice versa, any Flow QoS policy attached to the port remains applied after the mode change.
- There are 3 types of FNF flow records: ipv4, ipv6, and datalink. The datalink flow record is mutually exclusive with the ipv4 and ipv6 flow records; a classmap with the datalink flow record cannot co-exist with classmap having a ipv4 or ipv6 flow record in the same policy and vice-versa.
- Classmap class-default is not editable; it cannot be configured with the match flow record. Instead, you can configure the policy with a class-map that uses a match any filter and the flow record.
- Traffic is classified in the same order in which class-map is defined in a policy. Hence, if a FNF flow record is the only match statement in a class-map, the classifier matches all packets of the type identified by the flow record. This means that any subsequent class-map in the same policy matching on the same traffic type will be redundant and will never be hit.
- Policers associated with classmap having flow record are called *microflow policers*. The CIR and PIR rates for microflow policers cannot be configured using the percent keyword.
- Flow records within the same policy must include appropriate key fields to ensure flows created from different classmaps are unique and distinct. Otherwise, the resulting flows from different classmap cannot be distinguished. In such cases, policy actions corresponding to the classmap which created the first flow in hardware will apply and results will not be always be as expected.
- Flows from traffic received on different QoS targets are distinct even if the same policy is applied to those targets.
- A flow is aged out if the it is inactive for more than 5 seconds; there is no traffic matching the flow for a period longer than 5 sec.
- When a flow is aged out, policer state information associated with the flow is also deleted. When a new flow is created, the policer instance for the flow is re-initialized.
- Flows created by flow based QoS policy exist in hardware only and cannot be exported (as with FNF monitor).
- Per-flow statistics are not available for flows created by flow based QoS policy.
- Class-map statistics indicate the number of packets matching the classifier. It does not represent individual flow stats.
- Policer statistics show the aggregate policer statistics of individual flow.
- Information about the flows created by hardware are not available and not displayed in the show commands associated with QoS policy-map. Only class-map and policer statistics are displayed in the output of the **show policy-map** commands.

Configuring CoS Mutation

CoS reflection and CoS mutation are supported on Supervisor Engine 6-E. Below is an example of how to apply CoS reflection.

Let us say that traffic arrives on interface gigabit 2/5 with VLAN 10 and COS 1, 2, We want traffic to egress interface gigabit 2/6 with outer tag VLAN 11 and CoS copied from C-tag, where C-tag is VLAN 10 and COS 1, 2, ...

```
class-map match-all c2
```

```
match cos 2

class-map match-all c1

match cos 1

!

policy-map my

class c1

set cos 1

class c2

set cos 2

interface GigabitEthernet2/5

switchport mode trunk

switchport vlan mapping 10 dot1q-tunnel 11

spanning-tree bpdupfilter enable

service-policy input my

!

interface GigabitEthernet2/6

switchport mode trunk
```

Configuring System Queue Limit



Note This feature is available only from Cisco IOS Release 15.0(2)SG1 and later and Cisco IOS Release XE 3.2.1SG.

With the **hw-module system max-queue-limit** command, the Catalyst 4500 series switch allows you to change the queue limit for all interfaces globally, instead of applying a policy with queue limit to all the interfaces.

To set the queue limit globally, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# hw-module system max-queue-limit <i>max-queue-limit</i>	Sets the queue limit for all interfaces globally. Valid values are from 1024 to 8184. The value must be a multiple of 8.

	Command	Purpose
Step 3	Switch(config)# exit	Returns to privileged EXEC mode.
Step 4	Switch# reload	Reloads standalone supervisor engine.
	or	
	Switch# redundancy reload shelf	Reloads redundancy supervisor engine in SSO mode.
	Switch# redundancy force-switchover	Reloads redundancy supervisor engine in RPR mode. This command must be followed by another redundancy force-switchover.

This is a global configuration command. You can override it with the per port, per class, **queue-limit** command.

For a standalone supervisor engine, you must reboot the engine after applying this command.

For redundant supervisors in SSO mode, you must enter the **redundancy reload shelf** command enforce reboot to both the supervisors. For redundancy supervisors in RPR mode, you must execute two consecutive switchovers to enforce the system queue limit on both the supervisors.

This example shows how to set the queue limit globally to 1024 on a standalone supervisor engine:

```
Switch> enable
Switch# configure terminal
Switch(config)# hw-module system max-queue-limit 1024
Switch(config)# exit
Switch# reload (for standalone supervisors)
Switch# redundancy reload shelf (for redundancy supervisors in SSO mode)
or
Switch# redundancy force-switchover (followed by another redundancy force-switchover, for
redundancy supervisors in RPR mode)
```

Configuring VSS Auto-QoS

- [Auto-QoS Overview, page 44-82](#)
- [Auto-QoS Policy and Class Maps, page 44-83](#)
- [Auto-Qos Compact, page 44-88](#)
- [Effects of Auto-QoS and Auto-Qos Compact on Running Configuration, page 44-89](#)

Auto-QoS Overview



Note

Auto-QoS cannot be applied to VLANs or EtherChannel interfaces.



Note

If you have an auto-QoS policy on a port connected to a device that supports CDP, the port is automatically trusted. However, if the device does not support CDP (like legacy Digital Media Player), QoS trust must be applied manually.

You can use the auto-QoS feature to simplify the deployment of QoS features. Auto-QoS determines the network design and enables QoS configurations so that the switch can prioritize different traffic flows. You can also hide all the auto-QoS-generated configuration from the running configuration, by using the auto-QoS compact feature.

The switch employs the MQC model. This means that instead of using certain global configurations (like qos and qos dbf), auto-QoS applied to any interface on a switch configures several global class-maps and policy-maps.

We need QoS in both directions, both on inbound and outbound. Inbound, the switch port needs to trust the DSCP in the packet (done by default). Outbound, the switch port needs to give voice packets "front of line" priority. If voice is delayed too long by waiting behind other packets in the outbound queue, the end host drops the packet because it arrives outside of the receive window for that packet.



Note QoS is a two way street. So, it might work in one direction and not in the other.

Auto-QoS Policy and Class Maps

There are 7 policy maps that must be defined (5 Input, 2 output) on all ports

- AutoQos-4.0-Input-Policy
- AutoQos-VoIP-Input-Cos-Policy
- AutoQos-VoIP-Input-Dscp-Policy
- AutoQos-4.0-Cisco-Phone-Input-Policy
- AutoQos-4.0-Output-Policy
- AutoQos-4.0-Cisco-Softphone-Input-Policy
- AutoQos-VoIP-Output-Policy

The problem with COS is that packets on the native VLAN is marked as zero.

The class maps used for input matching are as follows:

```
! for control traffic between the phone and the callmanager
! and phone to phone [Bearer] DSCP matching

class-map match-all AutoQos-VoIP-Control-Dscp26
  match dscp af31
class-map match-all AutoQos-VoIP-Control-Dscp24
  match dscp cs3
class-map match-all AutoQos-VoIP-Bearer-Dscp
  match dscp ef

! for control traffic and phone to phone [Bearer] COS matching
! Note: Both CS3 and AF31 control traffic maps to COS 3

class-map match-all AutoQos-VoIP-Control-Cos
  match cos 3
class-map match-all AutoQos-VoIP-Bearer-Cos
  match cos 5

! for control traffic between the softphonephone and the callmanager
! and softphone to softphonephone [Bearer] DSCP matching
Class Map match-all AutoQos-4.0-Multimedia-Conf-Classify (id 36)
  Match access-group name AutoQos-4.0-ACL-Multimedia-Conf
Class Map match-all AutoQos-4.0-Signaling-Classify (id 2)
```

```

    Match access-group name AutoQos-4.0-ACL-Signaling
Class Map match-all AutoQos-4.0-Transaction-Classify (id 18)
    Match access-group name AutoQos-4.0-ACL-Transactional-Data
Class Map match-all AutoQos-4.0-Bulk-Data-Classify (id 29)
    Match access-group name AutoQos-4.0-ACL-Bulk-Data
Class Map match-all AutoQos-4.0-Scavenger-Classify (id 1)
    Match access-group name AutoQos-4.0-ACL-Scavenger

! for untrusted interfaces
class-map match-all AutoQos-4.0-Multimedia-Conf-Classify
    match access-group name AutoQos-4.0-ACL-Multimedia-Conf
class-map match-all AutoQos-4.0-Signaling-Classify
    match access-group name AutoQos-4.0-ACL-Signaling
class-map match-all AutoQos-4.0-Transaction-Classify
    match access-group name AutoQos-4.0-ACL-Transactional-Data
class-map match-all AutoQos-4.0-Bulk-Data-Classify
    match access-group name AutoQos-4.0-ACL-Bulk-Data
class-map match-all AutoQos-4.0-Scavenger-Classify
    match access-group name AutoQos-4.0-ACL-Scavenger
    class-map match-all AutoQos-4.0-Default-Classify
    match access-group name AutoQos-4.0-ACL-Default

! for interfaces with video devices
class-map match-any AutoQos-4.0-VoIP
    match dscp ef
    match cos 5
class-map match-all AutoQos-4.0-Broadcast-Vid
    match dscp cs5
class-map match-all AutoQos-4.0-Realtime-Interact
    match dscp cs4
class-map match-all AutoQos-4.0-Network-Ctrl
    match dscp cs7
class-map match-all AutoQos-4.0-Internetwork-Ctrl
    match dscp cs6
class-map match-any AutoQos-4.0-Signaling
    match dscp cs3
    match cos 3
class-map match-all AutoQos-4.0-Network-Mgmt
    match dscp cs2
class-map match-any AutoQos-4.0-Multimedia-Conf
    match dscp af41
    match dscp af42
    match dscp af43
class-map match-any AutoQos-4.0-Multimedia-Stream
    match dscp af31
    match dscp af32
    match dscp af33
class-map match-any AutoQos-4.0-Transaction-Data
    match dscp af21
    match dscp af22
    match dscp af23
class-map match-any AutoQos-4.0-Bulk-Data
    match dscp af11
    match dscp af12
    match dscp af13
class-map match-all AutoQos-4.0-Scavenger
    match dscp cs1

```

The class maps are intended to identify control and data (bearer) voice traffic for either an Layer 2 or Layer 3 interface.

The 2 Input policy maps, one for matching DSCP and one for matching COS, where DSCP and COS are set to an assigned qos-group used in outbound policy-maps are as follows:

```
policy-map AutoQos-VoIP-Input-Dscp-Policy
  class AutoQos-VoIP-Bearer-Dscp
  class AutoQos-VoIP-Control-Dscp26
  class AutoQos-VoIP-Control-Dscp24
```

! Note: For COS, Control traffic only has a single COS value of 3 (versus DSCP which has 2 values to match). So, only 2 class-maps instead of 3 like above.

```
policy-map AutoQos-VoIP-Input-Cos-Policy
  class AutoQos-VoIP-Bearer-Cos
  class AutoQos-VoIP-Control-Cos
```

The input policy maps are as follows:

```
policy-map AutoQos-4.0-Input-Policy
  class AutoQos-4.0-VoIP
  class AutoQos-4.0-Broadcast-Vid
  class AutoQos-4.0-Realtime-Interact
  class AutoQos-4.0-Network-Ctrl
  class AutoQos-4.0-Internetwork-Ctrl
  class AutoQos-4.0-Signaling
  class AutoQos-4.0-Network-Mgmt
  class AutoQos-4.0-Multimedia-Conf
  class AutoQos-4.0-Multimedia-Stream
  class AutoQos-4.0-Transaction-Data
  class AutoQos-4.0-Bulk-Data
  class AutoQos-4.0-Scavenger
  policy-map AutoQos-4.0-Classify-Police-Input-Policy
  class AutoQos-4.0-Multimedia-Conf-Classify
    set dscp af41
    set cos 4
    police cir 5000000 bc 8000
    exceed-action drop
  class AutoQos-4.0-Signaling-Classify
    set dscp cs3
    set cos 3
    police cir 32000 bc 8000
    exceed-action drop
  class AutoQos-4.0-Transaction-Classify
    set dscp af21
    set cos 2
    police cir 10000000 bc 8000
    exceed-action set-dscp-transmit cs1
    exceed-action set-cos-transmit 1
  class AutoQos-4.0-Bulk-Data-Classify
    set dscp af11
    set cos 1
    police cir 10000000 bc 8000
    exceed-action set-dscp-transmit cs1
    exceed-action set-cos-transmit 1
  class AutoQos-4.0-Scavenger-Classify
    set dscp cs1
    set cos 1
    police cir 10000000 bc 8000
    exceed-action drop
  class AutoQos-4.0-Default-Classify
    set dscp default
    set cos 0
```

```

    police cir 10000000 bc 8000
        exceed-action set-dscp-transmit cs1
        exceed-action set-cos-transmit 1
policy-map AutoQos-4.0-Cisco-Phone-Input-Policy
class AutoQos-4.0-VoIP-Data-Cos
    set dscp ef
    police cir 128000 bc 8000
        exceed-action set-dscp-transmit cs1
        exceed-action set-cos-transmit 1
class AutoQos-4.0-VoIP-Signal-Cos
    set dscp cs3
    police cir 32000 bc 8000
        exceed-action set-dscp-transmit cs1
        exceed-action set-cos-transmit 1
class class-default
    set dscp default
    set cos 0
policy-map AutoQos-4.0-Classify-Input-Policy
class AutoQos-4.0-Multimedia-Conf-Classify
    set dscp af41
    set cos 4
class AutoQos-4.0-Signaling-Classify
    set dscp cs3
    set cos 3
class AutoQos-4.0-Transaction-Classify
    set dscp af21
    set cos 2
class AutoQos-4.0-Bulk-Data-Classify
    set dscp af11
    set cos 1
class AutoQos-4.0-Scavenger-Classify
    set dscp cs1
    set cos 1
class AutoQos-4.0-Default-Classify
    set dscp default
    set cos 0
!
```

The output policy maps are as follows:

```

policy-map AutoQos-4.0-Output-Policy
class AutoQos-4.0-Scavenger-Queue
    bandwidth remaining percent 1
class AutoQos-4.0-Priority-Queue
    priority
    police cir percent 30 bc 33 ms
class AutoQos-4.0-Control-Mgmt-Queue
    bandwidth remaining percent 10
class AutoQos-4.0-Multimedia-Conf-Queue
    bandwidth remaining percent 10
class AutoQos-4.0-Multimedia-Stream-Queue
    bandwidth remaining percent 10
class AutoQos-4.0-Trans-Data-Queue
    bandwidth remaining percent 10
    dbl
class AutoQos-4.0-Bulk-Data-Queue
    bandwidth remaining percent 4
    dbl
class class-default
    bandwidth remaining percent 25
    dbl
```


The three policy maps are defined as follows:

- **policy-map AutoQos-VoIP-Input-Dscp-Policy**
This policy map is applied as an input service policy on an Layer 3 interface (such as an uplink connection to a neighboring switch) when auto-QoS is configured on the port.
- **policy-map AutoQos-VoIP-Input-Cos-Policy**
This policy map is applied as an input service policy on an Layer 2 interface that could be either an uplink connection or a port hooked to a Cisco IP Phone.
- **policy-map AutoQos-VoIP-Output-Policy**
This policy map is applied as an output policy for any port on which auto-QoS is configured, establishing policy governing egress traffic on the port based on whether it is voice data or control traffic.

The purpose of the input policy maps is to identify voice data or control traffic and mark it as such as it traverses the switch. The output policy map matches the packets on the marking occurring on ingress and then applies egress parameters such as bandwidth, policing and/or priority queuing.

For switch-to-switch connections, the **[no] auto qos voip trust** command is used to apply an input and output service policy on the interface:

```
service-policy input AutoQos-VoIP-Input-Cos-Policy
```

OR

```
service-policy input AutoQos-VoIP-Input-Dscp-Policy
```

AND

```
service-policy output AutoQos-VoIP-Output-Policy
```

The selection of the input policy depends on whether the port is Layer 2 or Layer 3. For Layer 2, the policy trusts the Cos setting in the received packets. For Layer 3 ports, it relies on the DSCP value contained in the packets.

For phone connected ports, the **[no] auto qos voice cisco-phone** command is used to apply the following service policy to the port:

```
qos trust device cisco-phone
```

```
service-policy input AutoQos-VoIP-Input-Cos-Policy
```

AND

```
service-policy output AutoQos-VoIP-Output-Policy
```

It establishes a trusted boundary that recognizes Cisco IP Phones and trusts the Cos setting of the packets from the phone. If a Cisco IP Phone is not detected, the Cos field is ignored and the packets are not classified as voice traffic. Upon detecting a Cisco phone, the ingress packets are marked based on the Cos value in the packets. This marking is used on egress for proper traffic classification and handling.

Auto qos srnd4—Is generated when any new auto qos command is configured on an interface and migrates from legacy CLIs to generate new configurations. This CLI only generates a global configuration if during migration, one or more interfaces has legacy auto-QoS enabled

Auto qos video—Generates QoS configuration for untrusted interfaces. It incorporates a service-policy to classify the traffic coming from untrusted desktops/devices and marks them accordingly.

Auto qos void cisco-softphone—Generate QoS configuration for interfaces connected to PCs running the Cisco IP SoftPhone application and marks as police traffic stemming from such interfaces. Ports configured with this CLI are considered untrusted.

Auto qos classify—Generates QoS configuration for untrusted interfaces. It applies a service-policy to classify the traffic stemming from untrusted desktops or devices and marks them accordingly. The service-policies generated do not police.

Auto-Qos Compact

When you enter an auto-QoS command, the switch goes on to display all the generated commands as if the commands were entered from the CLI. Enable auto-QoS compact if you want to hide auto-QoS generated commands from the running configuration.

To enable auto-Qos compact, enter this command:

Command	Purpose
Switch# configure terminal	Enters global configuration mode.
Switch(config)# auto qos global compact	<p>Enables auto-Qos compact and generates global auto-Qos configurations that are hidden from the running configuration.</p> <p>You can then enter the auto-QoS command you want to configure, in the interface configuration mode. The interface commands that the system generates are also hidden from running configuration.</p> <p>To display the auto-QoS configuration that has been applied, use these the privileged EXEC commands: show derived-config, show policy-map, show policy-map interface, show class-map, show table-map, and show auto-qos, show ip access-list.</p>

To disable auto-QoS compact, first remove all the auto-Qos configuration by entering the **no** form of the corresponding auto-QoS interface configuration command and then enter the **no auto qos global compact** global configuration command. Disabling auto-QoS compact is a required task when have to perform an ISSU from an image where auto-QoS compact is supported to an image where this feature is not available.

The following example shows you how to enable auto-qos compact and configure the **auto qos voip cisco-phone** interface configuration command and then display configuration details:

```
Switch# configure terminal
Switch(config)# auto qos global compact
Switch(config)# interface GigabitEthernet1/2
Switch(config-if)# auto qos voip cisco-phone

Switch(config-if)# show auto-qos
GigabitEthernet1/2
auto qos voip cisco-phone
```

Effects of Auto-QoS and Auto-Qos Compact on Running Configuration

The auto-QoS interface configuration commands and the generated global configurations are added to the running configuration. When you save this configuration, all generated commands and any user-entered configuration that was not overridden, is saved to memory.

If auto-QoS compact is enabled, only the list of auto-QoS commands you have entered are displayed in the running configuration. The generated global and interface configurations are hidden from the running configuration. When you save this configuration, only the auto-QoS commands you have entered are saved (and not the hidden configuration). When you reload the switch, the system detects and re-executes the saved auto-QoS commands, re-sets the values, and generates an auto-QoS SRND4.0 complaint configuration set.

**Note**

Do not make changes to the auto-QoS-generated commands when auto-QoS compact is enabled, because user-modifications are overridden when the switch reloads.

Configuring Auto-QoS on a Standalone Supervisor Engine 6-E/6L-E or Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, and 7-E

- [Auto-QoS Overview, page 44-89](#)
- [Auto-QoS Policy and Class Maps, page 44-90](#)
- [Auto-QoS Compact, page 44-97](#)
- [Effects of Auto-QoS and Auto-Qos Compact on Running Configuration, page 44-98](#)

Auto-QoS Overview

**Note**

Auto-QoS cannot be applied to VLANs or EtherChannel interfaces.

**Note**

If you have an auto-QoS policy on a port connected to a device that supports CDP, the port is automatically trusted. However, if the device does not support CDP (like legacy Digital Media Player), QoS trust must be applied manually.

You can use the auto-QoS feature to simplify the deployment of QoS features. Auto-QoS determines the network design and enables QoS configurations so that the switch can prioritize different traffic flows. You can also hide all the auto-QoS-generated configuration from the running configuration, by using the auto-QoS compact feature.

The switch employs the MQC model. This means that instead of using certain global configurations (like qos and qos dbl), auto-QoS applied to any interface on a switch configures several global class-maps and policy-maps.

Auto-QoS matches traffic and assigns each matched packet to qos-groups. This allows the output policy map to put specific qos-groups into specific queues, including into the priority queue.

We need QoS in both directions, both on inbound and outbound. Inbound, the switch port needs to trust the DSCP in the packet (done by default). Outbound, the switch port needs to give voice packets "front of line" priority. If voice is delayed too long by waiting behind other packets in the outbound queue, the end host drops the packet because it arrives outside of the receive window for that packet.



Note QoS is a two way street. So, it might work in one direction and not in the other.

Auto-QoS Policy and Class Maps

- There are 7 policy maps that must be defined (5 Input, 2 output)
- AutoQos-4.0-Input-Policy
- AutoQos-VoIP-Input-Cos-Policy
- AutoQos-VoIP-Input-Dscp-Policy
- AutoQos-4.0-Cisco-Phone-Input-Policy
- AutoQos-4.0-Output-Policy
- AutoQos-4.0-Cisco-Softphone-Input-Policy
- AutoQos-VoIP-Output-Policy

On all ports. The problem with COS is that packets on the native VLAN is marked as zero.

The class maps used for input matching are as follows:

```
! for control traffic between the phone and the callmanager
! and phone to phone [Bearer] DSCP matching
! Note: Control traffic can be either AF31 or CS3. So, we match to both values and assign
them to different qos-groups when matching DSCP and only a single group when matching COS.
```

```
class-map match-all AutoQos-VoIP-Control-Dscp26
  match dscp af31
class-map match-all AutoQos-VoIP-Control-Dscp24
  match dscp cs3
class-map match-all AutoQos-VoIP-Bearer-Dscp
  match dscp ef
```

```
! for control traffic and phone to phone [Bearer] COS matching
! Note: Both CS3 and AF31 control traffic maps to COS 3
```

```
class-map match-all AutoQos-VoIP-Control-Cos
  match cos 3
class-map match-all AutoQos-VoIP-Bearer-Cos
  match cos 5
```

```
! for control traffic between the softphonephone and the callmanager
! and softphone to softphonephone [Bearer] DSCP matching
Class Map match-all AutoQos-4.0-Multimedia-Conf-Classify (id 36)
  Match access-group name AutoQos-4.0-ACL-Multimedia-Conf
Class Map match-all AutoQos-4.0-Signaling-Classify (id 2)
  Match access-group name AutoQos-4.0-ACL-Signaling
Class Map match-all AutoQos-4.0-Transaction-Classify (id 18)
  Match access-group name AutoQos-4.0-ACL-Transactional-Data
Class Map match-all AutoQos-4.0-Bulk-Data-Classify (id 29)
  Match access-group name AutoQos-4.0-ACL-Bulk-Data
Class Map match-all AutoQos-4.0-Scavenger-Classify (id 1)
  Match access-group name AutoQos-4.0-ACL-Scavenger
```

```

! for untrusted interfaces
class-map match-all AutoQos-4.0-Multimedia-Conf-Classify
    match access-group name AutoQos-4.0-ACL-Multimedia-Conf
class-map match-all AutoQos-4.0-Signaling-Classify
    match access-group name AutoQos-4.0-ACL-Signaling
class-map match-all AutoQos-4.0-Transaction-Classify
    match access-group name AutoQos-4.0-ACL-Transactional-Data
class-map match-all AutoQos-4.0-Bulk-Data-Classify
    match access-group name AutoQos-4.0-ACL-Bulk-Data
class-map match-all AutoQos-4.0-Scavenger-Classify
    match access-group name AutoQos-4.0-ACL-Scavenger
    class-map match-all AutoQos-4.0-Default-Classify
    match access-group name AutoQos-4.0-ACL-Default

! for interfaces with video devices
class-map match-any AutoQos-4.0-VoIP
    match dscp ef
    match cos 5
class-map match-all AutoQos-4.0-Broadcast-Vid
    match dscp cs5
class-map match-all AutoQos-4.0-Realtime-Interact
    match dscp cs4
class-map match-all AutoQos-4.0-Network-Ctrl
    match dscp cs7
class-map match-all AutoQos-4.0-Internetwork-Ctrl
    match dscp cs6
class-map match-any AutoQos-4.0-Signaling
    match dscp cs3
    match cos 3
class-map match-all AutoQos-4.0-Network-Mgmt
    match dscp cs2
class-map match-any AutoQos-4.0-Multimedia-Conf
    match dscp af41
    match dscp af42
    match dscp af43
class-map match-any AutoQos-4.0-Multimedia-Stream
    match dscp af31
    match dscp af32
    match dscp af33
class-map match-any AutoQos-4.0-Transaction-Data
    match dscp af21
    match dscp af22
    match dscp af23
class-map match-any AutoQos-4.0-Bulk-Data
    match dscp af11
    match dscp af12
    match dscp af13
class-map match-all AutoQos-4.0-Scavenger
    match dscp cs1

```

The class maps are intended to identify control and data (bearer) voice traffic for either an Layer 2 or Layer 3 interface.

The 2 Input policy maps, one for matching DSCP and one for matching COS, where DSCP and COS are set to an assigned qos-group used in outbound policy-maps are as follows:

```

policy-map AutoQos-VoIP-Input-Dscp-Policy
    class AutoQos-VoIP-Bearer-Dscp
        set qos-group 46
    class AutoQos-VoIP-Control-Dscp26
        set qos-group 26
    class AutoQos-VoIP-Control-Dscp24
        set qos-group 24

```

! Note: For COS, Control traffic only has a single COS value of 3 (versus DSCP which has 2 values to match). So, only 2 class-maps instead of 3 like above.

```
policy-map AutoQos-VoIP-Input-Cos-Policy
  class AutoQos-VoIP-Bearer-Cos
    set qos-group 46
  class AutoQos-VoIP-Control-Cos
    set qos-group 24
```

```
Policy Map AutoQos-4.0-Input-Policy
  Class AutoQos-4.0-VoIP
    set qos-group 32
  Class AutoQos-4.0-Broadcast-Vid
    set qos-group 32
  Class AutoQos-4.0-Realtime-Interact
    set qos-group 32
  Class AutoQos-4.0-Network-Ctrl
    set qos-group 16
  Class AutoQos-4.0-Internetwork-Ctrl
    set qos-group 16
  Class AutoQos-4.0-Signaling
    set qos-group 16
  Class AutoQos-4.0-Network-Mgmt
    set qos-group 16
  Class AutoQos-4.0-Multimedia-Conf
    set qos-group 34
  Class AutoQos-4.0-Multimedia-Stream
    set qos-group 26
  Class AutoQos-4.0-Transaction-Data
    set qos-group 18
  Class AutoQos-4.0-Bulk-Data
    set qos-group 10
  Class AutoQos-4.0-Scavenger
    set qos-group 8
```

```
Policy Map AutoQos-4.0-Cisco-Phone-Input-Policy
  Class AutoQos-4.0-VoIP-Data-Cos
    set dscp ef
    set qos-group 32
    police cir 128000 bc 8000
      conform-action transmit
      exceed-action set-dscp-transmit cs1
      exceed-action set-cos-transmit 1
  Class AutoQos-4.0-VoIP-Signal-Cos
    set dscp cs3
    set qos-group 16
    police cir 32000 bc 8000
      conform-action transmit
      exceed-action set-dscp-transmit cs1
      exceed-action set-cos-transmit 1
  Class AutoQos-4.0-Default-Classify
    set dscp default
    set cos 0
    police cir 10000000 bc 8000
      conform-action transmit
      exceed-action set-dscp-transmit cs1
      exceed-action set-cos-transmit 1
```

```
Policy Map AutoQos-4.0-Cisco-Softphone-Input-Policy
  Class AutoQos-4.0-VoIP-Data
    set dscp ef
    set cos 5
```

```
    set qos-group 32
  police cir 128000 bc 8000
    conform-action transmit
    exceed-action set-dscp-transmit cs1
    exceed-action set-cos-transmit 1
Class AutoQos-4.0-VoIP-Signal
  set dscp cs3
  set cos 3
  set qos-group 16
  police cir 32000 bc 8000
    conform-action transmit
    exceed-action set-dscp-transmit cs1
    exceed-action set-cos-transmit 1
Class AutoQos-4.0-Multimedia-Conf-Classify
  set dscp af41
  set cos 4
  set qos-group 34
  police cir 5000000 bc 8000
    conform-action transmit
    exceed-action drop
Class AutoQos-4.0-Signaling-Classify
  set dscp cs3
  set cos 3
  set qos-group 16
  police cir 32000 bc 8000
    conform-action transmit
    exceed-action drop
Class AutoQos-4.0-Transaction-Classify
  set dscp af21
  set cos 2
  set qos-group 18
  police cir 10000000 bc 8000
    conform-action transmit
    exceed-action set-dscp-transmit cs1
    exceed-action set-cos-transmit 1
Class AutoQos-4.0-Bulk-Data-Classify
  set dscp af11
  set cos 1
  set qos-group 10
  police cir 10000000 bc 8000
    conform-action transmit
    exceed-action set-dscp-transmit cs1
    exceed-action set-cos-transmit 1
Class AutoQos-4.0-Scavenger-Classify
  set dscp cs1
  set cos 1
  set qos-group 8
  police cir 10000000 bc 8000
    conform-action transmit
    exceed-action drop
Class AutoQos-4.0-Default-Classify
  set dscp default
  set cos 0

Policy Map AutoQos-4.0-Classify-Input-Policy
  Class AutoQos-4.0-Multimedia-Conf-Classify
    set dscp af41
    set cos 4
    set qos-group 34
  Class AutoQos-4.0-Signaling-Classify
    set dscp cs3
    set cos 3
    set qos-group 16
  Class AutoQos-4.0-Transaction-Classify
```

```

        set dscp af21
        set cos 2
        set qos-group 18
Class AutoQos-4.0-Bulk-Data-Classify
        set dscp af11
        set cos 1
        set qos-group 10
Class AutoQos-4.0-Scavenger-Classify
        set dscp cs1
        set cos 1
        set qos-group 8
Class AutoQos-4.0-Default-Classify
        set dscp default
        set cos 0

```

The class maps used for Output matching are as follows:

```

! Since we assigned matched traffic to a qos-group on input,
! we only need to match the qos-group on output

```

```

! Note: Any other traffic not matched on input and assigned to a qos-group goes into the
class-default queue

```

```

! for control traffic (CS3 and AF31)
class-map match-all AutoQos-VoIP-Control-QosGroup24
    match qos-group 24
class-map match-all AutoQos-VoIP-Control-QosGroup26
    match qos-group 26

```

```

! For phone to phone (Bearer EF) traffic
class-map match-all AutoQos-VoIP-Bearer-QosGroup
    match qos-group 46

```

! For softphone

Class Map match-any AutoQos-4.0-Scavenger-Queue (id 24)

```
    Match qos-group 8
```

```
    Match dscp cs1 (8)
```

Class Map match-all AutoQos-4.0-Priority-Queue (id 3)

```
    Match qos-group 32
```

Class Map match-all AutoQos-4.0-Control-Mgmt-Queue (id 28)

```
    Match qos-group 16
```

Class Map match-all AutoQos-4.0-Multimedia-Conf-Queue (id 10)

```
    Match qos-group 34
```

Class Map match-all AutoQos-4.0-Multimedia-Stream-Queue (id 5)

```
    Match qos-group 26
```

Class Map match-all AutoQos-4.0-Trans-Data-Queue (id 30)

```
    Match qos-group 18
```

Class Map match-all AutoQos-4.0-Bulk-Data-Queue (id 17)

```
    Match qos-group 10
```


**Note**

The previous section listing defines the AutoQoS macros for defining QoS guidelines prior to *Solution Reference Network Design 4.0 (SRND4)*. Starting with Cisco Release XE 3.3.0(SG) and 15.1(1)SG, the Catalyst 4500 series switch supports the **auto qos srnd4** command.

The following classes are required and generated by all SRND4 CLIs.

```
class-map match-all AutoQos-4.0-Priority-Queue
  match qos-group 32
class-map match-all AutoQos-4.0-Control-Mgmt-Queue
  match qos-group 16
class-map match-all AutoQos-4.0-Multimedia-Conf-Queue
  match qos-group 34
class-map match-all AutoQos-4.0-Multimedia-Stream-Queue
  match qos-group 26
class-map match-all AutoQos-4.0-Trans-Data-Queue
  match qos-group 18
class-map match-all AutoQos-4.0-Bulk-Data-Queue
  match qos-group 10
class-map match-any AutoQos-4.0-Scavenger-Queue
  match qos-group 8
  match dscp cs1
```

The output policy maps are as follows:

```
! Each class maps to a different qos-group with
! class-default taking any traffic not assigned to a qos-group
```

! Note: in this example, the outbound policy map drops voice packets when the priority queue exceeds 33% utilization of the link. Each deployment must establish their own upper bound for voice packets.

```
policy-map AutoQos-VoIP-Output-Policy
  class AutoQos-VoIP-Bearer-QosGroup
    set dscp ef
    set cos 5
    priority
    police cir percent 33
  class AutoQos-VoIP-Control-QosGroup26
    set dscp af31
    set cos 3
    bandwidth remaining percent 5
  class AutoQos-VoIP-Control-QosGroup24
    set dscp cs3
    set cos 3
    bandwidth remaining percent 5
  class class-default
    dbl
```

**Note**

There are no default cos-to-dscp or dscp-to-cos mappings on the. Values must be explicitly set for trunks.

```
Policy Map AutoQos-4.0-Output-Policy
  Class AutoQos-4.0-Scavenger-Queue
    bandwidth remaining percent 1
  Class AutoQos-4.0-Priority-Queue
    priority
    police cir percent 30 bc 33 ms
    conform-action transmit
    exceed-action drop
```

```

Class AutoQos-4.0-Control-Mgmt-Queue
  bandwidth remaining percent 10
Class AutoQos-4.0-Multimedia-Conf-Queue
  bandwidth remaining percent 10
Class AutoQos-4.0-Multimedia-Stream-Queue
  bandwidth remaining percent 10
Class AutoQos-4.0-Trans-Data-Queue
  bandwidth remaining percent 10
  dbl
Class AutoQos-4.0-Bulk-Data-Queue
  bandwidth remaining percent 4
  dbl
Class class-default
  bandwidth remaining percent 25
  dbl

```

The three policy maps are defined as follows:

- **policy-map AutoQos-VoIP-Input-Dscp-Policy**
This policy map is applied as an input service policy on an Layer 3 interface (such as an uplink connection to a neighboring switch) when auto-QoS is configured on the port.
- **policy-map AutoQos-VoIP-Input-Cos-Policy**
This policy map is applied as an input service policy on an Layer 2 interface that could be either an uplink connection or a port hooked to a Cisco IP Phone.
- **policy-map AutoQos-VoIP-Output-Policy**
This policy map is applied as an output policy for any port on which auto-QoS is configured, establishing policy governing egress traffic on the port based on whether it is voice data or control traffic.

The purpose of the input policy maps is to identify voice data or control traffic and mark it as such as it traverses the switch. The output policy map matches the packets on the marking occurring on ingress and then applies egress parameters such as bandwidth, policing and/or priority queuing.

For switch-to-switch connections, the **[no] auto qos voip trust** command is used to apply an input and output service policy on the interface:

```
service-policy input AutoQos-VoIP-Input-Cos-Policy
```

OR

```
service-policy input AutoQos-VoIP-Input-Dscp-Policy
```

AND

```
service-policy output AutoQos-VoIP-Output-Policy
```

The selection of the input policy depends on whether the port is Layer 2 or Layer 3. For Layer 2, the policy trusts the Cos setting in the received packets. For Layer 3 ports, it relies on the DSCP value contained in the packets.

For phone connected ports, the **[no] auto qos voice cisco-phone** command is used to apply the following service policy to the port:

```
qos trust device cisco-phone
```

```
service-policy input AutoQos-VoIP-Input-Cos-Policy
```

AND

```
service-policy output AutoQos-VoIP-Output-Policy
```

It establishes a trusted boundary that recognizes Cisco IP Phones and trusts the Cos setting of the packets from the phone. If a Cisco IP Phone is not detected, the Cos field is ignored and the packets are not classified as voice traffic. Upon detecting a Cisco phone, the ingress packets are marked based on the Cos value in the packets. This marking is used on egress for proper traffic classification and handling.

Auto qos srnd4—Is generated when any new auto qos command is configured on an interface and migrates from legacy CLIs to generate new configurations. This CLI only generates a global configuration if during migration, one or more interfaces has legacy auto-QoS enabled

Auto qos video—Generates QoS configuration for untrusted interfaces. It incorporates a service-policy to classify the traffic coming from untrusted desktops/devices and marks them accordingly.

Auto qos void cisco-softphone—Generate QoS configuration for interfaces connected to PCs running the Cisco IP SoftPhone application and marks as police traffic stemming from such interfaces. Ports configured with this CLI are considered untrusted.

Auto qos classify—Generates QoS configuration for untrusted interfaces. It applies a service-policy to classify the traffic stemming from untrusted desktops or devices and marks them accordingly. The service-policies generated do not police.

Auto-QoS Compact

When you enter an auto-QoS command, the switch goes on to display all the generated commands as if the commands were entered from the CLI. Enable auto-QoS compact if you want to hide auto-QoS generated commands from the running configuration.

To enable auto-QoS compact, enter this command:

Command	Purpose
Switch# configure terminal	Enters global configuration mode.
Switch(config)# auto qos global compact	<p>Enables auto-QoS compact and generates global auto-QoS configurations that are hidden from the running configuration.</p> <p>You can then enter the auto-QoS command you want to configure, in the interface configuration mode. The interface commands that the system generates are also hidden from running configuration.</p> <p>To display the auto-QoS configuration that has been applied, use these the privileged EXEC commands: show derived-config, show policy-map, show policy-map interface, show class-map, show table-map, and show auto-qos, show ip access-list.</p>

To disable auto-QoS compact, first remove all the auto-QoS configuration by entering the **no** form of the corresponding auto-QoS interface configuration command and then enter the **no auto qos global compact** global configuration command. Disabling auto-QoS compact is a required task when have to perform an ISSU from an image where auto-QoS compact is supported to an image where this feature is not available.

The following example shows how to enable auto-QoS compact and configure the **auto qos voip cisco-phone** interface configuration command and then display configuration details:

```
Switch# configure terminal
Switch(config)# auto qos global compact
Switch(config)# interface GigabitEthernet1/2
Switch(config-if)# auto qos voip cisco-phone

Switch(config-if)# show auto-qos
GigabitEthernet1/2
auto qos voip cisco-phone
```

Effects of Auto-QoS and Auto-Qos Compact on Running Configuration

The auto-QoS interface configuration commands and the generated global configurations are added to the running configuration. When you save this configuration, all generated commands and any user-entered configuration that was not overridden, is saved to memory.

If auto-QoS compact is enabled, only the list of auto-QoS commands you have entered are displayed in the running configuration. The generated global and interface configurations are hidden from the running configuration. When you save this configuration, only the auto-QoS commands you have entered are saved (and not the hidden configuration). When you reload the switch, the system detects and re-executes the saved auto-QoS commands, re-sets the values, and generates an auto-QoS SRND4.0 complaint configuration set.



Note

Do not make changes to the auto-QoS-generated commands when auto-QoS compact is enabled, because user-modifications are overridden when the switch reloads.



Configuring AVC with DNS-AS

The Application Visibility Control (AVC) with Domain Name System as an Authoritative Source (DNS-AS) feature (AVC with DNS-AS) provides a centralized means of controlling the identification and classification of trusted network traffic in an organization. It accomplishes this by using:

- Network metadata stored in a DNS server that is authoritative to the domain in question, to identify applications
- Modular QoS CLI (MQC), to classify the corresponding traffic and apply suitable policies
- Flexible NetFlow (FNF), to monitor and export application information to an external collector.

Starting with Cisco IOS XE Release 3.9.0E, the feature is available on Catalyst 4500E Series Switches with Supervisor Engine 8-E, 8L-E, 7-E, 7L-E, and Catalyst 4500-X Series Switches.

Starting with Cisco IOS XE Release 3.9.2E, you can export application information using FNF.

Starting with Cisco IOS XE Release 3.10.0E, support is extended to Supervisor Engine 9-E.

Benefits of the feature:

- Application Visibility—Ensuring unambiguous visibility of applications.

The DNS-AS mechanism snoops requests and does not require a CPU-intensive, deep packet inspection (DPI). Since traffic classification is by means of a DNS request and not DPI, this feature is compatible in scenarios where network traffic is encrypted.

- Metadata Driven—Using information about applications.

This enables you to holistically program the network so it behaves like a self-driving car. You now have information about all the required applications in your network, irrespective of whether traffic is encrypted or not.

- Centralized Control—Using a cross-domain application intent policy controller.

The feature leverages an existing, universally available query-response mechanism, to enable local DNS servers within an organization to act as authoritative servers and propagate application classification information to client devices (switches) in an enterprise network.

- Control without Administrative Access—Proving alternatives to controller-based approaches.

The feature supports scenarios where your network may be in the cloud and you may not own it. You can still control network devices across the Internet, even though you may not have administrative control of these devices.

This chapter describes how to configure AVC with DNS-AS. It includes the following major sections:

- [About AVC with DNS-AS, page 45-3](#)
- [Configuring AVC with DNS-AS, page 45-7](#)

- [Monitoring AVC with DNS-AS, page 45-21](#)
- [Troubleshooting AVC with DNS-AS, page 45-25](#)

About AVC with DNS-AS

- [Overview, page 45-3](#)
- [Key Concepts, page 45-3](#)
- [AVC with DNS-AS Process Flow, page 45-5](#)
- [High Availability and ISSU for AVC with DNS-AS, page 45-6](#)
- [Default Configuration, page 45-7](#)

Overview

The process starts with an organization's requirements relating to management and control of network traffic. You begin by assessing—the software applications that run on the various hosts (phones, PCs etc.) in your network, the domains (websites) and applications accessed by these devices, and the business-relevance of these domains and applications in your organization.

The assessment helps you arrive at a list of domains and applications that are “trusted” by your organization - designating all remaining domains and applications as untrusted.

With DNS-AS enabled on your network and the list of trusted domains at hand, the networking devices or DNS-AS clients in your network identify which applications the network traffic belongs to or which domains are being requested. As long as the traffic is part of the trusted list, the switch requests the DNS server for metadata and IP address information. This request is sent in the form of a DNS-query. The response, once received, is cached locally until the Time-to-Live (TTL) for that resource record expires. The response is bound to the traffic and allows the DNS-AS client to now identify, classify, and forward traffic accordingly.

Key Concepts

Metadata (RFC6759)	<p>In the context of the AVC with DNS-AS feature, this includes traffic classification information, application identification information, and business relevance information.</p> <p>Metadata is maintained in the form of TXT records. The following is a sample metadata record in the prescribed format:</p> <p>CISCO-CLS=app-name:example app-class:TD business:YES app-id:CU/28202</p>
Forward look-up	<p>A request for an IP address or a request for an “A” record, originating from a host.</p> <p>Being able to snoop these forward lookups in the network traffic is fundamental to the DNS-AS feature.</p>
Host	<p>A PC or mobile where users run software applications, access websites and so on.</p> <p>Only hosts with a wired connection to the network are considered.</p> <p>Forward look-up requests originate from hosts.</p>

Client or DNS-AS client	<p>Networking devices throughout your network. Host traffic is always routed through such a client.</p> <p>Note This configuration chapter deals with DNS-AS configuration on Cisco Catalyst Switches that are deployed as access switches only. Throughout this document, the term client, DNS-AS client, refers to the switch where AVC with DNS-AS is enabled.</p> <p>DNS-AS Clients receive metadata from an authoritative DNS server and maintain a database of this information in the form of records. How long the record remains in the client's database, is determined by the record's TTL.</p>
Binding table	<p>A table that resides in the client and serves as a database of parsed DNS server responses [TXT records and "A" records].</p> <p>Every client has a binding table of its own.</p>
An "A" record	<p>A record containing the domain name and IP address information [Only IPv4 address]. This is one of the DNS-Server responses (the other being the TXT record) and has a predefined lifespan.</p> <p>A forward lookup request from a host is a request for an "A" record.</p>
TXT DNS-AS resource record or TXT record	<p>A record containing metadata. This is one of the DNS-Server responses (the other being the "A" record) and has a predefined lifespan.</p> <p>A TXT record is limited to 255 characters.</p> <p>For AVC with DNS-AS, the TXT attribute is always CISCO-CLS. Any TXT record that starts with CISCO-CLS= can be recognized as a DNS-AS message.</p> <p>Syntax— CISCO-CLS=<option>:<val>{ <option>:<val> }*</p>
Time-to-Live (TTL)	<p>The lifespan of an "A" record and TXT record in the binding table.</p> <p>TTL values are configured on the DNS server.</p> <p>While a TTL accompanies both TXT and "A" record responses, the DNS client only goes by the "A" record response from the DNS server.</p>
Authoritative DNS server	<p>The go-to DNS server for all client metadata and "A" record requests.</p> <p>Every DNS domain has only one authoritative DNS server.</p> <p>Such a server maintains records of application metadata in the form of a TXT record, and only returns responses to queries about domain names that have been maintained in the required format.</p> <p>The following is a sample metadata record in the prescribed format: CISCO-CLS=app-name:example app-class:TD business:YES app-id:CU/28202</p>

AVC with DNS-AS Process Flow

This involves the DNS snooping process and the DNS-AS client process—both of which are loosely coupled, but independent processes. [Figure 45-1](#) is a representation of both processes.

Part -I: DNS Snooping Process

-
- Step 1** The host initiates an “A” record request.
- A user from your organization is in a meeting room in an office building. The associated DNS-AS client here is a switch (the wired network traffic from this meeting room is routed through this switch). The user looks up a website `www.example.com`, which initiates the request for an “A” record.
- Step 2** The authoritative DNS-server responds with an “A” record response.

Part-II: DNS-AS Client Process


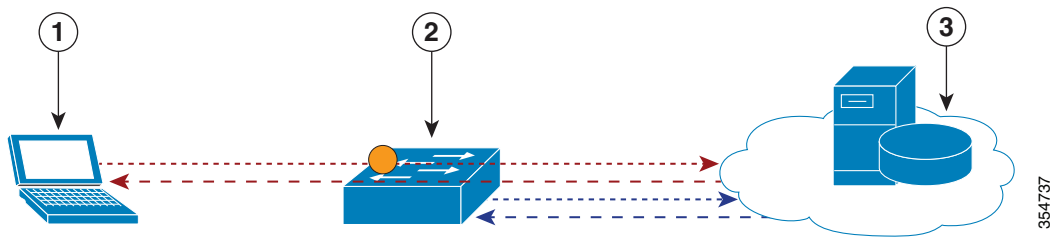
-
- Step 1** The DNS-AS client sends a DNS query (TXT request) to the authoritative DNS server.
- The DNS-AS client, which is constantly snooping for requests (based on the trusted domain list), finds the host’s forward look-up request.
-  **Note** The DNS-AS client receives a copy of the host’s “A” record request, and does not alter the host’s original request in any manner.
-
- Based on the snooped result, the DNS-AS client sends a TXT request to the authoritative DNS server.
- Step 2** The authoritative DNS-server responds with a TXT record response.
- Step 3** A successful TXT response is followed by an “A” record request.
- Step 4** The authoritative DNS-server responds with an “A” record response.
- Step 5** The DNS-AS client parses and saves the response in its binding table.
- The DNS-AS client saves the TXT record and “A” record in its binding table. The response will remain saved in the binding table for the duration specified by the TTL of the “A” record. The system automatically checks and prevents duplicate entries for a fully qualified domain name in the binding table.
- The DNS-AS client applies a QoS policy based on the metadata from the DNS server, and exports application information to a collector, based on how the flow record is configured.
- The DNS-AS client forwards information about identified applications to FNF, enabling you to export this information.

Figure 45-1 AVC with DNS-AS Process Flow



1	Host	3	Authoritative DNS Server
2	DNS-AS Client		

	An “A” record request from the host to the DNS server		An “A” record response from the DNS server to the host
	A copy of the host’s “A” record request that the DNS-AS client saves	—	—
	TXT record and “A” record request from the DNS-AS client to the DNS server		TXT record and “A” record response from the DNS server to the DNS-AS client

High Availability and ISSU for AVC with DNS-AS

The AVC with DNS-AS feature supports High Availability and ISSU.

For High Availability, the binding table database of the active DNS-AS client is synchronized with the standby DNS-AS client. As long as AVC with DNS-AS is enabled, no additional user configuration is required.

The binding table entries are synchronized when:

- The standby comes up (bulk synchronization).
- New entries are added to the binding table database.
- One or more entries are cleared from the database.

Note

AVC with DNS-AS is also supported in the VSS mode, and Quad-Supervisor VSS Mode.

Default Configuration

AVC with DNS-AS is disabled.

Configuring AVC with DNS-AS

- [Prerequisites for Configuring AVC with DNS-AS, page 45-7](#)
- [Restrictions and Guidelines for Configuring AVC with DNS-AS, page 45-7](#)
- [Generating Metadata Streams, page 45-8](#)
- [Configuring a DNS Server as the Authoritative Server, page 45-10](#)
- [Enabling AVC with DNS-AS, page 45-10](#)
- [Making an Entry in the Trusted Domain List, page 45-11](#)
- [Configuring QoS for AVC with DNS-AS, page 45-12](#)
- [Configuring FNF for AVC with DNS-AS, page 45-16](#)

Prerequisites for Configuring AVC with DNS-AS

- The DNS-AS client can snoop forward look-up requests originating from hosts.
- To ensure DNS packet logging or snooping, you must attach the policy map to the interface, by using the **service-policy input** command.
- You have maintained metadata in the authoritative DNS server and reachability exists - before you enable AVC with DNS-AS.

Restrictions and Guidelines for Configuring AVC with DNS-AS

- Only a forward look-up is supported.
- Two DNS servers are supported, in case of a failover. One is considered the primary DNS server and other, the secondary DNS server.
- IPv6 is not supported—AAAA requests, and IPv6 DNS servers are not supported.
- AVC with DNS-AS is supported only on physical interfaces, in the ingress direction.
- AVC with DNS-AS is not supported on wireless traffic.
- Virtual Routing and Forwarding (VRF) is not supported.
- We recommend a maximum of 300 AVC with DNS-AS applications (domain names) in the binding table, because of its effect on the ternary content addressable memory (TCAM). To know how the addition of applications affects the TCAM see the [Troubleshooting AVC with DNS-AS, page 45-25](#) section of this chapter

Generating Metadata Streams

Application metadata is configured and saved on the local, authoritative DNS server. You configure application classification information, for each trusted domain, in a prescribed format (a metadata stream). This is the information that the server propagates to switches when queried for application metadata. When the switch sends a TXT query regarding an application, the DNS server sends the relevant metadata in the TXT response.

To generate metadata streams, perform the following task:

Step 1	Command or Action	Purpose
	<p>Go to the AVC Resource Record Generator at: https://www.dns-as.org/support/avc-r-data</p>	<p>Helps you generate a metadata stream for an application or domain, in a TXT record format.</p> <p>You can specify the following metadata fields:</p> <ul style="list-style-type: none"> • (Optional) Domain Name • (Mandatory) Application Name—A value is mandatory. This can be an existing application name or custom application name. <ul style="list-style-type: none"> – Existing Application Name (app-name:)—Select from the list of standard applications. – Custom Application Name(app-name:)—If you enter a custom application name, you must also maintain the Traffic Class and Business Relevance information in the metadata stream. • (Optional) Selector ID (app-id:)—Consists of a classification engine ID (first eight bits) and a selector ID (the next twenty-four bits). <ul style="list-style-type: none"> – Classification Engine ID—Defines the context for the selector ID. Only these engine IDs are allowed: <ul style="list-style-type: none"> L3—IANA layer 3 protocol number L4—IANA layer 4 well-known port number L7—Cisco global application ID CU—Custom protocol. Use this engine ID for custom application names. – Selector ID—An application identifier, for a given classification engine ID. Enter a numeric value between 1 and 65535. <p>Note When you enter the engine ID and selector ID for existing application names, be sure to align with the Network Based Application Recognition (NBAR) standard. Only then will the FNF exporters report with a common ID and in a consistent manner.</p> <ul style="list-style-type: none"> • (Optional) Port Range (server-port:) • (Optional) Traffic Class (app-class:) • (Optional) Business Relevance (business:)—If you do not select yes or no, the business relevance value is set based on the app-class or app-name — in that order. <p>For information about how traffic class and business relevance fields here map to QoS traffic classification, see Table 45-1app-class and QoS Traffic Mapping.</p> <p>Sample metadata stream: CISCO-CLS=app-name:example app-class:TD business:YES app-id:CU/28202</p>

	Command or Action	Purpose
Step 2	Click Generate predefined	Generate predefined —Generates a predefined metadata stream for standard applications, using best practice defaults.
	OR Click Generate custom	Generate custom —Generates a custom metadata stream for your applications, using the custom values you have entered.
Step 3	Copy metadata into the corresponding TXT Resource Record of the DNS server in charge of the DNS domain that you have marked as a trusted domain.	Copy and paste the metadata stream from the website, to the authoritative DNS server you are using.

Configuring a DNS Server as the Authoritative Server

All DNS-AS clients in the network should be configured to send all DNS queries to one authoritative DNS server. On a Cisco Catalyst switch, perform the following task:

	Command or Action	Purpose
Step 1	configure terminal	Enters global configuration mode.
	Example: Switch# configure terminal	
Step 2	ip name-server server-address	Specifies the address of the authoritative DNS server. The port number is always 53.
	Example: Switch(config)# ip name-server 192.0.2.1 192.0.2.2	You can configure up to two DNS servers, in case of a failover.
		Note The command allows you configure up to six name servers (IPv4 and IPv6). Ensure that at least the first two IP addresses in the sequence are IPv4 addresses, because the AVC with DNS-AS feature will use only these. See the example below, here the first two addresses are IPv4 (192.0.2.1 and 192.0.2.2), the third one (2001:DB8::1) is an IPv6 address. AVC with DNS-AS will use the first two. Switch(config)# ip name-server 192.0.2.1 192.0.2.2 2001:DB8::1

Enabling AVC with DNS-AS

DNS-AS is disabled by default. To enable the feature on a Cisco Catalyst switch, perform the following task:

	Command or Action	Purpose
Step 1	configure terminal Example: Switch# configure terminal	Enters global configuration mode.
Step 2	[no] avc dns-as client enable Example: Switch(config)# avc dns-as client enable	<p>Enables AVC with DNS-AS on the switch (DNS-AS client).</p> <p>The system then creates a binding table where parsed DNS server responses are stored till the TTL expires.</p> <p>Note To ensure DNS packet logging or snooping, you must attach the policy map (containing the relevant class maps that will determine traffic class) to the interface by using the service-policy input command. For more information see Configuring QoS for AVC with DNS-AS, page 45-12.</p>

Making an Entry in the Trusted Domain List

When AVC with DNS-AS is first enabled on the switch, the trusted domain list is empty. You must maintain the list of trusted domains on the switch. The switch snoops only for network traffic that is maintained in this list. To make entries in this list, perform the following task

	Command or Action	Purpose
Step 1	configure terminal Example: Switch# configure terminal	Enters global configuration mode.
Step 2	[no] avc dns-as client trusted-domains Example: Switch(config)# avc dns-as client trusted-domains	Enters the trusted domain configuration mode.
Step 3	[no] domain domain-name Example: Switch(config-trusted-domains)# domain www.example.com	<p>Enter the domain name. This forms part of the list of trusted domains for the DNS-AS client. All remaining domains are ignored and will follow default forwarding behavior.</p> <p>You can enter up to 50 domains.</p> <p>You can use regular expressions to match the domain name. For example, to represent all the domains for an organization, if you enter:</p> <pre>Switch(config-trusted-domains)# domain *.example.*</pre> <p>The DNS-AS client matches www.example.com, ftp.example.org and any other domain that pertains to the organization “example”.</p> <p>But use such an entry at your discretion, because it could increase the size of the binding table considerably.</p>

Configuring QoS for AVC with DNS-AS

In order to isolate and classify trusted traffic as defined in the metadata stream, you must complete this sequence of tasks—create class maps (one for each traffic class), define traffic-class match criteria and business-relevance match criteria, create a policy map, attach the policy map to the interface. This sub-section provides the following information:

- [Class Map Configuration in the Easy QoS Model, page 45-12](#)
- [Policy Map Definitions in the Easy QoS Model, page 45-13](#)
- [App-Class and QoS Traffic Mapping, page 45-13](#)
- [Sample QoS Configuration for AVC with DNS-AS: Classifying Network Control Traffic, page 45-14](#)

For more QoS information, see the [Classification, page 44-6](#) section of the Configuring QoS chapter in this guide.

Class Map Configuration in the Easy QoS Model

In order to determine how many traffic classes should be provisioned, you can use the 12-class Easy QoS Model. This model provides uniform, standards-based recommendations to help ensure that QoS designs and deployments are unified and consistent across an organization.

The following shows the class map configuration for traffic class and business relevance as per the 12-class Easy QoS Model:

```
class-map match-all VOICE
  match protocol attribute traffic-class voip-telephony
  match protocol attribute business-relevance business-relevant
class-map match-all BROADCAST-VIDEO
  match protocol attribute traffic-class broadcast-video
  match protocol attribute business-relevance business-relevant
class-map match-all REAL-TIME-INTERACTIVE
  match protocol attribute traffic-class real-time-interactive
  match protocol attribute business-relevance business-relevant
class-map match-all MULTIMEDIA-CONFERENCING
  match protocol attribute traffic-class multimedia-conferencing
  match protocol attribute business-relevance business-relevant
class-map match-all MULTIMEDIA-STREAMING
  match protocol attribute traffic-class multimedia-streaming
  match protocol attribute business-relevance business-relevant
class-map match-all SIGNALING
  match protocol attribute traffic-class signaling
  match protocol attribute business-relevance business-relevant
class-map match-all NETWORK-CONTROL
  match protocol attribute traffic-class network-control
  match protocol attribute business-relevance business-relevant
class-map match-all NETWORK-MANAGEMENT
  match protocol attribute traffic-class ops-admin-mgmt
  match protocol attribute business-relevance business-relevant
class-map match-all TRANSACTIONAL-DATA
  match protocol attribute traffic-class transactional-data
  match protocol attribute business-relevance business-relevant
class-map match-all BULK-DATA
  match protocol attribute traffic-class bulk-data
  match protocol attribute business-relevance business-relevant
class-map match-all SCAVENGER
  match protocol attribute business-relevance business-irrelevant
```


Policy Map Definitions in the Easy QoS Model

The following sample output displays the policy map definitions, with traffic attribute marking for all the traffic classes in the 12-class Easy QoS Model:

```

policy-map MARKING
class VOICE
  set dscp ef
class BROADCAST-VIDEO
  set dscp cs5
class REAL-TIME-INTERACTIVE
  set dscp cs4
class MULTIMEDIA-CONFERENCING
  set dscp af41
class MULTIMEDIA-STREAMING
  set dscp af31
class SIGNALING
  set dscp cs3
class NETWORK-CONTROL
  set dscp cs6
class NETWORK-MANAGEMENT
  set dscp cs2
class TRANSACTIONAL-DATA
  set dscp af21
class BULK-DATA
  set dscp af11
class SCAVENGER
  set dscp cs1
class class-default
  set dscp default

```

App-Class and QoS Traffic Mapping

The following table shows how the app-class field in the metadata stream maps to the 12-class Easy QoS Model of traffic classification:



Note

The DNS-AS client applies default forwarding behavior in these cases:

- If the match attributes that you specify for the traffic class and business relevance do not match what you have defined in the metadata stream.
- If the binding table entry is no longer active. This refers to the age of the entry. Use the **show avc dns-as client binding-table** command to display the age of an entry

Table 45-1 app-class and QoS Traffic Mapping

app-class: <Long Text>	app-class: <Short text>	Corresponding Traffic Class and Business Relevance Label in the 12-Class Easy QoS Model
app-class: VOIP-TELEPHONY	app-class: VO	Traffic-class = voip-telephony Business-relevance = YES
app-class: BROADCAST-VIDEO	app-class: BV	Traffic-class = broadcast-video Business-relevance = YES

Table 45-1 app-class and QoS Traffic Mapping

app-class: REALTIME-INTERACTIVE	app-class: RTI	Traffic-class = real-time-interactive Business-relevance = YES
app-class: MULTIMEDIA-CONFERENCING	app-class: MMC	Traffic-class = multimedia-conferencing Business-relevance = YES
app-class: MULTIMEDIA-STREAMING	app-class: MMS	Traffic-class = multimedia-streaming Business-relevance = YES
app-class: NETWORK-CONTROL	app-class: NC	Traffic-class = network-control Business-relevance = YES
app-class: SIGNALING	app-class: CS	Traffic-class = Signaling Business-relevance = YES
app-class: OPS-ADMIN-MGMT	app-class: OAM	Traffic-class = ops-admin-mgmt Business-relevance = YES
app-class: TRANSACTIONAL-DATA	app-class: TD	Traffic-class = Transactional-Data Business-relevance = YES
app-class: BULK-DATA	app-class: BD	Traffic-class = bulk-data Business-relevance = YES
app-class: BEST-EFFORT	app-class: BE	Traffic-class = <no change> Business-relevance = default
app-class: SCAVENGER	app-class: SCV	Traffic-Class = <no change> Business-relevance = NO

Sample QoS Configuration for AVC with DNS-AS: Classifying Network Control Traffic

The following example shows how to classify network-control traffic based on the 12-class Easy QoS model. It shows how the DNS-AS client allows “example.org” to be classified under class-map NETWORK-CONTROL.

For this example, the corresponding metadata that should be maintained is:

```
CISCO-CLS=app-name:example|app-class:NC|business:YES
```

Create class maps and match attributes

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# class-map NETWORK-CONTROL
Switch(config-cmap)# match protocol attribute traffic-class network-control
Switch(config-cmap)# match protocol attribute business-relevance business-relevant
Switch(config-cmap)# end
```

Create the policy map, attach the class map to it and specify priority

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# policy-map MARKING
Switch(config-pmap)# class NETWORK-CONTROL
Switch(config-pmap-c)# set dscp ef
Switch(config-pmap-c)# end
Switch#
```

Attach the policy map to an interface

```
Switch# configure terminal  
Enter configuration commands, one per line. End with CNTL/Z.  
Switch(config)# interface tengigabitethernet 1/0/1  
Switch(config-if)# service-policy input MARKING  
Switch(config-if)# end
```

Configuring FNF for AVC with DNS-AS

With FNF, you can gain visibility into the applications running on your network, and use FNF option templates to export application ID, description, and attribute information.

You must configure these FNF settings on the DNS-AS client:

- Configure a flow record to collect nonkey field **application-name**, and the key fields **ipv4 source address** and **ipv4 destination address**
- Configure a flow exporter and the two option templates, to fetch application information.

Option template **application-table**, exports only applications resolved by the DNS-AS client, that is, the application ID and name from the binding table. The corresponding application descriptions come from Network Based Application Recognition (NBAR) definition for standard applications. A constructed help string is used for custom applications.

Option **application-attributes** fetches attribute information by mapping it to the application name. Where standard application names are used, the option template uses standard NBAR attribute definitions; where custom application names are used, user-defined application names and only certain attribute fields are guaranteed to carry values.

- Configure a flow monitor and apply it to an interface to enable network traffic monitoring.

FNF Interaction with DNS-AS—With every flow that is created in the flow table, the DNS-AS client resolves the application name for the flow (if the entry exists in the binding table), by using the destination IP address (and if not available), the source IP address.

At periodic, configured intervals (600 seconds, by default), FNF exports option template data, that is mapped to the corresponding application name, to an external collector.

For more information about FNF, see the [Configuring Flexible NetFlow](#) chapter in this guide.

These sections provide more information:

- [Option Templates](#), page 45-16
- [Sample FNF Configuration for AVC with DNS-AS](#), page 45-18

Option Templates

The **application-table** and **application-attributes** options templates are supported. These templates determine the information that will be exported to an external collector.

option application-table

Exports the application name, application tag, and description to the external collector.

On a device where AVC with DNS-AS is enabled, only applications resolved by the DNS-AS client are exported. But in addition, the application-table template exports two applications called *unclassified* and *unknown*, irrespective of whether the feature is enabled or not.

- **Application Name**—For custom and standard applications, this information is derived from the TXT response (**app-name**;) that is saved in the binding table.
- **Application Tag**—This is same as application ID in the context of the AVC with DNS-AS feature and consists of the engine ID and selector ID.

- **Engine ID or Classification Engine ID**—Defines the context for the selector ID. Only these values are supported:

L3—IANA layer 3 protocol number (IANA_L3_STANDARD, ID: 1)

L4—IANA layer 4 well-known port number (IANA_L4_STANDARD, ID: 3)

L7—Cisco global application ID (CISCO_L7_GLOBAL, ID: 13)

CU—Custom protocol, (NBAR_CUSTOM, ID: 6). For custom applications, the DNS-AS client automatically uses this engine ID.

- Selector ID—Uniquely identifies the application or classification.

For standard applications, the application tag information is derived from these sources, in the given order of precedence:

1. TXT response (**app-id**.)
2. The NBAR definition for standard applications (if the TXT response does not carry a value)

For custom applications, the following applies to application tag information:

1. It is derived only from the TXT response (**app-id**.)
 2. For the engine ID, the DNS-AS client automatically uses CU—Custom protocol, (NBAR_CUSTOM, ID: 6).
 3. For the selector ID, the DNS-AS client allots a custom selector ID. A maximum of 120 custom applications are supported - out of which 110 are available to the DNS-AS client. Starting with selector ID value 243, IDs are assigned in descending order. When there are no remaining IDs to assign, the entry is not saved in the binding table.
- Description—This information is derived from the NBAR definition for standard applications. For custom applications, the DNS-AS client uses: User Defined Protocol <app-name>.

option application-attributes

Enables the collector to map the application names (from the **option application-table**) to their attributes. Attributes are statically assigned to each protocol or application, and are not dependent on traffic.

For standard applications—

- Application Tag—Guidelines that apply this field as part of the option application-table template apply here as well.
- Category—Groups applications based on the first level of categorization for each protocol as the match criteria. Similar applications are grouped together under one category. For example, the email category contains all email applications such as, Internet Mail Access Protocol (IMAP), Simple Mail Transfer Protocol (SMTP), Lotus Notes, and so on.
- Sub-category—Groups applications based on the second level of categorization for each protocol as the match criteria. For example, clearcase, dbase, rda, mysql and other database applications are grouped under the database group.
- Application Group—Groups the same networking applications together. For instance, Example-Messenger, Example-VoIP-messenger, and Example-VoIP-over-SIP are grouped together under the example-messenger-group
- Peer-to-peer (p2p)—Groups protocols based on whether or not they use p2p technology.
- Tunnel—Groups protocols based on whether or not a protocol tunnels the traffic of other protocols. Protocols for which the NBAR does not provide any value are categorized under the unassigned tunnel group. For example, Layer 2 Tunneling Protocols (L2TP).
- Encryption—Groups applications based on the encrypted and nonencrypted status of the applications. Protocols for which the NBAR does not provide any value are categorized under the unassigned encrypted group.

- Traffic class—Groups applications and protocols based on the traffic class they belong to. For example, all applications that have traffic class `TD`.

Traffic class information is derived from these sources, in the given order of precedence:

1. TXT response (**app-class:**)
 2. The NBAR definition for standard applications (if the TXT response does not carry a value)
- Business relevance—Groups applications based on whether or not they have been marked as business-relevant. For example, all applications that have business relevance as `YES`.

Business relevance information is derived from these sources, in the given order of precedence:

1. TXT response (**business:**)
2. The NBAR definition for standard applications (if the TXT response does not carry a value)

For custom applications—

Only these attributes of the application-attributes option template are guaranteed to carry a value:

- Application Tag—See the Application Tag info in section [option application-table, page 45-16](#) above. The same applies here as well.
- Traffic class—This information is derived from the TXT response (**app-class:**)
- Business Relevance—This information is derived from the TXT response (**business:**)

Sample FNF Configuration for AVC with DNS-AS

The following example shows how you can configure FNF for AVC with DNS-AS:

1. Create a flow record. As in the example, you must configure:
 - The source and destination IP addresses as key fields, in order to resolve application names.
 - The use of the application name as a nonkey field in flow record.

Additionally (not mandatory), you can also configure the number of bytes or packets in a flow as a nonkey field, to display the number of applications sent to the collector.

```
Switch# configure terminal
Switch(config)# flow record example-record1
Switch(config-flow-record)# match ipv4 source address
Switch(config-flow-record)# match ipv4 destination address
Switch (config-flow-record)# collect application name
Switch (config-flow-record)# collect counter packets
Switch (config-flow-record)# exit
```

```
Switch# show flow record example-record1
flow record example-record1
match ipv4 source address
match ipv4 destination address
collect application name
collect counter packets
```

2. Create a flow exporter. Also configure the **application-table** and **application-attributes** option templates in the exporter. Without option templates, the collector cannot retrieve meaningful application information. At a minimum we recommend that you configure the application-table option. For attribute information, also configure the application-attribute option.

You can also change the frequency of template export in seconds (the allowed range is 1 to 86400 seconds; the default is 600 seconds)

```
Switch(config)# flow exporter example-exporter1
```

```

Switch(config-flow-exporter)# option application-table
Switch(config-flow-exporter)# option application-attributes
Switch(config-flow-exporter)# template data timeout 500
Switch(config-flow-exporter)# exit

Switch#show flow exporter example-exporter1
Flow Exporter example-exporter1:
  Description:                User defined
  Export protocol:            NetFlow Version 9
  Transport Configuration:
    Destination IP address:    192.0.1.254
    Source IP address:        192.51.100.2
    Transport Protocol:       UDP
    Destination Port:         9995
    Source Port:              54964
    DSCP:                     0x0
    TTL:                      255
    Output Features:          Not Used
  Options Configuration:
    application-table (timeout 500 seconds)
    application-attributes (timeout 500 seconds)

Switch# show flow exporter example-exporter1 statistics
Flow Exporter example-exporter1:
  Packet send statistics (last cleared 00:00:48 ago):
    Successfully sent:         2                               (924 bytes)

  Client send statistics:
    Client: Option options application-name
      Records added:           4
      - sent:                  4
      Bytes added:             332
      - sent:                  332

    Client: Option options application-attributes
      Records added:           2
      - sent:                  2
      Bytes added:             388
      - sent:                  388

```

3. Create a flow monitor and apply it to an interface to perform network traffic monitoring.

The interface you apply the flow monitor to, can also be the same interface you have applied the QoS policy to. This example applies the QoS policy created as part of the sample QoS configuration [Sample QoS Configuration for AVC with DNS-AS: Classifying Network Control Traffic](#), page 45-14.

```

Switch# configure terminal
Switch(config)# flow monitor example-monitor1
Switch(config-flow-monitor)# record example-record1
Switch(config-flow-monitor)# exporter exporter-export1
Switch(config-flow-monitor)# exit
Switch(config)# interface tengigabitethernet 1/0/1
Switch(config)# switchport access vlan 100
Switch(config)# switchport mode access
Switch(config-if)# ip flow monitor example-monitor1 input
Switch(config-if)# service-policy input MARKING
Switch(config-if)# end

Switch# show flow monitor
flow monitor example-monitor1
  record example-record1
  exporter example-exporter1
!

```

```

!
interface tengigabitethernet1/0/1
 switchport access vlan 100
 switchport mode access
 ip flow monitor example-monitor1 input

Switch# show flow monitor example-monitor1 cache
Cache type: Normal
Cache size: 16640
Current entries: 3
High Watermark: 3

Flows added: 6
Flows aged: 3
- Active timeout ( 1800 secs) 0
- Inactive timeout ( 30 secs) 3
- Event aged 0
- Watermark aged 0
- Emergency aged 0

IPV4 SOURCE ADDRESS: 192.0.1.254
IPV4 DESTINATION ADDRESS: 192.51.100.2
counter packets long: 7479
application name: appexample1

IPV4 SOURCE ADDRESS: 192.51.100.11
IPV4 DESTINATION ADDRESS: 203.0.113.125
counter packets long: 445
application name: appexample2

IPV4 SOURCE ADDRESS: 192.51.51.51
IPV4 DESTINATION ADDRESS: 203.0.113.100
counter packets long: 14325
application name: appexample3
Switch#

```

4. Other related show commands:

```

Switch# show avc dns-as client binding-table detail
DNS-AS generated protocols:
Max number of protocols :50
Customization interval [min] :N/A

Age      : The amount of time that the entry is active
TTL      : Time to live which was learned from DNS-AS server
Time To Expire : Entry expiration time in case device does not see DNS traffic for
the entry host

Protocol-Name : appexample1
VRF           : <default>
Host          : www.appexample1.com
Age[min]      : 2
TTL[min]      : 60
Time To Expire[min] : 58
TXT Record    : app-name:appexample1|app-class:VO|business:YES
Traffic Class : voip-telephony
Business Relevance : business relevant
IP            : 192.0.1.254

Protocol-Name : appexample2
VRF           : <default>
Host          : www.appexample2.com
Age[min]      : 2
TTL[min]      : 60

```



```

Time To Expire[min] : 58
TXT Record         : app-name:appexample2|app-class:VO|business:YES
Traffic Class      : voip-telephony
Business Relevance : business relevant
IP                 : 192.51.100.11

<output truncated>

Switch# show flow exporter option application engines
Engine: prot (IANA_L3_STANDARD, ID: 1)
Engine: port (IANA_L4_STANDARD, ID: 3)
Engine: NBAR (NBAR_CUSTOM, ID: 6)
Engine: cisco (CISCO_L7_GLOBAL, ID: 13)

Switch# show flow exporter option application table

Engine: prot (IANA_L3_STANDARD, ID: 1)
appID  Name          Description
-----
Engine: port (IANA_L4_STANDARD, ID: 3)
appID  Name          Description
-----

Engine: NBAR (NBAR_CUSTOM, ID: 6)
appID  Name          Description
-----
6:28202appexample1 User defined protocol dns-as-www

Engine: cisco (CISCO_L7_GLOBAL, ID: 13)
appID  Name          Description
-----
13:0   unclassified Unclassified traffic
13:1   unknown       Unknown application
13:518 appexample2     appexample2, social web application and service

```

Monitoring AVC with DNS-AS

To display the various AVC with DNS-AS settings you have configured, use these **show** commands in the privileged EXEC mode:

Table 45-2 AVC with DNS-AS Monitoring Commands

Command	Purpose	Example
show avc dns-as client status	Displays current status of the DNS-AS client—whether AVC with DNS-AS is enabled or not.	Example: show avc dns-as client status
show avc dns-as client trusted-domains	Displays list of trusted domains configured.	Example: show avc dns-as client trusted-domains
show avc dns-as client binding-table and show avc dns-as client binding-table detail	Displays AVC with DNS-AS metadata for the list of trusted domains and resolved entries. You can filter the output by application name, domain name, and so on. Both commands display the same information, in different formats.	Example: show avc dns-as client binding-table detail

Table 45-2 AVC with DNS-AS Monitoring Commands

Command	Purpose	Example
show avc dns-as client statistics	Displays packet logging information—the number of DNS queries sent and the number of responses received.	Example: show avc dns-as client statistics
show avc dns-as client name-server brief	Displays information about the DNS server to which the metadata request was sent.	Example: show avc dns-as client name-server brief
show ip name-server	Displays all the name server IP addresses that have been maintained	Example: show ip name-server

Example: show avc dns-as client status

```
Switch# show avc dns-as client status
DNS-AS client is enabled
```

Back to [Table 45-2](#).

Example: show avc dns-as client trusted-domains

```
Switch #show avc dns-as client trusted-domains
Id | Trusted domain
-----
1| example.com
2| www.example.com
3| example.net
4| www.example.net
5| example.org
```

Back to [Table 45-2](#).

Example: show avc dns-as client binding-table detail

```
Switch# show avc dns-as client binding-table detailed
DNS-AS generated protocols:
Max number of protocols      :50
Customization interval [min] :N/A

Age          : The amount of time that the entry is active
TTL          : Time to live which was learned from DNS-AS server
Time To Expire : Entry expiration time in case device does not see DNS traffic for the
entry host

Protocol-Name : example
VRF           : <default>
Host          : www.example.com
Age[min]      : 2
TTL[min]      : 60
Time To Expire[min] : 58
TXT Record    : app-name:example|app-class:VO|business:YES
Traffic Class : voip-telephony
Business Relevance : business relevant
IP            : 192.0.2.121
              : 192.0.2.254
              : 198.51.100.1
              : 198.51.100.254
              : 192.51.100.12
              : 203.0.113.125

<output truncated>
```

Back to [Table 45-2](#).

Example: show avc dns-as client statistics



Note Two DNS servers are configured in this example.

```
Switch# show avc dns-as client statistics
```

```

Server details: vrf-id = 0 vrf-name = <default> ip = 192.0.2.1
AAAA Query      Error packets 0
AAAA Query      TX      packets 0
AAAA Response RX  packets 0
TXT  Query      Error packets 0
TXT  Query      TX      packets 8
TXT  Response RX  packets 0
A    Query      Error packets 0
A    Query      TX      packets 6
A    Response RX  packets 0
Server details: vrf-id = 0 vrf-name = <default> ip = 192.0.2.2
AAAA Query      Error packets 0
AAAA Query      TX      packets 0
AAAA Response RX  packets 0
TXT  Query      Error packets 0
TXT  Query      TX      packets 2
TXT  Response RX  packets 2
A    Query      Error packets 0
A    Query      TX      packets 4
A    Response RX  packets 2
Total Drop      packets 0

avc_dns_as_pkts_logged      = 2
avc_dns_as_q_pkts_processed = 2

```

Back to [Table 45-2](#).

Example: show avc dns-as client name-server brief

```

Switch# show avc dns-as client name-server brief
Server-IP | Vrf-name
-----
192.0.2.1 | <default>
192.0.2.2 | <default>

```

Back to [Table 45-2](#).

Example: show ip name-server

```

Switch# show ip name-server
192.0.2.1
192.0.2.2
2001:DB8::1

```

Back to [Table 45-2](#).

Troubleshooting AVC with DNS-AS

Problem	Possible Causes and Solutions
There are no entries in the binding table	<p>The binding table may be empty because of one or both of these reasons:</p> <ul style="list-style-type: none"> • Metadata is not maintained in DNS server—complete task Generating Metadata Streams, page 45-8 • The entry is not maintained in the trusted domain list—complete task Making an Entry in the Trusted Domain List, page 45-11
Unsuccessful DNS snooping or packet logging.	To ensure DNS snooping and packet logging, you must attach the policy map (containing the relevant class maps that will determine traffic class) to the interface—See the example in the Configuring QoS for AVC with DNS-AS section.
The DNS server does not return correct values	<p>Verify that the correct DNS-AS metadata is maintained in the DNS system</p> <ul style="list-style-type: none"> • Using Linux dig: <pre>dig TXT +short www.example.org [dns-server-ip] "CISCO-CLS=app-name:example app-class:TD business:YES app-id:CU/28 202"</pre> • Using Windows nslookup: <pre>C:\Windows\system32>NSLookup.exe -q=TXT www.example.org [dns-server-ip] www.example.org text = "CISCO-CLS=app-name:example app-class:TD business:YES app-id:CU/28 202"</pre>
The QoS policy you applied to the port is removed.	<p>When the DNS-AS client recognises an application, along with saving the "A" record response in the binding table, the system utilises the TCAM to save the IP address of the application. A single application can in effect have multiple IP addresses, each utilising additional space in the TCAM. When the TCAM is exhausted, QoS policies cease to be applied.</p> <p>To avoid the problem, monitor TCAM utilisation on a regular basis. Enter the show platform tcam utilisation command in privilege EXEC mode, to display information about TCAM availability.</p>



Configuring Voice Interfaces

This chapter describes how to configure voice interfaces for the Catalyst 4500 series switches.

This chapter includes the following major sections:

- [About Voice Interfaces, page 46-1](#)
- [Configuring a Port to Connect to a Cisco 7960 IP Phone, page 46-2](#)
- [Configuring Voice Ports for Voice and Data Traffic, page 46-3](#)
- [Overriding the CoS Priority of Incoming Frames, page 46-4](#)
- [Configuring Power, page 46-5](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About Voice Interfaces

Catalyst 4500 series switches can connect to a Cisco 7960 IP Phone and carry IP voice traffic. If necessary, the switch can supply electrical power to the circuit connecting it to the Cisco 7960 IP Phone.

Because the sound quality of an IP telephone call can deteriorate if the data is unevenly sent, the switch uses quality of service (QoS) based on IEEE 802.1p class of service (CoS). QoS uses classification and scheduling to transmit network traffic from the switch in a predictable manner. See [Chapter 44, “Configuring Quality of Service,”](#) for more information on QoS.

You can configure the Cisco 7960 IP Phone to forward traffic with an 802.1p priority. You can use the CLI to configure a Catalyst 4500 series switch to honor or ignore a traffic priority assigned by a Cisco 7960 IP Phone.

The Cisco 7960 IP Phone contains an integrated three-port 10/100 switch. The ports are dedicated connections as described below:

- Port 1 connects to the Catalyst 4500 series switch or other device that supports voice-over-IP.
- Port 2 is an internal 10/100 interface that carries the phone traffic.
- Port 3 connects to a PC or other device.

[Figure 46-1](#) shows one way to configure a Cisco 7960 IP Phone.

Figure 46-1 Cisco 7960 IP Phone Connected to a Catalyst 4500 Series Switch

Cisco IP Phone Voice Traffic

You can configure an access port with an attached Cisco IP phone to use one VLAN for voice traffic and another VLAN for data traffic from a device attached to the phone. You can configure access ports on the switch to send Cisco Discovery Protocol (CDP) packets that instruct an attached phone to send voice traffic to the switch in any of these ways:

- In the voice VLAN tagged with a Layer 2 CoS priority value
- In the access VLAN tagged with a Layer 2 CoS priority value
- In the access VLAN, untagged (no Layer 2 CoS priority value)

**Note**

In all configurations, the voice traffic carries a Layer 3 IP precedence value (the default is 5 for voice traffic and 3 for voice control traffic).

Cisco IP Phone Data Traffic

The switch can also process tagged data traffic (traffic in IEEE 802.1Q or IEEE 802.1p frame types) from the device attached to the access port on the Cisco IP phone. See [Figure 46-1](#). You can configure Layer 2 access ports on the switch to send CDP packets that instruct the attached phone to configure the phone access port in one of these modes:

- In trusted mode, all traffic received using the access port on the Cisco IP phone passes using the phone unchanged.
- In untrusted mode, all traffic in IEEE 802.1Q or IEEE 802.1p frames received using the access port on the Cisco IP phone receive a configured Layer 2 CoS value. The default Layer 2 CoS value is 0. Untrusted mode is the default.

**Note**

Untagged traffic from the device attached to the Cisco IP phone passes using the phone unchanged, regardless of the trust state of the access port on the phone.

Configuring a Port to Connect to a Cisco 7960 IP Phone

Because a Cisco 7960 IP Phone also supports connection to a PC or another device, an interface connecting a Catalyst 4500 series switch to a Cisco 7960 IP Phone can carry a mix of voice and data traffic.

The three configurations for a port connected to a Cisco 7960 IP Phone are as follows:

- All traffic is transmitted according to the default CoS priority of the port. it is the default.

- Voice traffic is given a higher priority by the phone (CoS priority is always 5), and all traffic is in the same VLAN.
- Voice and data traffic are carried on separate VLANs.

To configure a port to instruct the phone to give voice traffic a higher priority and to forward all traffic using the 802.1Q native VLAN, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters configuration mode.
Step 2	Switch(config)# interface { fastethernet gigabitethernet } <i>slot/port</i>	Specifies the interface to configure.
Step 3	Switch(config-if)# switchport voice vlan dot1p	Instructs the switch to use 802.1p priority tagging for voice traffic and to use VLAN 1 (default native VLAN) to carry all traffic.
Step 4	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 5	Switch# show interface { fastethernet gigabitethernet } <i>slot/port</i> switchport	Verifies the port configuration.

Configuring Voice Ports for Voice and Data Traffic

Because voice and data traffic can travel using the same voice port, you should specify a different VLAN for each type of traffic. You can configure a switch port to forward voice and data traffic on different VLANs.



Note

For information on configuring sticky port security on voice VLANs, see the [“Configuring Port Security on Voice Ports”](#) section on page 55-22.



Note

For information on using 802.1X with voice VLANs, see the [“Using 802.1X with Voice VLAN Ports”](#) section on page 49-21.

To configure a port to receive voice and data traffic from a Cisco IP phone on different VLANs, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters configuration mode.
Step 2	Switch(config)# interface { fastethernet gigabitethernet } <i>slot/port</i>	Specifies the interface to configure.
Step 3	Switch(config-if)# switchport mode access	Configures the interface as an access port. The voice VLAN is active only on access ports.
Step 4	Switch(config-if)# switchport voice vlan <i>vlan_num</i>	Instructs the Cisco IP phone to forward all voice traffic through a specified VLAN. The Cisco IP phone forwards the traffic with an 802.1p priority of 5.
Step 5	Switch(config-if)# switchport access vlan <i>data_vlan_num</i>	Configures the access VLAN (the data VLAN) on the port.

	Command	Purpose
Step 6	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 7	Switch# show interface { fastethernet gigabitethernet } <i>slot/port</i> switchport	Verifies the configuration.

In the following example, VLAN 1 carries data traffic, and VLAN 2 carries voice traffic. In this configuration, you must connect all Cisco IP phones and other voice-related devices to switch ports that belong to VLAN 2.

```
Switch# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface fastEthernet 3/1
Switch(config-if)# switchport mode access
Switch(config-if)# switchport voice vlan 2
Switch(config-if)# switchport access vlan 3
Switch(config-if)# end
Switch# show interfaces fastEthernet 3/1 switchport
Name: Fa3/1
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 3 (VLAN0003)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: 2 (VLAN0002)
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk Native VLAN tagging: enabled
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk private VLANs: none
Operational private-vlan: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL
Unknown unicast blocked: disabled
Unknown multicast blocked: disabled
Appliance trust: none
Switch#
```

Overriding the CoS Priority of Incoming Frames

A PC or another data device can connect to a Cisco 7960 IP Phone port. The PC can generate packets with an assigned CoS value. You can also use the switch CLI to override the priority of frames arriving on the phone port from connected devices, and you can set the phone port to accept (trust) the priority of frames arriving on the port.

To override the CoS priority setting received from the non-voice port on the Cisco 7960 IP Phone, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters configuration mode.
Step 2	Switch(config)# interface { fastethernet gigabitethernet } <i>slot/port</i>	Specifies the interface to configure.
Step 3	Switch(config-if)# [no] qos trust extend cos 3	Sets the phone port to override the priority received from the PC or the attached device and forward the received data with a priority of 3. Use the no keyword to return the port to its default setting.
Step 4	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 5	Switch# show interface { fastethernet gigabitethernet } <i>slot/port</i> switchport	Verifies the change.

Configuring Power

The Catalyst 4500 series switch recognizes that it is connected to a Cisco 7960 IP Phone. The Catalyst 4500 series switch can supply Power over Ethernet (PoE) to the Cisco 7960 IP Phone if there is no power on the circuit. The Cisco 7960 IP Phone can also be connected to an AC power source and supply its own power to the voice circuit. If there is power on the circuit, the switch does not supply it.

You can configure the switch not to supply power to the Cisco 7960 IP Phone and to disable the detection mechanism. For information on the CLI commands that you can use to supply PoE to a Cisco 7960 IP Phone, see [Chapter 15, “Configuring Power over Ethernet.”](#)



Configuring Private VLANs

This chapter describes how to implement private VLANs (PVLANS) on Catalyst 4500 series switches. It also provides restrictions, procedures, and configuration examples.

This chapter includes the following major sections:

- [About Private VLANs, page 47-1](#)
- [PVLAN Commands, page 47-10](#)
- [Configuring PVLANS, page 47-11](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About Private VLANs

The private VLAN (PVLAN) feature addresses two problems that service providers face when using VLANs:

- The switch supports up to 4094 active VLANs. If a service provider assigns one VLAN per customer, this limits the numbers of customers the service provider can support.
- To enable IP routing, each VLAN is assigned a subnet address space or a block of addresses, which can result in wasting the unused IP addresses, and cause IP address management problems.

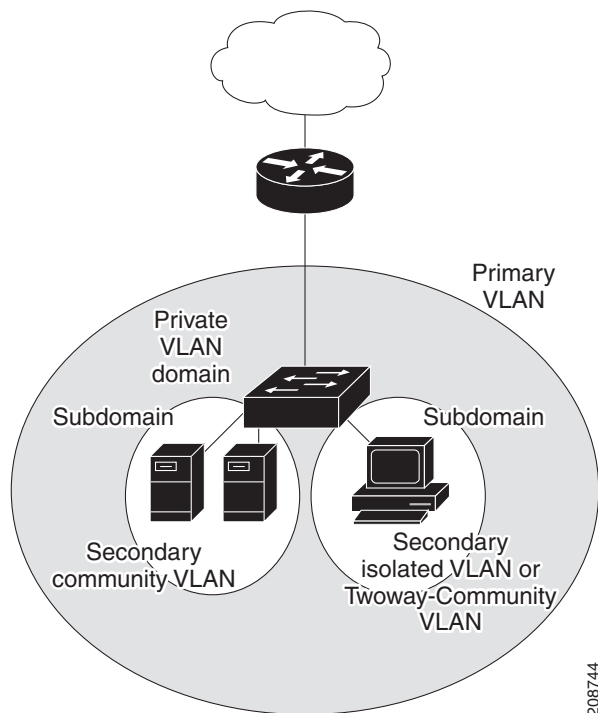
To configure PVLANS, you need to understand the concepts in these sections:

- [Purpose of a PVLAN, page 47-2](#)
- [PVLAN Terminology, page 47-3](#)
- [PVLANS across Multiple Switches, page 47-5](#)
- [PVLAN Modes Over Gigabit Etherchannel, page 47-8](#)
- [Private-VLAN Interaction with Other Features, page 47-8](#)

Purpose of a PVLAN

Using PVLANs provides scalability and IP address management benefits for service providers and Layer 2 security for customers. PVLANs partition a regular VLAN domain into subdomains. A subdomain is represented by a pair of VLANs: a *primary* VLAN and a *secondary* VLAN. A PVLAN can have multiple VLAN pairs, one pair for each subdomain. All VLAN pairs in a PVLAN share the same primary VLAN. The secondary VLAN ID differentiates one subdomain from another. See [Figure 47-1](#).

Figure 47-1 Private-VLAN Domain



The three types of secondary VLANs are as follows:

- Isolated VLANs—Ports within an isolated VLAN cannot communicate with each other at the Layer 2 level.
- Community VLANs—Ports within a community VLAN can communicate with each other but cannot communicate with ports in other communities at the Layer 2 level.
- Twoway-Community VLANs—Bidirectional VLAN. Ports within a twoway-community VLAN can communicate with each other but not with communities or twoway-communities at the Layer 2 level.



Note

Beginning with Cisco IOS Release 15.0(2)SG, you can use a twoway-community VLAN to apply VACLs or QoS in both directions per-community and per-customer.

A promiscuous port can serve only one primary VLAN, one isolated VLAN, and multiple community (or twoway-community) VLANs. Layer 3 gateways are typically connected to the switch through a promiscuous port.

In a switched environment, you can assign an individual PVLAN and associated IP subnet to each individual or common group of end stations. The end stations need to communicate only with a default gateway to communicate outside the PVLAN.

You can use PVLANs to control access to end stations in these ways:

- Configure selected interfaces connected to end stations as isolated ports to prevent any communication at Layer 2. For example, if the end stations are servers, this configuration prevents Layer 2 communication between the servers.
- Configure interfaces connected to default gateways and selected end stations (such as, backup servers) as promiscuous ports to allow all end stations access to a default gateway.
- Reduce VLAN and IP subnet consumption; you can prevent traffic between end stations even though they are in the same VLAN and IP subnet.

With a promiscuous port, you can connect a wide range of devices as access points to a PVLAN. For example, you can connect a promiscuous port to the server port of a LocalDirector to connect an isolated VLAN or a number of community (or twoway-community) VLANs to the server. LocalDirector can load balance the servers present in the isolated, community, or twoway-community VLANs, or you can use a promiscuous port to monitor or back up all the PVLAN servers from an administration workstation.

PVLAN Terminology

The following table defines the key terms used in this chapter:

Term	Definition
PVLANs	PVLANs are sets of VLAN pairs that share a common primary identifier and provide a mechanism for achieving layer-2 separation between ports while sharing a single layer-3 router port and IP subnet.
Secondary VLAN	A type of VLAN used to implement PVLANs. Secondary VLANs are associated with a primary VLAN, and are used to carry traffic from hosts to other allowed hosts or to routers.
Community Port	A community port is a host port that belongs to a community secondary VLAN. Community ports communicate with other ports in the same community VLAN and with promiscuous ports. These interfaces are isolated at Layer 2 from all other interfaces in other communities and from isolated ports within their PVLAN.
Community VLAN	Community VLAN—A community VLAN is a secondary VLAN that carries upstream traffic from the community ports to the promiscuous port gateways and to other host ports in the same community. You can configure multiple community VLANs in a PVLAN.

Term	Definition
Isolated Port	An isolated port is a host port that belongs to an isolated secondary VLAN. It has complete Layer 2 separation from other ports within the same PVLAN, except for the promiscuous ports. PVLANS block all traffic to isolated ports except traffic from promiscuous ports. Traffic received from an isolated port is forwarded only to promiscuous ports.
Isolated VLAN	Isolated VLAN —A PVLAN has only one isolated VLAN. An isolated VLAN is a secondary VLAN that carries unidirectional traffic upstream from the hosts toward the promiscuous ports and the gateway.
Primary VLAN	Primary VLAN—A PVLAN has only one primary VLAN. Every port in a PVLAN is a member of the primary VLAN. The primary VLAN carries unidirectional traffic downstream from the promiscuous ports to the (isolated and community) host ports and to other promiscuous ports.
PVLAN Trunk Port	A PVLAN trunk port can carry multiple secondary (isolated only) and non-PVLANS. Packets are received and transmitted with secondary or regular VLAN tags on the PVLAN trunk ports. Note Only IEEE 802.1q encapsulation is supported.
Promiscuous Port	A promiscuous port belongs to the primary VLAN and can communicate with all interfaces, including the community and isolated host ports and PVLAN trunk ports that belong to the secondary VLANs associated with the primary VLAN.
Promiscuous Trunk Port	A promiscuous trunk port can carry multiple primary and normal VLANs. Packets are received and transmitted with primary or regular VLAN tags. Other than that, the port behaves just like a promiscuous access port. Note Only IEEE 802.1q encapsulation is supported.
Two-way-Community Ports	A two-way-community port is a host port that belongs to a two-way-community secondary VLAN. Ports within a two-way-community VLAN can communicate with each other but not with ports in other communities or two-way-communities at the Layer 2 level. These interfaces are isolated at Layer 2 from all other interfaces in other two-way communities and from isolated ports within their PVLAN.
Two-way-Community VLANs	A bidirectional VLAN. Ports within a 2-way community VLAN can communicate with each other but cannot communicate with ports in other 2-way communities at the Layer 2 level.

PVLANS across Multiple Switches

This section discusses the following topics:

- [Standard Trunk Ports, page 47-5](#)
- [Isolated PVLAN Trunk Ports, page 47-6](#)
- [Promiscuous PVLAN Trunk Ports, page 47-7](#)

Standard Trunk Ports

As with regular VLANs, PVLANS can span multiple switches. A trunk port carries the primary VLAN and secondary VLANs to a neighboring switch. The trunk port treats the PVLAN as any other VLAN. A feature of PVLANS across multiple switches is that traffic from an isolated port in switch A does not reach an isolated port on Switch B. See [Figure 47-2](#).

To maintain the security of your private-VLAN configuration and to avoid other use of the VLANs configured as PVLANS, configure PVLANS on all intermediate devices, including devices that have no private-VLAN ports.



Note

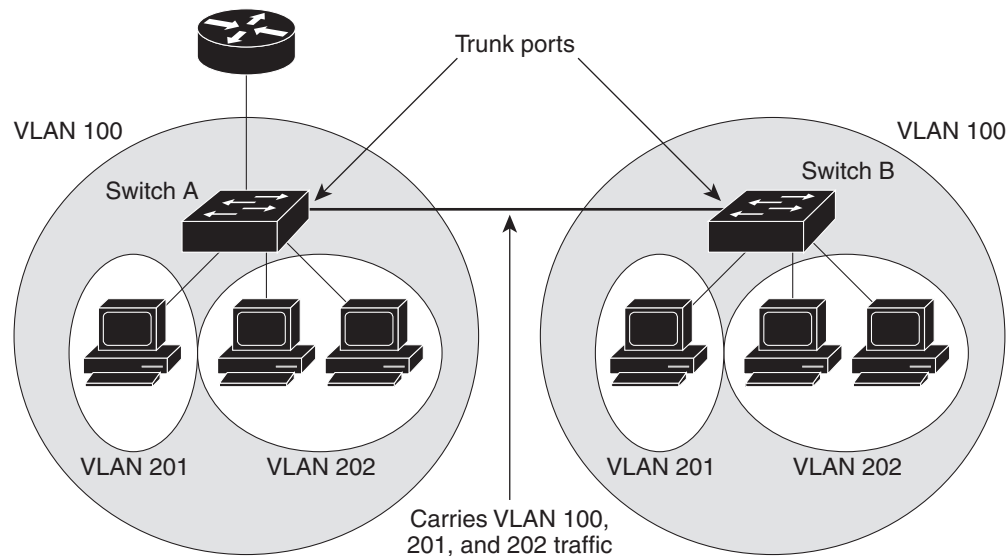
Trunk ports carry traffic from regular VLANs and also from primary, isolated, community or twoway community VLANs.



Note

You should use standard trunk ports if both switches undergoing trunking support PVLANS.

Figure 47-2 PVLANS across Switches



VLAN 100 = Primary VLAN
 VLAN 201 = Secondary isolated VLAN
 VLAN 202 = Secondary community or Twoway-community VLAN

208745

Because VTP does not support PVLANS, you must manually configure PVLANS on all switches in the Layer 2 network. If you do not configure the primary and secondary VLAN association in some switches in the network, the Layer 2 databases in these switches are not merged. This can result in unnecessary flooding of private-VLAN traffic on those switches.

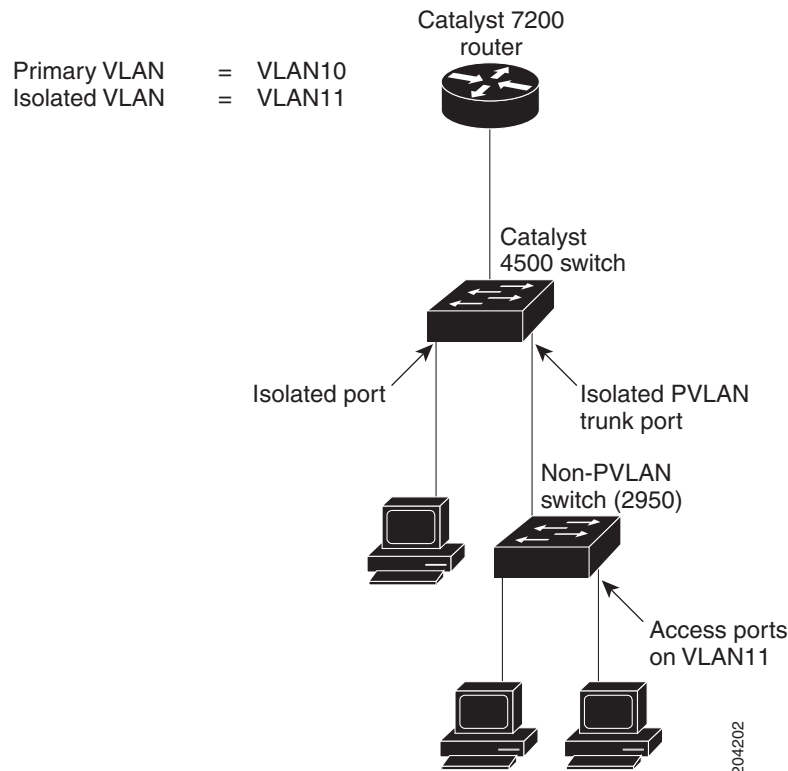
**Note**

PVLANS are supported in VTP v3 under server mode.

Isolated PVLAN Trunk Ports

You would use a isolated PVLAN trunk ports when you would anticipate using PVLAN isolated host ports to carry multiple VLANs, either normal VLANs or for multiple PVLAN domains. This makes it useful for connecting a downstream switch that does not support PVLANS such as Catalyst 2950.

Figure 47-3 **Isolated PVLAN Trunk Ports**



In this illustration, a Catalyst 4500 switch is being used to connect a downstream switch that does not support PVLANS.

Traffic being sent in the downstream direction towards host1 from the router is received by the Catalyst 4500 series switch on the promiscuous port and in the primary VLAN (VLAN 10). The packets are then switched out of the isolated PVLAN trunk. Rather than being tagged with the primary VLAN (VLAN 10), they are transmitted with the isolated VLAN's tag (VLAN 11). In this way, when the packets arrive on the non-PVLAN switch, they can be bridged to the destination hosts' access port.

Traffic in the upstream direction is sent by host1 to the non-PVLAN switch, arriving in VLAN 11. The packets are then transmitted to the switch tagged with that VLAN's tag (VLAN 11) over the trunk port. On the switch, VLAN 11 is configured as the isolated VLAN, and the traffic is forwarded as if it came from an isolated host port.

**Note**

When an isolated trunk is used in this way, Catalyst 4500 series switch provides isolation between the isolated trunk and directly connected hosts (such as host3) but not between hosts connected to the non-PVLAN switch (such as host1 and host2). The non-PVLAN switch must provide isolation between these hosts, using a feature such as protected ports on a Catalyst 2950.

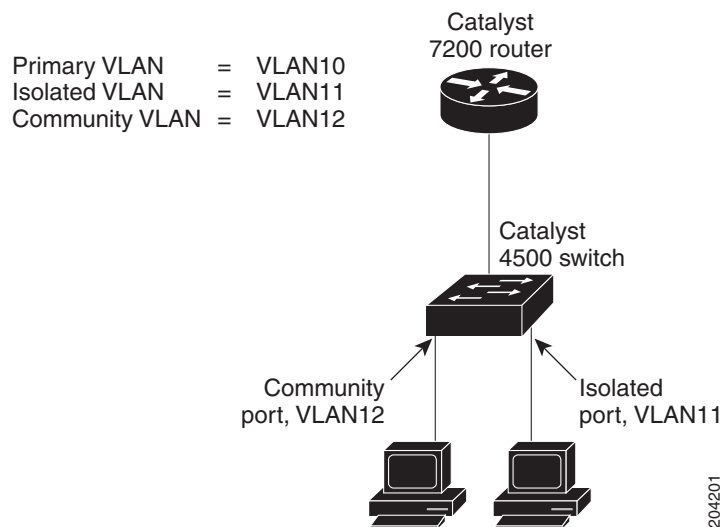
For details on protected ports, see the URL:

http://www.cisco.com/en/US/docs/switches/lan/catalyst2950/software/release/12.1_22_ea11x/configuration/guide/swtrafc.html#wp1158863

Promiscuous PVLAN Trunk Ports

PVLAN promiscuous trunks are used in situations where one would normally use a PVLAN promiscuous host port but where it is necessary to carry multiple VLANs, either normal VLANs or for multiple PVLAN domains. This makes it useful for connecting an upstream router that does not support PVLANs, such as a Cisco 7200.

Figure 47-4 Promiscuous PVLAN Trunk Ports



In [Figure 47-4](#), a Catalyst 4500 series switch connects a PVLAN domain to an upstream router that does not support PVLANs. Traffic being sent upstream by host1 arrives on the switch in the community VLAN (VLAN 12). When this traffic is bridged onto the promiscuous PVLAN trunk towards the router, it is tagged with the primary VLAN (VLAN 10). This way it can be routed using the correct subinterface configured on the router.

Traffic in the downstream direction is received on the promiscuous PVLAN trunk port by the switch in the primary VLAN (VLAN 10), just as if it had been received on a promiscuous host port. It can then be bridged to the destination host as in any PVLAN domain.

PVLAN promiscuous trunks interact with VLAN QoS. Refer to the section [“PVLANs and VLAN ACL/QoS”](#) section on [page 47-8](#).

PVLAN Modes Over Gigabit Etherchannel

Beginning with Cisco IOS Release 15.0(2)SG you can configure PVLAN modes over Etherchannel. These new modes are:

- Host mode - Isolated, Community and 2-way community
- Promiscuous mode
- Secondary Isolated trunks
- Promiscuous trunks

The process of bundling ports has not changed. PVLAN modes are added to already existing modes such as access, trunk, routed, tunneled etc.

Feature interactions include:

- A primary VLAN can be associated with multiple community and twoway-community VLANs, but only one isolated VLAN.
- An isolated or community VLAN or 2-way community VLAN can be associated with only one primary VLAN.
- If you delete a VLAN used in a PVLAN configuration, the PVLAN ports associated with the VLAN become inactive.
- The default native VLAN for promiscuous trunk port is VLAN 1 (management VLAN). All untagged packets are forwarded in the native VLAN. Either the primary VLANs or a regular VLAN can be configured as the native VLAN.
- No default native VLAN set exists on an isolated secondary trunks. All untagged packets are dropped, if no native VLAN is configured.
- Community and twoway-community VLANs cannot be propagated or carried over PVLAN trunks.
- For IGMP Snooping, IGMP reports are learned on the primary VLAN and the platform decides if packet must be forwarded in the primary or secondary VLANs.

For details on configuring PVLANs over EtherChannel, Refer to the section [“Configuring PVLAN over EtherChannel”](#) section on page 47-24.

Private-VLAN Interaction with Other Features

PVLANs have specific interaction with some other features, described in these sections:

- [PVLANs and VLAN ACL/QoS, page 47-8](#)
- [PVLANs and Unicast, Broadcast, and Multicast Traffic, page 47-9](#)
- [PVLANs and SVIs, page 47-10](#)
- [Per-Virtual Port Error-Disable on PVLANs, page 47-10](#)

For details, see the section “PVLAN Configuration Guidelines and Restrictions” on page 12.

PVLANs and VLAN ACL/QoS

PVLAN ports use primary and secondary VLANs, as follows:

- A packet received on a PVLAN host port belongs to the secondary VLAN.

- A packet received on a PVLAN trunk port belongs to the secondary VLAN if the packet is tagged with a secondary VLAN or if the packet is untagged and the native VLAN on the port is a secondary VLAN.

A packet received on a PVLAN host or trunk port and assigned to a secondary VLAN is bridged on the secondary VLAN. Because of this bridging, the secondary VLAN ACL as well as the secondary VLAN QoS (on input direction) apply.

When a packet is transmitted out of a PVLAN host or trunk port, the packet logically belongs to the primary VLAN. This relationship applies even though the packet may be transmitted with the secondary VLAN tagging for PVLAN trunk ports. In this situation, the primary VLAN ACL and the primary VLAN QoS on output apply to the packet.

- Similarly, a packet received on a PVLAN promiscuous access port belongs to primary VLAN.
- A packet received on a PVLAN promiscuous trunk port could belong to the primary VLAN or normal VLAN depending on incoming VLAN.

For traffic flowing in normal VLAN on promiscuous trunk ports, normal VLAN ACL and QoS policies apply. For traffic flowing in a PVLAN domain, a packet received on a promiscuous port is bridged in primary VLAN. The primary VLAN ACL and QoS policies apply on input.

For egress traffic on twoway-community host port, the secondary VLAN ACL and secondary VLAN QoS apply to egress unicast routed traffic stemming from the integrated router port.

When a packet is transmitted out of a promiscuous trunk port, the packet could logically belong to secondary VLAN if received from a secondary port, or in primary VLAN if bridged from another promiscuous port. Because we cannot differentiate between both packets, all VLAN QoS policies are ignored on packets egressing promiscuous trunk ports.

PVLANS and Unicast, Broadcast, and Multicast Traffic

In regular VLANs, devices in the same VLAN can communicate with each other at the Layer 2 level, but devices connected to interfaces in different VLANs must communicate at the Layer 3 level. In PVLANS, the promiscuous ports are members of the primary VLAN, while the host ports belong to secondary VLANs. Because the secondary VLAN is associated to the primary VLAN, members of these VLANs can communicate with each other at the Layer 2 level.

In a regular VLAN, broadcasts are forwarded to all ports in that VLAN. PVLAN broadcast forwarding depends on the port sending the broadcast:

- An isolated port sends a broadcast only to the promiscuous ports or trunk ports.
- A community port sends a broadcast to all promiscuous ports, trunk ports, and ports in the same community VLAN.
- A promiscuous port sends a broadcast to all ports in the PVLAN (other promiscuous ports, trunk ports, isolated ports, and community ports).

Multicast traffic is routed or bridged across private-VLAN boundaries and within a single community VLAN. Multicast traffic is not forwarded between ports in the same isolated VLAN or between ports in different secondary VLANs.

Transmitting Multicast traffic through secondary private VLAN where the multicast source is located is not supported. This might result in software switching of the packets and thereby causing significantly lower forwarding rate and packet loss.

PVLANS and SVIs

In a Layer 3 switch, a switch virtual interface (SVI) represents the Layer 3 interface of a VLAN. Layer 3 devices communicate with a PVLAN only using the primary VLAN and not through secondary VLANs. Configure Layer 3 VLAN interfaces (SVIs) only for primary VLANs. You cannot configure Layer 3 VLAN interfaces for secondary VLANs. SVIs for secondary VLANs are inactive while the VLAN is configured as a secondary VLAN.

- If you try to configure a VLAN with an active SVI as a secondary VLAN, the configuration is not allowed until you disable the SVI.
- If you try to create an SVI on a VLAN that is configured as a secondary VLAN and the secondary VLAN is already mapped at Layer 3, the SVI is not created, and an error is returned. If the SVI is not mapped at Layer 3, the SVI is created, but it is automatically shut down.

When the primary VLAN is associated with and mapped to the secondary VLAN, any configuration on the primary VLAN is propagated to the secondary VLAN SVIs. For example, if you assign an IP subnet to the primary VLAN SVI, this subnet is the IP subnet address of the entire PVLAN.

Per-Virtual Port Error-Disable on PVLANS

For PVLANS, per-virtual port error-disable behavior is defined as follows:

- On a PVLAN promiscuous or promiscuous trunk ports, if a violation occurs on the primary VLAN, it is error-disabled.
- On a PVLAN host or trunk port, if a violation occurs on the secondary VLAN, the associated primary VLAN is error-disabled.
- On a standard trunk port that carries both primary and secondary VLANs, if a violation occurs on the primary VLAN, this VLAN and all its associated secondary VLANs are error-disabled. If a violation occurs on a secondary VLAN, the associated primary VLAN and all its associated secondary VLANs are error-disabled.

PVLAN Commands

This table lists the commands most commonly used with PVLANS.

Command	Purpose	Location
private-vlan { community twoway-community isolated primary }	Configures a VLAN as a PVLAN.	Configuring a VLAN as a PVLAN, page 47-15
private-vlan association { <i>secondary_vlan_list</i> add <i>secondary_vlan_list</i> remove <i>secondary_vlan_list</i> }	Associates the secondary VLAN with the primary VLAN. The list can contain only one isolated VLAN ID; it can also contain multiple community VLAN IDs.	Associating a Secondary VLAN with a Primary VLAN, page 47-16
show vlan private-vlan [type]	Verifies the configuration.	Configuring a VLAN as a PVLAN, page 47-15 Associating a Secondary VLAN with a Primary VLAN, page 47-16

Command	Purpose	Location
show interface private-vlan mapping	Verifies the configuration.	Permitting Routing of Secondary VLAN Ingress Traffic, page 47-23
switchport mode private-vlan {host promiscuous trunk promiscuous trunk [secondary]}	Configures a Layer 2 interface as a PVLAN port.	Configuring PVLANS, page 47-11
switchport private-vlan mapping [trunk] <i>primary_vlan_ID</i> { <i>secondary_vlan_list</i> add <i>secondary_vlan_list</i> remove <i>secondary_vlan_list</i> }	Maps the PVLAN promiscuous port to a primary VLAN and to selected secondary VLANs.	Configuring a Layer 2 Interface as a PVLAN Promiscuous Port, page 47-17 Configuring a Layer 2 Interface as a Promiscuous PVLAN Trunk Port, page 47-21
Switch(config-if)# switchport private-vlan host-association <i>primary_vlan_ID secondary_vlan_ID</i>	Associates the Layer 2 interface with a PVLAN. Note You can associate only one primary-secondary VLAN pair to the isolated port.	Configuring a Layer 2 Interface as a PVLAN Host Port, page 47-18
switchport private-vlan association trunk <i>primary_vlan_ID</i> <i>secondary_vlan_ID</i>	Configures association between primary VLANs and secondary VLANs the PVLAN trunk port with a PVLAN. Note You can configure the isolated trunk port with multiple primary-secondary pair.	Configuring a Layer 2 Interface as an Isolated PVLAN Trunk Port, page 47-19
switchport private-vlan trunk allowed vlan <i>vlan_list</i> all none [add remove except] <i>vlan_atom</i> [, <i>vlan_atom</i> ...]	Configures a list of allowed normal VLANs on a PVLAN trunk port.	Configuring a Layer 2 Interface as an Isolated PVLAN Trunk Port, page 47-19
switchport private-vlan trunk native vlan <i>vlan_id</i>	Configures a VLAN to which untagged packets (as in IEEE 802.1Q tagging) are assigned on a PVLAN trunk port.	Configuring a Layer 2 Interface as an Isolated PVLAN Trunk Port, page 47-19

Configuring PVLANS

These sections describe how to configure PVLANS:

- [Basic PVLAN Configuration Procedure, page 47-12](#)
- [Default Private-VLAN Configuration, page 47-12](#)
- [PVLAN Configuration Guidelines and Restrictions, page 47-12](#)
- [Configuring a VLAN as a PVLAN, page 47-15](#)
- [Associating a Secondary VLAN with a Primary VLAN, page 47-16](#)
- [Configuring a Layer 2 Interface as a PVLAN Promiscuous Port, page 47-17](#)
- [Configuring a Layer 2 Interface as a PVLAN Host Port, page 47-18](#)

- [Configuring a Layer 2 Interface as an Isolated PVLAN Trunk Port](#), page 47-19
- [Configuring a Layer 2 Interface as a Promiscuous PVLAN Trunk Port](#), page 47-21
- [Permitting Routing of Secondary VLAN Ingress Traffic](#), page 47-23
- [Configuring PVLAN over EtherChannel](#), page 47-24

Basic PVLAN Configuration Procedure

To configure a PVLAN, follow these basic steps:

-
- Step 1** Set VTP mode to transparent. See the [“VLAN Trunking Protocol”](#) section on page 17-7.
 - Step 2** Create the secondary VLANs. See the [“Configuring a VLAN as a PVLAN”](#) section on page 47-15.
 - Step 3** Create the primary VLAN. See the [“Configuring a VLAN as a PVLAN”](#) section on page 47-15.
 - Step 4** Associate the secondary VLAN to the primary VLAN. See the [“Associating a Secondary VLAN with a Primary VLAN”](#) section on page 47-16.



Note Only one isolated VLAN can be mapped to a primary VLAN, but more than one community (or twoway-community) VLAN can be mapped to a primary VLAN.

- Step 5** Configure an interface as an isolated or community host or trunk port. See the [“Configuring a Layer 2 Interface as a PVLAN Host Port”](#) section on page 47-18 and [“Configuring a Layer 2 Interface as an Isolated PVLAN Trunk Port”](#) section on page 47-19.
 - Step 6** Associate the isolated port or community port to the primary-secondary VLAN pair. See the [“Associating a Secondary VLAN with a Primary VLAN”](#) section on page 47-16.
 - Step 7** Configure an interface as a promiscuous port. See the [“Configuring a Layer 2 Interface as a PVLAN Promiscuous Port”](#) section on page 47-17.
 - Step 8** Map the promiscuous port to the primary-secondary VLAN pair. See the [“Configuring a Layer 2 Interface as a PVLAN Promiscuous Port”](#) section on page 47-17.
 - Step 9** If you plan to use inter-VLAN routing, configure the primary SVI, and map secondary VLANs to the primary. See the [“Permitting Routing of Secondary VLAN Ingress Traffic”](#) section on page 47-23.
 - Step 10** Verify private-VLAN configuration. See the [“Switch#”](#) section on page 47-24.
-

Default Private-VLAN Configuration

No PVLANS are configured.

PVLAN Configuration Guidelines and Restrictions

When using (or configuring) PVLANS, consider these guidelines and restrictions:

- To configure a PVLAN correctly, enable VTP in transparent mode in VTP version 1 and VTP version 2. (VTP version 3 enables you to create it in server mode).

You cannot change the VTP mode to client or server for PVLANS.

- Do not include VLAN 1 or VLANs 1002 through 1005 in PVLANS.
- Use only PVLAN commands to assign ports to primary, isolated, community VLANs, or twoway-community VLANs.

Layer 2 interfaces on primary, isolated, community VLANs, or twoway-community VLANs are inactive in PVLANS. Layer 2 trunk interfaces remain in the STP forwarding state.
- You cannot configure Layer 3 VLAN interfaces for secondary VLANs.

Layer 3 VLAN interfaces for isolated and community (secondary) VLANs are inactive while the VLAN is configured as an isolated or community VLAN.
- Do not apply dynamic access control entries (ACEs) to primary VLANs.

Cisco IOS dynamic ACL configuration applied to a primary VLAN is inactive while the VLAN is part of the PVLAN configuration.
- To prevent spanning tree loops due to misconfigurations, enable PortFast on the PVLAN trunk ports with the **spanning-tree portfast trunk** command.
- Any VLAN ACL configured on a secondary VLAN is effective in the input direction, and any VLAN ACL configured on the primary VLAN associated with the secondary VLAN is effective in the output direction. Exception case is given below.
- On twoway-community host ports, secondary VLAN ACL and QoS are applied on egress unicast routed traffic stemming from the integrated router port
- You can stop Layer 3 switching on an isolated or community VLAN by deleting the mapping of that VLAN with its primary VLAN.
- PVLAN ports can be on different network devices as long as the devices are trunk-connected and the primary and secondary VLANs remain associated with the trunk
- Isolated ports on two different devices cannot communicate with each other, but community VLAN ports can.
- PVLANS support the following SPAN features:
 - You can configure a PVLAN port as a SPAN source port.
 - To monitor egress or ingress traffic separately, you can use VLAN-based SPAN (VSPAN) on primary, isolated, community VLANs, twoway-community VLANs, or use SPAN on only one VLAN.

For more information about SPAN, see [Chapter 69, “Configuring SPAN and RSPAN.”](#)

- Although a PVLAN contains more than one VLAN, only one Spanning Tree Protocol (STP) instance runs for the entire PVLAN. When a secondary VLAN is associated with the primary VLAN, the STP parameters of the primary VLAN are propagated to the secondary VLAN. A separate STP instance exists for each of the primary and secondary VLANs.
- A primary VLAN can be associated with multiple community VLANs, or twoway-community VLANs, but only one isolated VLAN.
- An isolated or community VLAN can be associated with only one primary VLAN.
- If you delete a VLAN used in a PVLAN configuration, the PVLAN ports associated with the VLAN become inactive.
- VTP does not support PVLANS. You must configure PVLANS on each device in which you plan to use PVLAN ports.
- To maintain the security of your PVLAN configuration and avoid other use of VLANs configured as PVLANS, configure PVLANS on all intermediate devices, even if the devices have no PVLAN ports.

- Prune the PVLANS from trunks on devices that carry no traffic in the PVLANS.
- With port ACLS functionality available, you can apply Cisco IOS ACLs to secondary VLAN ports and Cisco IOS ACLs to PVLANS (VACLs). For more information on VACLs, see [Chapter 62, “Configuring Network Security with ACLs.”](#)
- You can apply different quality of service (QoS) configurations to primary, isolated, community VLANs, and twoway-community VLANs. See [Chapter 44, “Configuring Quality of Service.”](#) Cisco IOS ACLs applied to the Layer 3 VLAN interface of a primary VLAN automatically apply to the associated isolated, community VLANs, and twoway-community VLANs.
- On a PVLAN trunk port a secondary VLAN ACL is applied on ingress traffic and a primary VLAN ACL is applied on egress traffic.
- On a promiscuous port the primary VLAN ACL is applied on ingress traffic.
- Both PVLAN secondary and promiscuous trunk ports support only IEEE 802.1q encapsulation.
- Community VLANs cannot be propagated or carried over PVLAN trunks.
- ARP entries learned on Layer 3 PVLAN interfaces are termed “sticky” ARP entries (we recommend that you display and verify PVLAN interface ARP entries).
- For security reasons, PVLAN port sticky ARP entries do not age out. Connecting a device with a different MAC address but with the same IP address generates an error message and the ARP entry is not created.
- Because PVLAN port sticky ARP entries do not age out, you must manually remove the entries if you change the MAC address. To overwrite a sticky ARP entry, first delete the entry with the **no arp** command, then overwrite the entry with the **arp** command.
- In a DHCP environment, if you shut down your PC, it is not possible to give your IP address to someone else. To solve this problem, the Catalyst 4500 series switch supports the **no ip sticky-arp** command. This command promotes IP address overwriting and reuse in a DHCP environment.
- Normal VLANs can be carried on a promiscuous or isolated trunk port.
- The default native VLAN for promiscuous trunk port is VLAN 1, the management VLAN. All untagged packets are forwarded in the native VLAN. Either the primary VLANs or a regular VLAN can be configured as native VLAN.
- Promiscuous trunks cannot be configured to carry secondary VLANs. If a secondary VLAN is specified in the allowed VLAN list, the configuration is accepted but the port is not operational/forwarding in the secondary VLAN. This includes even those VLANs that are of secondary but not associated with any primary VLAN on given port.
- On a promiscuous trunk port, the primary VLAN ACL and QoS are applied on ingress traffic coming in primary VLANs.
- On a promiscuous trunk port, no VLAN ACL or QoS is applied to the egress traffic. It is because for upstream direction, traffic in PVLAN logically flows in the secondary VLAN. Due to VLAN translation in hardware, information about received secondary VLANs has been lost. No policies are applied. This restriction also applies to traffic bridged from other ports in the same primary VLANs.
- Do not configure port security on PVLAN promiscuous trunk port and vice versa.
If port security is enabled on a promiscuous trunk port, that port may behave in an unpredictable manner because this functionality is not supported.
- Do not configure IEEE 802.1X on a PVLAN promiscuous trunk port.

**Note**

Community or twoway-community PVLAN trunk ports are not supported.

Configuring a VLAN as a PVLAN

To configure a VLAN as a PVLAN, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters configuration mode.
Step 2	Switch(config)# vlan <i>vlan_ID</i>	Enters VLAN configuration mode.
Step 3	Switch(config-vlan)# private-vlan { community twoway-community isolated primary }	Configures a VLAN as a PVLAN. <ul style="list-style-type: none"> This command does not take effect until you exit VLAN configuration submode. You can use the no keyword to clear PVLAN status.
Step 4	Switch(config-vlan)# end	Exits VLAN configuration mode.
Step 5	Switch# show vlan private-vlan [<i>type</i>]	Verifies the configuration.

This example shows how to configure VLAN 202 as a primary VLAN and verify the configuration:

```
Switch# configure terminal
Switch(config)# vlan 202
Switch(config-vlan)# private-vlan primary
Switch(config-vlan)# end
Switch# show vlan private-vlan
Primary Secondary Type Interfaces
-----
202                primary
```

This example shows how to configure VLAN 303 as a community VLAN and verify the configuration:

```
Switch# configure terminal
Switch(config)# vlan 303
Switch(config-vlan)# private-vlan community
Switch(config-vlan)# end
Switch# show vlan private-vlan
Primary Secondary Type Interfaces
-----
202                primary
                303 community
```

This example shows how to configure VLAN 440 as an isolated VLAN and verify the configuration:

```
Switch# configure terminal
Switch(config)# vlan 440
Switch(config-vlan)# private-vlan isolated
Switch(config-vlan)# end
Switch# show vlan private-vlan
Primary Secondary Type Interfaces
-----
202                primary
                303 community
                440 isolated
```

This example shows how to configure VLAN 550 as a twoway-community VLAN and verify the configuration:

```
Switch# configure terminal
Switch(config)# vlan 550
Switch(config-vlan)# private-vlan twoway-community
Switch(config-vlan)# end
Switch# show vlan private-vlan
```

```
Primary Secondary Type Interfaces
-----
202                primary
303             community
440             isolated
550    twoway-community
```

Associating a Secondary VLAN with a Primary VLAN

To associate secondary VLANs with a primary VLAN, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters configuration mode.
Step 2	Switch(config)# vlan <i>primary_vlan_ID</i>	Enters VLAN configuration mode for the primary VLAN.
Step 3	Switch(config-vlan)# private-vlan association { <i>secondary_vlan_list</i> add <i>secondary_vlan_list</i> remove <i>secondary_vlan_list</i> }	Associates the secondary VLAN with the primary VLAN. The list can contain only one isolated VLAN ID; it can also contain multiple community or twoway-community VLAN IDs. You can use the no keyword to clear all secondary associations.
Step 4	Switch(config-vlan)# end	Exits VLAN configuration mode.
Step 5	Switch# show vlan private-vlan [<i>type</i>]	Verifies the configuration.

When you associate secondary VLANs with a primary VLAN, note the following:

- The *secondary_vlan_list* parameter cannot contain spaces. It can contain multiple comma-separated items. Each item can be a single PVLAN ID or a hyphenated range of PVLAN IDs.
- The *secondary_vlan_list* parameter can contain multiple community or twoway-community VLAN IDs.
- The *secondary_vlan_list* parameter can contain only one isolated VLAN ID.
- Enter a *secondary_vlan_list* or use the **add** keyword with a *secondary_vlan_list* to associate secondary VLANs with a primary VLAN.
- Use the **remove** keyword with a *secondary_vlan_list* to clear the association between secondary VLANs and a primary VLAN.
- The command does not take effect until you exit VLAN configuration submode.

This example shows how to associate community VLANs 303 through 307 and 309, twoway-community VLANs 550 through 552, and isolated VLAN 440 with primary VLAN 202 and verify the configuration:

```
Switch# configure terminal
Switch(config)# vlan 202
Switch(config-vlan)# private-vlan association 303-307,309,440
Switch(config-vlan)# end
Switch# show vlan private-vlan
```

Primary	Secondary	Type	Interfaces
202	303	community	
202	304	community	
202	305	community	
202	306	community	
202	307	community	
202	309	community	
202	440	isolated	
202	550	twoway-community	
202	551	twoway-community	
202	552	twoway-community	
	308	community	

**Note**

The secondary VLAN 308 has no associated primary VLAN.

Configuring a Layer 2 Interface as a PVLAN Promiscuous Port

To configure a Layer 2 interface as a PVLAN promiscuous port, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface {fastethernet gigabitethernet tengigabitethernet} slot/port	Specifies the LAN interface to configure.
Step 3	Switch(config-if)# switchport mode private-vlan {host promiscuous trunk promiscuous trunk [secondary]}	Configures a Layer 2 interface as a PVLAN promiscuous port.
Step 4	Switch(config-if)# [no] switchport private-vlan mapping [trunk] primary_vlan_ID {secondary_vlan_list add secondary_vlan_list remove secondary_vlan_list}	Maps the PVLAN promiscuous port to a primary VLAN and to selected secondary VLANs.
Step 5	Switch(config-if)# end	Exits configuration mode.
Step 6	Switch# show interfaces {fastethernet gigabitethernet tengigabitethernet} slot/port switchport	Verifies the configuration.

**Note**

The maximum number of unique PVLAN pairs supported by the **switchport private-vlan mapping** command is 1000.

When you configure a Layer 2 interface as a PVLAN promiscuous port, note the following:

- The *secondary_vlan_list* parameter cannot contain spaces. It can contain multiple comma-separated items. Each item can be a single PVLAN ID or a hyphenated range of PVLAN IDs.
- Enter a *secondary_vlan_list* or use the **add** keyword with a *secondary_vlan_list* to map the secondary VLANs to the PVLAN promiscuous port.

- Use the **remove** keyword with a *secondary_vlan_list* to clear the mapping between secondary VLANs and the PVLAN promiscuous port.

This example shows how to configure interface FastEthernet 5/2 as a PVLAN promiscuous port, map it to a PVLAN, and verify the configuration:

```
Switch# configure terminal
Switch(config)# interface fastethernet 5/2
Switch(config-if)# switchport mode private-vlan promiscuous
Switch(config-if)# switchport private-vlan mapping 200 2
Switch(config-if)# end
Switch# show interfaces fastethernet 5/2 switchport
Name:Fa5/2
Switchport:Enabled
Administrative Mode:private-vlan promiscuous
Operational Mode:private-vlan promiscuous
Administrative Trunking Encapsulation:negotiate
Operational Trunking Encapsulation:native
Negotiation of Trunking:Off
Access Mode VLAN:1 (default)
Trunking Native Mode VLAN:1 (default)
Voice VLAN:none
Administrative Private VLAN Host Association:none
Administrative Private VLAN Promiscuous Mapping:200 (VLAN0200) 2 (VLAN0002)
Private VLAN Trunk Native VLAN:none
Administrative Private VLAN Trunk Encapsulation:dot1q
Administrative Private VLAN Trunk Normal VLANs:none
Administrative Private VLAN Trunk Private VLANs:none
Operational Private VLANs:
    200 (VLAN0200) 2 (VLAN0002)
Trunking VLANs Enabled:ALL
Pruning VLANs Enabled:2-1001
Capture Mode Disabled
Capture VLANs Allowed:ALL
```

Configuring a Layer 2 Interface as a PVLAN Host Port

To configure a Layer 2 interface as a PVLAN host port, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters configuration mode.
Step 2	Switch(config)# interface { fastethernet gigabitethernet tengigabitethernet } <i>slot/port</i>	Specifies the LAN port to configure.
Step 3	Switch(config-if)# switchport mode private-vlan { host promiscuous trunk promiscuous trunk [secondary] }	Configures a Layer 2 interface as a PVLAN host port.
Step 4	Switch(config-if)# [no] switchport private-vlan host-association <i>primary_vlan_ID secondary_vlan_ID</i>	Associates the Layer 2 interface with a PVLAN. You can use the no keyword to delete all associations from the primary VLAN.
Step 5	Switch(config-if)# end	Exits configuration mode.
Step 6	Switch# show interfaces { fastethernet gigabitethernet tengigabitethernet } <i>slot/port</i> switchport	Verifies the configuration.

This example shows how to configure interface FastEthernet 5/1 as a PVLAN host port and verify the configuration:

```
Switch# configure terminal
Switch(config)# interface fastethernet 5/1
Switch(config-if)# switchport mode private-vlan host
Switch(config-if)# switchport private-vlan host-association 202 440
Switch(config-if)# end

Switch# show interfaces fastethernet 5/1 switchport
Name: Fa5/1
Switchport: Enabled
Administrative Mode: private-vlan host
Operational Mode: private-vlan host
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Voice VLAN: none
Appliance trust: none
Administrative Private Vlan
  Host Association: 202 (VLAN0202) 440 (VLAN0440)
  Promiscuous Mapping: none
  Trunk encapsulation : dot1q
  Trunk vlans:
Operational private-vlan(s):
  202 (VLAN0202) 440 (VLAN0440)
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL
```

Configuring a Layer 2 Interface as an Isolated PVLAN Trunk Port

To configure a Layer 2 interface as an isolated PVLAN trunk port, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface { fastethernet gigabitethernet tengigabitethernet } <i>slot/port</i>	Specifies the LAN port to configure.
Step 3	Switch(config-if)# switchport mode private-vlan { host promiscuous trunk promiscuous trunk [secondary] }	Configures a Layer 2 interface as a PVLAN trunk port.

	Command	Purpose
Step 4	Switch(config-if)# [no] switchport private-vlan association trunk <i>primary_vlan_ID</i> <i>secondary_vlan_ID</i>	Configures association between primary VLANs and secondary VLANs the PVLAN trunk port with a PVLAN. Note Multiple PVLAN pairs can be specified using this command so that a PVLAN trunk port can carry multiple secondary VLANs. If an association is specified for the existing primary VLAN, the existing association is replaced. If there is no trunk association, any packets received on secondary VLANs are dropped. You can use the no keyword to delete all associations from the primary VLAN.
Step 5	Switch(config-if)# [no] switchport private-vlan trunk allowed vlan <i>vlan_list</i> all none [add remove except] <i>vlan_atom[,vlan_atom...]</i>	Configures a list of allowed normal VLANs on a PVLAN trunk port. You can use the no keyword to remove all allowed normal VLANs on a PVLAN trunk port.
Step 6	Switch(config-if)# switchport private-vlan trunk native vlan <i>vlan_id</i>	Configures a VLAN to which untagged packets (as in IEEE 802.1Q tagging) are assigned on a PVLAN trunk port. If there is no native VLAN configured, all untagged packets are dropped. If the native VLAN is a secondary VLAN and the port does not have the association for the secondary VLAN, the untagged packets are dropped. You can use the no keyword to remove all native VLANs on a PVLAN trunk port.
Step 7	Switch(config-if)# end	Exits configuration mode.
Step 8	Switch# show interfaces { fastethernet gigabitethernet tengigabitethernet } <i>slot/port</i> switchport	Verifies the configuration.

This example shows how to configure interface FastEthernet 5/2 as a secondary trunk port, and verify the configuration:

```
Switch# configure terminal
Switch(config)# interface fastethernet 5/2
Switch(config-if)# switchport mode private-vlan trunk secondary
Switch(config-if)# switchport private-vlan trunk native vlan 10
Switch(config-if)# switchport private-vlan trunk allowed vlan 10. 3-4
Switch(config-if)# switchport private-vlan association trunk 3 301
Switch(config-if)# end
Switch# show interfaces fastethernet 5/2 switchport
Name: Fa5/2
  Switchport: Enabled
Administrative Mode: private-vlan trunk secondary
Operational Mode: private-vlan trunk secondary
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
```



```

Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
Administrative private-vlan host-association: none A
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: 10
Administrative private-vlan trunk Native VLAN tagging: enabled
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk associations:
    3 (VLAN0003) 301 (VLAN0301)
Administrative private-vlan trunk mappings: none
Operational private-vlan: none
Operational Normal VLANs: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled Capture VLANs Allowed: ALL

Unknown unicast blocked: disabled
Unknown multicast blocked: disabled
Appliance trust: none

```

Configuring a Layer 2 Interface as a Promiscuous PVLAN Trunk Port

To configure a Layer 2 interface as a promiscuous PVLAN trunk port, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface { fastethernet gigabitethernet tengigabitethernet } <i>slot/port</i>	Specifies the LAN interface to configure.
Step 3	Switch(config-if)# switchport mode private-vlan { host promiscuous trunk promiscuous trunk [secondary] }	Configures a Layer 2 interface as a PVLAN promiscuous trunk port.
Step 4	Switch(config-if)# [no] switchport private-vlan mapping [trunk] <i>primary_vlan_ID</i> { <i>secondary_vlan_list</i> add <i>secondary_vlan_list</i> remove <i>secondary_vlan_list</i> }	Maps the promiscuous PVLAN port to a primary VLAN and to selected secondary VLANs. This command offers 3 levels of removal. See the examples that follow this table.
Step 5	Switch(config-if)# end	Exits configuration mode.
Step 6	Switch# show interfaces { fastethernet gigabitethernet tengigabitethernet } <i>slot/port</i> switchport	Verifies the configuration.



Note

The maximum number of unique PVLAN pairs supported by the **switchport private-vlan mapping trunk** command is 500. For example, 500 isolated secondary VLANs could map to 500 primary VLANs, because only one isolated VLAN association per primary VLAN is supported. Or, 500 community secondary VLANs could map to one primary VLAN. Or, 250 community secondary VLANs could map to 1 primary VLAN, and another 250 community secondary VLANs could map to another primary VLAN for a total of 500 pairs.



Note

By default, when you configure the mode to PVLAN trunk **promiscuous**, the native VLAN is set to 1.

The **[no] switchport private-vlan mapping** command provides the following three levels of removal:

- Remove one or more secondary VLANs from the list. For example:
Switch(config-if)# **switchport private-vlan mapping trunk 2 remove 222**
- Remove the entire mapping of PVLAN promiscuous trunk port to the specified primary VLAN (and all of its selected secondary VLANs). For example:
Switch(config-if)# **no switchport private-vlan mapping trunk 2**
- Remove the mapping of a PVLAN promiscuous trunk port to all previously configured primary VLANs (and all of their selected secondary VLANs). For example:
Switch(config-if)# **no switchport private-vlan mapping trunk**

When you configure a Layer 2 interface as a PVLAN promiscuous port, note the following:

- Multiple PVLAN pairs can be specified using the **switchport private-vlan mapping trunk** command so that a promiscuous trunk port can carry multiple primary VLANs.
- The *secondary_vlan_list* parameter cannot contain spaces. It can contain multiple comma-separated items. Each item can be a single PVLAN ID or a hyphenated range of PVLAN IDs.
- Enter a *secondary_vlan_list* or use the **add** keyword with a *secondary_vlan_list* to map the secondary VLANs to the PVLAN promiscuous port.
- Use the **remove** keyword with a *secondary_vlan_list* to clear the mapping between secondary VLANs and the PVLAN promiscuous port.

This example shows how to configure interface FastEthernet 5/2 as a promiscuous trunk port and to verify the configuration:

```
Switch# configure terminal
Switch(config)# interface fastethernet 5/2
Switch(config-if)# switchport mode private-vlan trunk promiscuous
Switch(config-if)# switchport private-vlan trunk native vlan 10
Switch(config-if)# switchport private-vlan trunk allowed vlan 10, 3-4
Switch(config-if)# switchport private-vlan mapping trunk 3 301, 302
Switch(config-if)# end
Switch# show interfaces fastethernet 5/2 switchport
Name: Fa5/2
Switchport: Enabled
Administrative Mode: private-vlan trunk promiscuous
Operational Mode: private-vlan trunk promiscuous
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: 10
Administrative private-vlan trunk Native VLAN tagging: enabled
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: 3-4,10
Administrative private-vlan trunk associations: none
Administrative private-vlan trunk mappings:
    3 (VLAN0003) 301 (VLAN0301) 302 (VLAN0302)
Operational private-vlan:
    3 (VLAN0003) 301 (VLAN0301) 302 (VLAN0302)
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
```

```

Capture Mode Disabled
Capture VLANs Allowed: ALL

Unknown unicast blocked: disabled
Unknown multicast blocked: disabled
Appliance trust: none

```

Permitting Routing of Secondary VLAN Ingress Traffic



Note

Isolated, community VLANs, and twoway-community VLANs are called secondary VLANs.

To permit routing of secondary VLAN ingress traffic, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface vlan <i>primary_vlan_ID</i>	Enters interface configuration mode for the primary VLAN.
Step 3	Switch(config-if)# [no] private-vlan mapping <i>primary_vlan_ID</i> { <i>secondary_vlan_list</i> add <i>secondary_vlan_list</i> remove <i>secondary_vlan_list</i> }	To permit routing on the secondary VLAN ingress traffic, map the secondary VLAN to the primary VLAN. You can use the no keyword to delete all associations from the primary VLAN.
Step 4	Switch(config-if)# end	Exits configuration mode.
Step 5	Switch# show interface private-vlan mapping	Verifies the configuration.

When you permit routing on the secondary VLAN ingress traffic, note the following:

- The **private-vlan mapping** interface configuration command only affects PVLAN ingress traffic that is Layer 3 switched.
- The *secondary_vlan_list* parameter cannot contain spaces. It can contain multiple comma-separated items. Each item can be a single PVLAN ID or a hyphenated range of PVLAN IDs.
- Enter a *secondary_vlan_list* parameter or use the **add** keyword with a *secondary_vlan_list* parameter to map the secondary VLANs to the primary VLAN.
- Use the **remove** keyword with a *secondary_vlan_list* parameter to clear the mapping between secondary VLANs and the primary VLAN.

This example shows how to permit routing of secondary VLAN ingress traffic from PVLANS 303 through 307, 309, and 440 and verify the configuration:

```

Switch# configure terminal
Switch(config)# interface vlan 202
Switch(config-if)# private-vlan mapping add 303-307,309,440
Switch(config-if)# end
Switch# show interfaces private-vlan mapping
Interface Secondary VLAN Type
-----
vlan202    303          community
vlan202    304          community
vlan202    305          community
vlan202    306          community
vlan202    307          community

```

```
vlan202 309 community
vlan202 440 isolated

Switch#
```

Configuring PVLAN over EtherChannel

After creating a Layer 2 Etherchannel, you can configure it with any of the four PVLAN port modes (promiscuous host, secondary host, isolated trunk, promiscuous trunk).

This section includes the following topics:

- [Configuring a Layer 2 EtherChannel, page 47-24](#)
- [Configuring a Layer 2 Etherchannel as a PVLAN Promiscuous Port, page 47-24](#)
- [Configuring a Layer 2 EtherChannel as a PVLAN Host Port, page 47-26](#)
- [Configuring a Layer 2 EtherChannel as an Isolated PVLAN Trunk Port, page 47-27](#)
- [Configuring a Layer 2 Etherchannel as a Promiscuous PVLAN Trunk Port, page 47-28](#)

Configuring a Layer 2 EtherChannel

Do the following:

-
- Step 1** Configure a VLAN as a PVLAN.
Refer to the URL:
<http://www.cisco.com/en/US/docs/switches/lan/catalyst4500/12.2/01xo/configuration/guide/pvlans.html#wp1174853>
- Step 2** Associate a secondary VLAN with a primary VLAN.
Refer to the URL:
<http://www.cisco.com/en/US/docs/switches/lan/catalyst4500/12.2/01xo/configuration/guide/pvlans.html#wp1121802>
- Step 3** Configuring a Layer 2 EtherChannel.
Refer to the URL
<http://www.cisco.com/en/US/docs/switches/lan/catalyst4500/12.2/01xo/configuration/guide/channel.html#wp1020670>
-

Configuring a Layer 2 Etherchannel as a PVLAN Promiscuous Port

Perform the following task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface port-channel <i>interface-number</i>	Specifies the LAN interface to configure.

	Command	Purpose
Step 3	Switch(config-if)# switchport mode private-vlan { host promiscuous trunk promiscuous trunk [secondary]}	Configures a Layer 2 Etherchannel as a PVLAN promiscuous port.
Step 4	Switch(config-if)# [no] switchport private-vlan mapping [trunk] <i>primary_vlan_ID</i> { <i>secondary_vlan_list</i> add <i>secondary_vlan_list</i> }	(Maps the PVLAN promiscuous port to a primary VLAN and to selected secondary VLANs.
Step 5	Switch(config-if)# end	Exits configuration mode.
Step 6	Switch# show interface port-channel <i>interface-number</i> switchport	Verifies the configuration.

**Note**

The maximum number of unique PVLAN pairs supported by the **switchport private-vlan mapping** command is 1000.

When you configure a Layer 2 Etherchannel as a PVLAN promiscuous port, note the following:

- The *secondary_vlan_list* parameter cannot contain spaces. It can contain multiple comma-separated items. Each item can be a single PVLAN ID or a hyphenated range of PVLAN IDs.
- Enter a *secondary_vlan_list* or use the **add** keyword with a *secondary_vlan_list* to map the secondary VLANs to the PVLAN promiscuous port.
- Use the **remove** keyword with a *secondary_vlan_list* to clear the mapping between secondary VLANs and the PVLAN promiscuous port.

This example shows how to configure interface port channel 63 as a PVLAN promiscuous port, map it to a PVLAN, and verify the configuration:

```
Switch# configure terminal
Switch(config)# interface port-channel 63
Switch(config-if)# switchport mode private-vlan promiscuous
Switch(config-if)# switchport private-vlan mapping 200 2
Switch(config-if)# end
Switch# show interfaces port-channel 63 switchport
Name: Po63
Switchport: Enabled
Administrative Mode: private-vlan promiscuous
Operational Mode: private-vlan promiscuous
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Voice VLAN: none
Administrative Private VLAN Host Association: none
Administrative Private VLAN Promiscuous Mapping: 200 (VLAN0200) 2 (VLAN0002)
Private VLAN Trunk Native VLAN: none
Administrative Private VLAN Trunk Encapsulation: dot1q
Administrative Private VLAN Trunk Normal VLANs: none
Administrative Private VLAN Trunk Private VLANs: none
Operational Private VLANs:
    200 (VLAN0200) 2 (VLAN0002)
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL
```

Configuring a Layer 2 EtherChannel as a PVLAN Host Port

To configure a Layer 2 EtherChannel as a PVLAN host port, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface port-channel <i>interface-number</i>	Specifies the LAN interface to configure.
Step 3	Switch(config-if)# switchport mode private-vlan {host promiscuous trunk promiscuous trunk [secondary]}	Configures a Layer 2 Etherchannel as a PVLAN host port.
Step 4	Switch(config-if)# [no] switchport private-vlan host-association <i>primary_vlanb_ID</i> <i>secondary_vlan_ID</i>	Associates the Layer 2 interface with a PVLAN. You can use the no keyword to delete all associations from the primary VLAN.
Step 5	Switch(config-if)# end	Exits configuration mode.
Step 6	Switch# show interface port-channel <i>interface-number</i> switchport	Verifies the configuration.

This example shows how to configure interface port channel 63 as a PVLAN host port and to verify the configuration:

```
Switch# configure terminal
Switch(config)# interface port-channel 63
Switch(config-if)# switchport mode private-vlan host
Switch(config-if)# switchport private-vlan host-association 202 440
Switch(config-if)# end
Switch# show interfaces port-channel 63 switchport
Name: Po63
Switchport: Enabled
Administrative Mode: private-vlan host
Operational Mode: private-vlan host
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Voice VLAN: none
Appliance trust: none
Administrative Private Vlan
  Host Association: 202 (VLAN0202) 440 (VLAN0440)
  Promiscuous Mapping: none
  Trunk encapsulation : dot1q
  Trunk vlans:
Operational private-vlan(s):
  202 (VLAN0202) 440 (VLAN0440)
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL
```

Configuring a Layer 2 EtherChannel as an Isolated PVLAN Trunk Port

To configure a Layer 2 EtherChannel as an isolated PVLAN trunk port, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface port-channel <i>interface-number</i>	Specifies the LAN interface to configure.
Step 3	Switch(config-if)# switchport mode private-vlan { host promiscuous trunk promiscuous trunk [secondary]}	Configures a Layer 2 Etherchannel as a PVLAN trunk port.
Step 4	Switch(config-if)# [no] switchport private-vlan association trunk <i>primary_vlanb_ID</i> <i>secondary_vlan_ID</i>	<p>(Configures association between primary VLANs and secondary VLANs the PVLAN trunk port with a PVLAN.</p> <p>Note Multiple PVLAN pairs can be specified using this command so that a PVLAN trunk port can carry multiple secondary VLANs. If an association is specified for the existing primary VLAN, the existing association is replaced. If there is no trunk association, any packets received on secondary VLANs are dropped.</p> <p>You can use the no keyword to delete all associations from the primary VLAN.</p>
Step 5	Switch(config-if)# [no] switchport private-vlan trunk allowed vlan <i>vlan_list</i> [all none [add remove except] <i>vlan_atom</i> [, <i>vlan_atom</i> ...]	<p>(Configures a list of allowed normal VLANs on a PVLAN trunk port</p> <p>You can use the no keyword to remove all allowed normal VLANs on a PVLAN trunk port.</p>
Step 6	Switch(config-if)# switchport private-vlan trunk native vlan <i>vlan_id</i>	<p>(Configures a VLAN to which untagged packets (as in IEEE 802.1Q tagging) are assigned on a PVLAN trunk port.</p> <p>If there no native VLAN is configured, all untagged packets are dropped.</p> <p>If the native VLAN is a secondary VLAN and the port does not have the association for the secondary VLAN, the untagged packets are dropped.</p> <p>You can use the no keyword to remove all native VLANs on a PVLAN trunk port.)</p>
Step 7	Switch(config-if)# end	Exits configuration mode.
Step 8	Switch# show interfaces Port-channel <i>interface-number</i> switchport	Verifies the configuration.

This example shows how to configure interface port channel 63 as a secondary trunk port, and to verify the configuration:

```
Switch# configure terminal
Switch(config)# interface port-channel 63
Switch(config-if)# switchport mode private-vlan trunk secondary
Switch(config-if)# switchport private-vlan trunk native vlan 10
Switch(config-if)# switchport private-vlan trunk allowed vlan 10. 3-4
Switch(config-if)# switchport private-vlan association trunk 3 301
Switch(config-if)# end
Switch# show interfaces port-channel 63 switchport
Name: Po63
Switchport: Enabled
Administrative Mode: private-vlan trunk secondary
Operational Mode: private-vlan trunk secondary
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: 10
Administrative private-vlan trunk Native VLAN tagging: enabled
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk associations:
    3 (VLAN0003) 301 (VLAN0301)
Administrative private-vlan trunk mappings: none
Operational private-vlan: none
Operational Normal VLANs: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled Capture VLANs Allowed: ALL
Unknown unicast blocked: disabled
Unknown multicast blocked: disabled
Appliance trust: none
```

Configuring a Layer 2 Etherchannel as a Promiscuous PVLAN Trunk Port

To configure a Layer 2 Etherchannel as a promiscuous PVLAN trunk port, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface port-channel <i>interface-number</i>	Specifies the LAN interface to configure.
Step 3	Switch(config-if)# switchport mode private-vlan {host promiscuous trunk promiscuous trunk [secondary]}	Configures a Layer 2 Etherchannel as a PVLAN promiscuous trunk port.
Step 4	Switch(config-if)# [no] switchport private-vlan mapping [trunk] primary_vlan_ID {secondary_vlan_list add secondary_vlan_list remove secondary_vlan_list}	Maps the promiscuous PVLAN port to a primary VLAN and to the selected secondary VLANs. This command offers 3 levels of removal.

	Command	Purpose
Step 5	Switch(config-if)# end	Exits configuration mode.
Step 6	Switch# show interfaces port-channel <i>interface-number</i> switchport	Verifies the configuration.

**Note**

The maximum number of unique PVLAN pairs supported by the switchport private-vlan mapping trunk command is 500. For example, 500 isolated secondary VLANs could map to 500 primary VLANs, because only one isolated VLAN association per primary VLAN is supported. Or, 500 community secondary VLANs could map to one primary VLAN. Or, 250 community secondary VLANs could map to 1 primary VLAN, and another 250 community secondary VLANs could map to another primary VLAN for a total of 500 pairs.

**Note**

By default, when you configure the mode to private VLAN trunk promiscuous, the native VLAN is set to 1.

The [no] **switchport private-vlan mapping** command provides the following three levels of removal:

- Remove one or more secondary VLANs from the list.

For example:

```
Switch(config-if)# switchport private-vlan mapping trunk 2 remove 222
```

- Remove the entire mapping of PVLAN promiscuous trunk port to the specified primary VLAN (and all of its selected secondary VLANs).

For example:

```
Switch(config-if)# no switchport private-vlan mapping trunk 2
```

- Remove the mapping of a PVLAN promiscuous trunk port to all previously configured primary VLANs (and all of their selected secondary VLANs).

For example:

```
Switch(config-if)# no switchport private-vlan mapping trunk
```

When you configure a Layer 2 etherchannel as a PVLAN promiscuous trunk port, observe that multiple private VLAN pairs can be specified with the **switchport private-vlan mapping trunk** command so that a promiscuous trunk port can carry multiple primary VLANs.

- The *secondary_vlan_list* parameter cannot contain spaces. It can contain multiple comma-separated items. Each item can be a single PVLAN ID or a hyphenated range of PVLAN IDs.
- Enter a *secondary_vlan_list* or use the **add** keyword with a *secondary_vlan_list* to map the secondary VLANs to the PVLAN promiscuous port.
- Use the **remove** keyword with a *secondary_vlan_list* to clear the mapping between secondary VLANs and the PVLAN promiscuous port.

This example shows how to configure interface Port-channel 63 as a promiscuous trunk port and to verify the configuration:

```
Switch# configure terminal
Switch(config)# interface port-channel 63
Switch(config-if)# switchport mode private-vlan trunk promiscuous
```

```
Switch(config-if)# switchport private-vlan trunk native vlan 10
Switch(config-if)# switchport private-vlan trunk allowed vlan 10, 3-4
Switch(config-if)# switchport private-vlan mapping trunk 3 301, 302
Switch(config-if)# end
Switch# show interfaces port-channel 63 switchport
Name: Po63
Switchport: Enabled
Administrative Mode: private-vlan trunk promiscuous
Operational Mode: private-vlan trunk promiscuous
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: 10
Administrative private-vlan trunk Native VLAN tagging: enabled
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: 3-4,10
Administrative private-vlan trunk associations: none
Administrative private-vlan trunk mappings:
    3 (VLAN0003) 301 (VLAN0301) 302 (VLAN0302)
Operational private-vlan:
    3 (VLAN0003) 301 (VLAN0301) 302 (VLAN0302)
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL
Unknown unicast blocked: disabled
Unknown multicast blocked: disabled
Appliance trust: none
```



Configuring MACsec Encryption

This chapter describes how to configure Media Access Control Security (MACsec) encryption on the Catalyst 4500 series switch.

MACsec is the IEEE 802.1AE standard for authenticating and encrypting packets between two MACsec-capable devices. The Catalyst 4500 series switch supports 802.1AE encryption with MACsec Key Agreement (MKA) on downlink ports for encryption between the switch and host devices. The switch also supports MACsec link layer switch-to-switch security by using Cisco TrustSec Network Device Admission Control (NDAC) and the Security Association Protocol (SAP) key exchange. Link layer security can include both packet authentication between switches and MACsec encryption between switches (encryption is optional).



Note

MACsec is supported on the Catalyst 4500 series switch universal k9 image. It is not supported with the NPE license or with a LAN Base service image.

All downlink ports on a switch can run Cisco TrustSec MACsec link layer switch-to-switch security.

Table 1 **MACsec Support on Switch Ports**

Interface	Connections	MACsec support
User-facing downlink ports	Switch-to-host	MKA MACsec encryption
Switchports connected to other switches	Switch-to-switch	Cisco TrustSec NDAC MACsec MKA MACsec encryption

Cisco TrustSec and Cisco SAP are meant only for switch-to-switch links and are not supported on switch ports connected to end hosts, such as PCs or IP phones. MKA is supported on both switch-to-host facing links, and switch-to-switch links as well. Host-facing links typically use flexible authentication ordering for handling heterogeneous devices with or without IEEE 802.1X, and can optionally use MKA encryption. Cisco NDAC and SAP are mutually exclusive with Network Edge Access Topology (NEAT), which is used for compact switches to extend security outside the wiring closet.

This chapter includes the following major sections:

- [Understanding Media Access Control Security and MACsec Key Agreement, page 48-2](#)
- [Configuring MACsec and MACsec Key Agreement, page 48-7](#)
- [Understanding MKA MACsec with EAP-TLS, page 48-12](#)
- [Understanding Cisco TrustSec MACsec, page 48-20](#)
- [Configuring Cisco TrustSec MACsec, page 48-22](#)

**Note**

For more information, refer to the *Cisco TrustSec Switch Configuration Guide*:

<http://www.cisco.com/en/US/docs/switches/lan/trustsec/configuration/guide/trustsec.html>

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

Understanding Media Access Control Security and MACsec Key Agreement

MACsec, defined in 802.1AE, provides MAC-layer encryption over wired networks by using out-of-band methods for encryption keying. The MACsec Key Agreement (MKA) Protocol provides the required session keys and manages the required encryption keys. MKA and MACsec are implemented after successful authentication using the 802.1X Extensible Authentication Protocol (EAP) and EAP-Transport Layer Security (EAP-TLS) framework. MKA MACsec supports both host facing links (links between network access devices and endpoint devices such as a PC or IP phone) and switch-to-switch links, beginning in Cisco IOS Release 15.2(5)E and Cisco IOS XE Release 3.9.0E.

A switch using MACsec accepts either MACsec or non-MACsec frames, depending on the policy associated with the client. MACsec frames are encrypted and protected with an integrity check value (ICV). When the switch receives frames from the client, it decrypts them and calculates the correct ICV by using session keys provided by MKA. The switch compares that ICV to the ICV within the frame. If they are not identical, the frame is dropped. The switch also encrypts and adds an ICV to any frames sent over the secured port (the access point used to provide the secure MAC service to a client) using the current session key.

The MKA Protocol manages the encryption keys used by the underlying MACsec protocol. The basic requirements of MKA are defined in 802.1X-2010. The MKA Protocol extends 802.1X to allow peer discovery with confirmation of mutual authentication and sharing of MACsec secret keys to protect data exchanged by the peers.

Pre-shared keys (PSKs) are used to generate Connectivity Association Keys (CAKs). In symmetric cryptography, PSK means a key or a shared secret. This key is shared between parties before it is used. The PSK is used to generate the Key Encryption Key (KEK) and the integrity check value (ICV) Key (ICK).

**Note**

PSK does not support Security Group Tagging and AES-GCM-256 encryption.

In a switch-to-switch connection using the PSK, there is no concept of the authenticator, because of the EAP authentication on the switch. So the switch with highest priority becomes the Key Server (KS). In the current implementation, MKA can act as a non-KS without much change, except for accepting the PSK instead of the CAK.

The EAP framework implements MKA as a newly defined EAP-over-LAN (EAPOL) packet. EAP authentication produces a master session key (MSK) shared by both partners in the data exchange. Entering the EAP session ID generates a secure connectivity association key name (CKN). Because the switch is the authenticator, it is also the key server, generating a random 128-bit secure association key

(SAK), which it sends it to the client partner. The client is never a key server and can only interact with a single MKA entity, the key server. After key derivation and generation, the switch sends periodic transports to the partner at a default interval of 2 seconds.

The CAK and CKN will be derived from the configured PSK name and value

The packet body in an EAPOL Protocol Data Unit (PDU) is referred to as a MACsec Key Agreement PDU (MKPDU). MKA sessions and participants are deleted when the MKA lifetime (6 seconds) passes with no MKPDU received from a participant. For example, if a client disconnects, the participant on the switch continues to operate MKA until 6 seconds have elapsed after the last MKPDU is received from the client.

These sections provide more details:

- [MKA Policies, page 48-3](#)
- [Key Lifetime and Hitless Key Rollover, page 48-3](#)
- [Encryption Algorithms for MKA Control Packets, page 48-4](#)
- [Virtual Ports, page 48-4](#)
- [MACsec, page 48-5](#)
- [MACsec, MKA, and 802.1X Host Modes, page 48-5](#)
- [MKA Statistics, page 48-6](#)

MKA Policies

By default, the MKA protocol default policy is enabled on an interface. However, you can apply a defined MKA policy to an interface. Removing the MKA policy configures the default MKA policy on that interface.

You can configure these options:

- Policy name, not to exceed 16 ASCII characters.
- Confidentiality (encryption) offset of 0, 30, or 50 bytes for each physical interface.
- Replay protection. You can configure MACsec window size, as defined by the number of out-of-order frames that are accepted. This value is used while installing the security associations in the MACsec. A value of 0 means that frames are accepted only in the correct order.



Note

MKA is not supported in the Virtual Switching System (VSS) mode.

Key Lifetime and Hitless Key Rollover

A MACsec key chain (MKA) can have multiple pre-shared keys (PSKs) each configured with a key ID and an optional lifetime. A key lifetime specifies the time period the key is valid. In the absence of a lifetime configuration, the default lifetime is unlimited. MKA rolls over to the next configured valid pre-shared key in the key chain, when a valid key expires. Time zone of the key can be local or UTC. Default time zone is UTC.

MKA rolls over to the next configured valid pre-shared key in the key chain, when a valid key expires.

Use the **key chain** *key-chain-name* **macsec** to configure the MACsec key chain.

To roll over to the next key within the same key chain, configure a second key in the key chain, and a lifetime for the first key. When the lifetime of the first key expires, it automatically rolls over to the next key in the list. If the same key is configured on both sides of the link at the same time, then the key rollover is hitless, that is, key rolls over without traffic interruption.

**Note**

The lifetime of the keys need to be overlapped to achieve hitless key rollover.

Encryption Algorithms for MKA Control Packets

Cryptographic algorithm selection for MKA control protocol packets encryption is as follows:

- The cryptographic algorithm to encrypt MKA control protocol packets is configured as part of the key chain. There can be only one cryptographic algorithm configured per key chain.
- A key server uses the configured MKA cryptographic algorithm from the key chain.
- All nonkey servers must use the same cryptographic algorithm as the key server.

If an MKA cryptographic algorithm is not configured, a default cryptographic algorithm of AES-CMAC-128 (Cipher-based Message Authentication Code with 128-bit Advanced Encryption Standard) is used.

The following is a sample encryption algorithm for data packets:

```
Switch(config)# mka policy p1
Switch(config-mka-policy)# macsec-cipher-suite gsm-aes-256
```

The following is a sample encryption algorithm for MKA control packets:

```
Switch(config)# key chain key-chain-name macsec
Switch(config-keychain-macsec)# key 01
Switch(config-keychain-macsec-key)# key-string 0001
Switch(config-keychain-macsec-key)# cryptographic-algorithm [aes-128-cmac | aes-256-cmac]
Switch(config-keychain-macsec-key)# end
```

Virtual Ports

You use virtual ports for multiple secured connectivity associations on a single physical port. Each connectivity association (pair) represents a virtual port, with a maximum of two virtual ports per physical port. Only one of the two virtual ports can be part of a data VLAN; the other must externally tag its packets for the voice VLAN. You cannot simultaneously host secured and unsecured sessions in the same VLAN on the same port. Because of this limitation, 802.1X multiple authentication mode is not supported.

The exception to this limitation is in multiple-host mode when the first MACsec supplicant is successfully authenticated and connected to a hub that is connected to the switch. A non-MACsec host connected to the hub can send traffic without authentication because it is in multiple-host mode. We do not recommend using multi-host mode because after the first successful client, authentication is not required for other clients.

Virtual ports represent an arbitrary identifier for a connectivity association and have no meaning outside the MKA Protocol. A virtual port corresponds to a separate logical port ID. Valid port IDs for a virtual port are 0x0002 to 0xFFFF. Each virtual port receives a unique secure channel identifier (SCI) based on the MAC address of the physical interface concatenated with a 16-bit port ID.

MACsec

A Catalyst 4500 series switch supervisor running MACsec maintains the configuration files that show which ports on the switch support MACsec. The supervisor-engine performs these functions:

- Processes secure channel and secure association creation and deletion.
- Programs the hardware with the derived secure association key (SAK) for encryption and decryption.

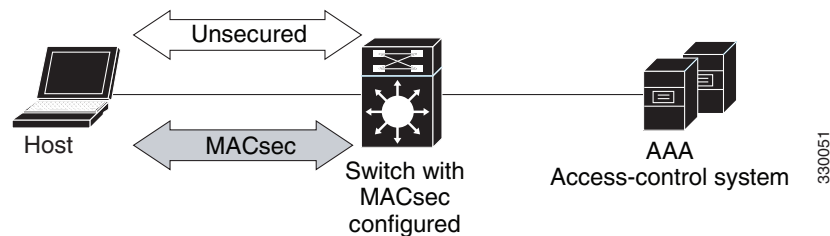
MACsec, MKA, and 802.1X Host Modes

You can use MACsec and the MKA Protocol with 802.1X single-host mode, multiple-host mode, or Multi Domain Authentication (MDA) mode.

Single-Host Mode

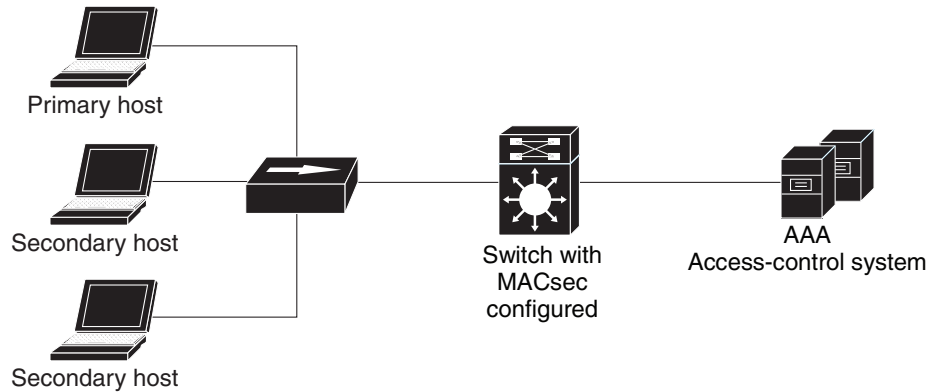
Figure 48-1 shows how a single EAP authenticated session is secured by MACsec using MKA.

Figure 48-1 *MACsec in Single-Host Mode with a Secured Data Session*



Multiple-Host Mode

In standard (not 802.1X-2010) 802. multiple-host mode, a port is open or closed based on a single authentication. If one user, the primary secured client services client host, is authenticated, the same level of network access is provided to any host connected to the same port. If a secondary host is a MACsec supplicant, it cannot be authenticated and traffic would no flow. A secondary host that is a non-MACsec host can send traffic to the network without authentication because it is in multiple-host mode. See Figure 48-2.

Figure 48-2 *MACsec in Standard Multiple-Host Mode - Unsecured*

We do not recommend using multi-host mode because after the first successful client, authentication is not required for other clients, which is not secure.

MKA Statistics

Some MKA counters are aggregated globally, while others are updated both globally and per session. You can also obtain information about the status of MKA sessions.

This is an example of the **show mka statistics** command output:

```

Switch# show mka statistics
MKA Global Statistics
=====
MKA Session Totals
    Secured..... 32
    Reauthentication Attempts.. 31

    Deleted (Secured)..... 1
    Keepalive Timeouts..... 0

CA Statistics
    Pairwise CAKs Derived..... 32
    Pairwise CAK Rekeys..... 31
    Group CAKs Generated..... 0
    Group CAKs Received..... 0

SA Statistics
    SAKs Generated..... 32
    SAKs Rekeyed..... 31
    SAKs Received..... 0
    SAK Responses Received.... 32

MKPDU Statistics
    MKPDUs Validated & Rx..... 580
        "Distributed SAK"..... 0
        "Distributed CAK"..... 0
    MKPDUs Transmitted..... 597
        "Distributed SAK"..... 32
        "Distributed CAK"..... 0

MKA Error Counter Totals
=====
    Bring-up Failures..... 0
    Reauthentication Failures..... 0
  
```



```

SAK Failures
  SAK Generation..... 0
  Hash Key Generation..... 0
  SAK Encryption/Wrap..... 0
  SAK Decryption/Unwrap..... 0

CA Failures
  Group CAK Generation..... 0
  Group CAK Encryption/Wrap..... 0
  Group CAK Decryption/Unwrap..... 0
  Pairwise CAK Derivation..... 0
  CKN Derivation..... 0
  ICK Derivation..... 0
  KEK Derivation..... 0
  Invalid Peer MACsec Capability.. 2

MACsec Failures
  Rx SC Creation..... 0
  Tx SC Creation..... 0
  Rx SA Installation..... 0
  Tx SA Installation..... 0

MKPDU Failures
  MKPDU Tx..... 0
  MKPDU Rx Validation..... 0
  MKPDU Rx Bad Peer MN..... 0
  MKPDU Rx Non-recent Peerlist MN.. 0

```

For description of the output fields, see the command reference for this release.

Configuring MACsec and MACsec Key Agreement

- [Default MKA MACsec Configuration, page 48-7](#)
- [Configuring an MKA Policy, page 48-7](#)
- [Configuring MACsec on an Interface, page 48-9](#)
- [Configuring MKA Pre-Shared Key, page 48-10](#)

Default MKA MACsec Configuration

MACsec is disabled. No MACsec Key Agreement (MKA) policies are configured.

Configuring an MKA Policy

To create an MKA Protocol policy, perform this task. Note that MKA also requires that you enable 802.1X.

	Command	Purpose
Step 1	configure terminal	Enters global configuration mode.
Step 2	mka policy <i>policy-name</i>	Identifies an MKA policy, and enter MKA policy configuration mode. The maximum policy name length is 16 characters.

	Command	Purpose
Step 3	replay-protection window-size <i>frames</i>	Enables replay protection, and configure the window size in number of frames. The range is from 0 to 4294967295. The default window size is 0. Entering a window size of 0 is not the same as entering the no replay-protection command. Configuring a window size of 0 uses replay protection with a strict ordering of frames. Entering no replay-protection turns off MACsec replay-protection.
Step 4	end	Returns to privileged EXEC mode.
Step 5	show mka policy	Verifies your entries.
Step 6	copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example configures the MKA policy *relay-policy*:

```
Switch(config)# mka policy relay-policy
Switch(config-mka-policy)# replay-protection window-size 300
Switch(config-mka-policy)# end
```

Let's say that we configure an MKA policy as follows:

```
Switch# conf terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# mka policy poll
Switch(config-mka-policy)# replay-protection window-size 1000
Switch(config-mka-policy)# confidentiality-offset 50
Switch(config-mka-policy)# end
```

We observe the following:

- The payload starting from the SA (source MAC address) + 50 bytes offset is encrypted.
- Replay protect is YES with a window size of 1000. If the frame received has a packet number (PN) of 1020, for example, all frames with a PN of 20 to 1020 can come out of order (i.e, frame with PN 900 can come first and frame with PN 800 can come later). However, if a frame with a PN of 1021 is received first, followed by a frame with a PN of 20, the frame with PN of 20 is dropped. In this scenario, the expected PN is 1022 and the window size is 1000, so the acceptable PN number is anything greater than or equal to (expected PN - window size) = 22. So, any frame with PN < 22 is dropped.

By default, the MKA protocol default policy is enabled on an interface, if no MKA policies are applied. All the values in the policy (such as confidentiality, offset, and replay protection) take the default values. For example,

- Confidentialityoffset is 0—Encrypts the payload that is immediately after the SA (source MAC address).
- Replay protect is YES with window size 0—Frames cannot come out of order.

Configuring MACsec on an Interface

To configure MACsec on an interface with one MACsec session for voice and one for data, perform this task:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>interface interface-id</code>	Identifies the MACsec interface, and enter interface configuration mode. The interface must be a physical interface.
Step 3	<code>switchport access vlan vlan-id</code>	Configures the access VLAN for the port.
Step 4	<code>switchport mode access</code>	Configures the interface as an access port.
Step 5	<code>macsec</code>	Enables 802.1ae MACsec on the interface.
Step 6	<code>authentication event linksec fail action authorize vlan vlan-id</code>	(Optional) Specifies that the switch processes authentication link-security failures resulting from unrecognized user credentials by authorizing a restricted VLAN on the port after a failed authentication attempt.
Step 7	<code>authentication host-mode multi-domain</code>	Configures authentication manager mode on the port to allow both a host and a voice device to be authenticated on the 802.1X-authorized port. If not configured, the default host mode is single.
Step 8	<code>authentication linksec policy must-secure</code>	Sets the LinkSec security policy to secure the session with MACsec if the peer is available. If not set, the default is <i>should secure</i> .
Step 9	<code>authentication port-control auto</code>	Enables 802.1X authentication on the port. The port changes to the authorized or unauthorized state based on the authentication exchange between the switch and the client
Step 10	<code>mka policy policy-name</code>	Applies an existing MKA protocol policy to the interface, and enable MKA on the interface. If no MKA policy was configured (by entering the mka policy global configuration command), you must apply the MKA default policy to the interface by entering the mka default-policy interface configuration command.
Step 11	<code>dot1x pae authenticator</code>	Configures the port as an 802.1X port access entity (PAE) authenticator.
Step 12	<code>spanning-tree portfast</code>	Enables spanning tree Port Fast on the interface in all its associated VLANs. When Port Fast feature is enabled, the interface changes directly from a blocking state to a forwarding state without making the intermediate spanning-tree state changes.
Step 13	<code>end</code>	Returns to privileged EXEC mode.
Step 14	<code>show authentication session interface interface-id</code>	Verifies the authorized session security status.
Step 15	<code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

This is an example of configuring and verifying MACsec on an interface:

```
Switch(config)# interface GigabitEthernet1/0/25
Switch(config-if)# switchport access vlan 10
Switch(config-if)# switchport mode access
Switch(config-if)# macsec
Switch(config-if)# authentication event linksec fail action authorize vlan 2
Switch(config-if)# authentication host-mode multi-domain
Switch(config-if)# authentication linksec policy must-secure
Switch(config-if)# authentication port-control auto
```




```



Switch(config-if)# mka policy replay-policy
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# spanning-tree portfast
Switch(config-if)# end
Switch# show authentication sessions interface gigabitethernet1/0/25
Interface: GigabitEthernet1/0/25
MAC Address: 001b.2140.ec3c
IP Address: 10.1.1.103
User-Name: ms1
Status: Authz Success
Domain: DATA
Security Policy: Must Secure &--- New
Security Status: Secured &--- New
Oper host mode: multi-domain
Oper control dir: both
Authorized By: Authentication Server
Vlan Policy: 10
Session timeout: 3600s (server), Remaining: 3567s
Timeout action: Reauthenticate
Idle timeout: N/A
Common Session ID: 0A05783B00000001700448BA8
Acct Session ID: 0x00000019
Handle: 0x06000017
Runnable methods list:
Method State
dot1x Authc Success

```

Configuring MKA Pre-Shared Key

To configure MACsec Key Agreement (MKA) pre-shared key, perform this task:

	Command	Purpose
Step 1	configure terminal	Enters global configuration mode.
Step 2	key chain <i>key-chain-name</i> [macsec]	Configures a key chain and enters keychain MACsec configuration mode. <div>  Note The key ID and the key string should not be all zeros. </div>
Step 3	key <i>hex-string</i>	Configures a key and enters keychain-MACsec key configuration mode. <ul style="list-style-type: none"> The key ID must be an even-digit-sized hex-string. <div>  Note Ensure that key is a non-zero value. If key is set to 00, validation failure occurs and the following message is displayed: MKA Session bring-up failure: FSM (Derive KEK /ICK) - Zero CKN input </div>
Step 4	cryptographic-algorithm [aes-128-cmac aes-256-cmac]	Sets the cryptographic encryption algorithm. <div>  Note Cisco Catalyst 4500-X Series Switches do not support AES-256 encryption. </div>

Command	Purpose
Step 5 <code>key-string {[0 6] pre-shared-key 7 pre-shared-key}</code>	Sets the pre-shared key for a key string. <ul style="list-style-type: none"> The key-string should be a 32 or 64-digit hex-string, that is in sync with the cryptographic algorithm that is configured. <div>  Note </div> Ensure that key-string is a non-zero value. If key-string is set to 00, validation failure occurs and the following message is displayed: MKA Session bring-up failure: FSM (Derive KEK /ICK) - Zero CAK input
Step 6 <code>lifetime {local hh:mm:ss hh:mm:ss} day month year {duration seconds hh:mm:ss day month infinite}</code>	<div>  Note </div> Sets a lifetime for the MACsec key.
Step 7 <code>end</code>	Returns to privileged EXEC mode.
Step 8 <code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

This example shows how to configure MKA pre-shared key:

```
Switch# configure terminal
Switch(config)# key chain keychain1 macsec
Switch(config-keychain-macsec)# key 0001
Switch(config-keychain-macsec-key)# cryptographic-algorithm aes-128-cmac
Switch(config-keychain-macsec-key)# key-string 0 pwd
Switch(config-keychain-macsec-key)# lifetime local 16:00:00 Nov 9 2014 duration 6000
Switch(config-keychain-macsec-key)# end
Switch# copy running-config startup-config
```

Example: Connectivity Association Key Rekey

The connectivity Association Key (CAK) is a long-lived master key that is used to generate all other keys needed for MKA/MACsec.

The CAK rekey happens in the following cases:

- When moving from Key 01 to Key 02 within the Key Chain K1.
- When moving from one Key Chain K1 to another Key Chain K2.



Note

We recommend that you configure keys such that there is an overlap between the lifetime of the keys so that CAK rekey is successful and there is a seamless transition between the keys/CA (without any traffic loss or session restart.)

```
Switch# show key chain k1
```

```
Key-chain k1:
```

```
MacSEC key chain
```

```
key 01 - text "c890433a1e05ef42d723a6b58af8fdbf7a25f42b3cda6a5eeb5ae4bf3a0a679f"
lifetime (00:00:00 UTC Oct 29 2014) - (12:10:00 UTC Oct 29 2014)
key 02 - text "14d9167d538819405c0ff78c655141ed4b3c7242562c0fb0f7a56f780bf29e52"
lifetime (12:00:00 UTC Oct 29 2014) - (18:05:00 UTC Oct 29 2014)
key 03 - text "88d971cb19d9f2598ad76edc562ade2e7e91e3ed70524f5c3c4d8d9599d0670e"
lifetime (18:00:00 UTC Oct 29 2014) - (18:10:00 UTC Oct 29 2014)
```

```
key 04 - text "75474bce819b49ad7e5bd06236bc0c944c69892f71e942e2f9812b7d3a7b2a5f"
lifetime (18:10:00 UTC Oct 29 2014) - (infinite)
```

!In this case, Key 01, 02, 03 have overlapping time, but not key 04. Here is the sequence, how this works:

```
@00:00:00 - A new MKA session is Secured with key 01
@12:00:00 - CAK Rekey triggers with key 02 and upon success goes to Secured state
@18:00:00 - CAK Rekey triggers with key 03 and upon success goes to Secured state
@18:10:00 - Key 03 dies, hence MKA session using this key is brought down
@18:10:00 - Key 04 becomes active and a new MKA session is triggered with this key. Upon
success, session will be Secured and UP for infinite time.
```

Understanding MKA MACsec with EAP-TLS

Beginning in Cisco IOS XE Release 3.9.0E, MKA MACsec is supported on switch-to-switch links on Cisco Catalyst 4500-X series switches and Cisco Catalyst 4500-E series switches with Supervisor Engine 8-E. Beginning in Cisco IOS XE Release 3.10.0E, support is extended to Supervisor Engine 9-E.

Using IEEE 802.1X Port-based Authentication with Extensible Authentication Protocol (EAP-TLS), you can configure MKA MACsec between device uplink ports. EAP-TLS allows mutual authentication and obtains an MSK (master session key) from which the connectivity association key (CAK) is derived for MKA operations. Device certificates are carried, using EAP-TLS, for authentication to the AAA server.



Note

MKA MACsec is not supported on multi-point to multi-point links.

Prerequisites for MKA MACsec with EAP-TLS

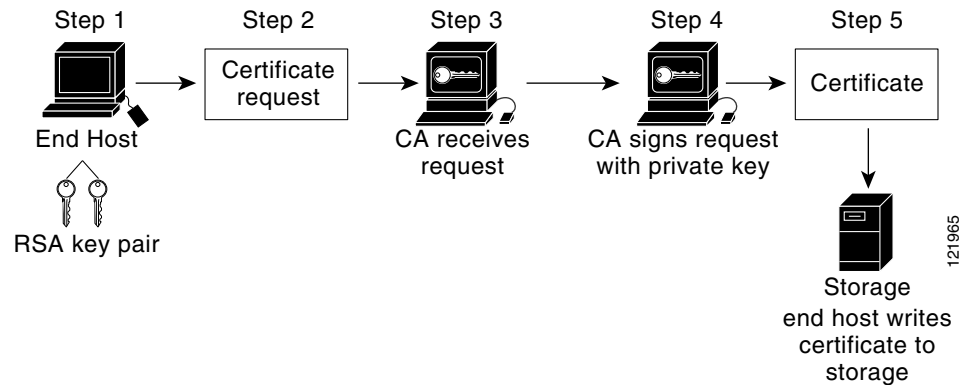
- Ensure that you have a Certificate Authority (CA) server configured for your network.
- Generate a CA certificate.
- We recommend that you configure Cisco Identity Services Engine (ISE) Release 2.0.
- Ensure that both the participating devices, the CA server, and Cisco Identity Services Engine (ISE) are synchronized using Network Time Protocol (NTP).
- Ensure that 802.1x authentication and AAA are configured on your device.

Limitations for MKA MACsec with EAP-TLS

- MKA is not supported on port-channels.
- MKA is not supported with High Availability and local authentication.
- MKA supports encryption using AES-GCM-128 encryption only. MKA does not support Security Group Tagging and AES-GCM-256 encryption.
- WS-C4510R+E chassis that runs IOS XE 3.11.xE supports only TLS 1.0. Later versions of TLS (like 1.1 or 1.2) are not supported.

Understanding Certificate Enrollment

Certificate enrollment is the process of obtaining a certificate from a Certificate Authority (CA). Each end host that wants to participate in the Cisco IOS public key infrastructure (PKI) must obtain a certificate. Certificate enrollment occurs between the end host requesting the certificate and the CA. The figure below and the following steps describe the certificate enrollment process.



1. The end host generates an RSA key pair.
2. The end host generates a certificate request and forwards it to the CA.
3. The CA receives the certificate enrollment request, and, depending on your network configuration, one of the following options occurs:
 - Manual intervention is required to approve the request.
 - The end host is configured to automatically request a certificate from the CA. Thus, operator intervention is no longer required at the time the enrollment request is sent to the CA server.



Note

If you configure the end host to automatically request certificates from the CA, you should have an additional authorization mechanism.

4. After the request is approved, the CA signs the request with its private key and returns the completed certificate to the end host.
5. The end host writes the certificate to a storage area such as NVRAM.

Generating RSA Key Pairs

To generate RSA key pairs, perform this task:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>crypto key generate rsa label <i>label</i> name <i>general-keys</i> modulus <i>size</i></code>	Generates a RSA key pair for signing and encryption. You can also assign a label to each key pair using the label keyword. The label is referenced by the trustpoint that uses the key pair. If you do not assign a label, the key pair is automatically labeled <Default-RSA-Key>. If you do not use additional keywords this command generates one general purpose RSA key pair. If the modulus is not specified, the default key modulus of 1024 is used. You can specify other modulus sizes with the modulus keyword.
Step 3	<code>end</code>	Returns to privileged EXEC mode.
Step 4	<code>show crypto key mypubkey rsa</code>	(Optional) Displays the RSA public keys of your device. This step allows you to verify that the RSA key pair has been successfully generated.
Step 5	<code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

Configuring Enrollment using SCEP

Simple Certificate Enrollment Protocol (SCEP) is a Cisco-developed enrollment protocol that uses HTTP to communicate with the certificate authority (CA) or registration authority (RA). SCEP is the most commonly used method for sending and receiving requests and certificates.

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>crypto pki trustpoint <i>server name</i></code>	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
Step 3	<code>enrollment url <i>url name pem</i></code>	Specifies the URL of the CA on which your device should send certificate requests. An IPv6 address can be added in the URL enclosed in brackets. For example: <code>http:// [2001:DB8:1:1::1]:80</code> . The pem keyword adds privacy-enhanced mail (PEM) boundaries to the certificate request.
Step 4	<code>rsa keypair <i>label</i></code>	Specifies which key pair to associate with the certificate.
Step 5	<code>serial-number none</code>	The none keyword specifies that a serial number will not be included in the certificate request.
Step 6	<code>ip-address none</code>	The none keyword specifies that no IP address should be included in the certificate request.
Step 7	<code>revocation-check <i>crl</i></code>	Specifies CRL as the method to ensure that the certificate of a peer has not been revoked.

	Command	Purpose
Step 8	<code>auto-enroll percent regenerate</code>	<p>Enables auto-enrollment, allowing the client to automatically request a rollover certificate from the CA.</p> <p>If auto-enrollment is not enabled, the client must be manually re-enrolled in your PKI upon certificate expiration.</p> <p>By default, only the Domain Name System (DNS) name of the device is included in the certificate.</p> <p>Use the percent argument to specify that a new certificate will be requested after the percentage of the lifetime of the current certificate is reached.</p> <p>Use the regenerate keyword to generate a new key for the certificate even if a named key already exists.</p> <p>If the key-pair being rolled over is exportable, the new key pair will also be exportable. The following comment will appear in the trustpoint configuration to indicate whether the key pair is exportable: “! RSA key pair associated with trustpoint is exportable.”</p> <p>It is recommended that a new key pair be generated for security reasons.</p>
Step 9	<code>crypto pki authenticate name</code>	Retrieves the CA certificate and authenticates it.
Step 10	<code>exit</code>	Exits Global Configuration mode.
Step 11	<code>show crypto pki certificate trustpoint name</code>	Displays information about the certificate for the trust point.

Configuring Manual Enrollment

If your CA does not support SCEP or if a network connection between the router and CA is not possible. Perform the following task to set up manual certificate enrollment:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>crypto pki trustpoint server name</code>	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.
Step 3	<code>enrollment terminal</code>	<p>Specifies the manual cut-and-paste certificate enrollment method.</p> <p>The certificate request will be displayed on the console terminal so that it may be manually copied (or cut).</p> <p>The pem keyword configures the trustpoint to generate PEM-formatted certificate requests to the console terminal.</p>
Step 4	<code>rsakeypair label</code>	Specifies which key pair to associate with the certificate.
Step 5	<code>serial-number none</code>	The none keyword specifies that a serial number will not be included in the certificate request.
Step 6	<code>ip-address none</code>	The none keyword specifies that no IP address should be included in the certificate request.
Step 7	<code>revocation-check crl</code>	Specifies CRL as the method to ensure that the certificate of a peer has not been revoked.

	Command	Purpose
Step 8	<code>exit</code>	Exits ca-trustpoint configuration mode and returns to global configuration mode.
Step 9	<code>crypto pki authenticate name</code>	Retrieves the CA certificate and authenticates it.
Step 10	<code>crypto pki enroll name</code>	<p>Generates certificate request and displays the request for copying and pasting into the certificate server.</p> <p>Enter enrollment information when you are prompted. For example, specify whether to include the device FQDN and IP address in the certificate request.</p> <p>You are also given the choice about displaying the certificate request to the console terminal.</p> <p>The base-64 encoded certificate with or without PEM headers as requested is displayed.</p>
Step 11	<code>crypto pki import name certificate</code>	<p>Imports a certificate via TFTP at the console terminal, which retrieves the granted certificate.</p> <p>The device attempts to retrieve the granted certificate via TFTP using the same filename used to send the request, except the extension is changed from “.req” to “.crt”. For usage key certificates, the extensions “-sign.crt” and “-encr.crt” are used.</p> <p>The device parses the received files, verifies the certificates, and inserts the certificates into the internal certificate database on the router.</p> <p>Note Some CAs ignore the usage key information in the certificate request and issue general purpose usage certificates. If your CA ignores the usage key information in the certificate request, only import the general purpose certificate. The device will not use one of the two key pairs generated.</p>
Step 12	<code>exit</code>	Exits Global Configuration mode.
Step 13	<code>show crypto pki certificate trustpoint name</code>	Displays information about the certificate for the trust point.
Step 14	<code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

Ensure that you enroll both the participating devices and the RADIUS server to the PKI infrastructure.

For more information on PKI configuration, see the [Public Key Infrastructure Configuration Guide](#).

Configuring MKA MACsec Using EAP-TLS

To configure MACsec with MKA on point-to-point links, perform these tasks:

- Configure an Authentication Policy
- Configure EAP-TLS Profiles and IEEE 802.1x Credentials
- Configure 802.1x and MKA MACsec using EAP-TLS on Interfaces

Configuring EAP-TLS and 802.1x Credentials

To configure EAP-TLS and 802.1x credentials, perform the following task:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>dot1x credentials <i>profile</i></code>	Creates 802.1x credentials profile. This must be attached to the port that is configured as supplicant.
Step 3	<code>username <i>name</i></code>	Creates a username.
Step 4	<code>password <i>password</i></code>	Creates a password.
Step 5	<code>exit</code>	Exits dot1x-creden configuration mode and returns to global configuration mode.
Step 6	<code>eap profile <i>name</i></code>	Configures the EAP profile, and enters eap-profile configuration mode.
Step 7	<code>method tls</code>	Configures the EAP-TLS method.
Step 8	<code>pki trustpoint <i>name</i></code>	Configures the default PKI trustpoint.
Step 9	<code>exit</code>	Exits eap-profile configuration mode and enters global configuration mode.
Step 10	<code>service-template <i>name</i></code>	Creates a service template and enters service template configuration mode.
Step 11	<code>linksec policy must-secure</code>	Sets a data link layer security policy. The must-secure keyword specifies that the device port must be authorized only if a secure MACsec session is established.
Step 12	<code>exit</code>	Exits service-template configuration mode and returns to global configuration mode.
Step 13	<code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

Configuring an Authentication Policy

To configure an authentication policy, perform the following task:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>exit</code>	Exits service-template configuration mode and returns to global configuration mode.
Step 3	<code>policy-map type control subscriber <i>control-policy-name</i></code>	Defines a control policy for subscriber sessions and enters control policy-map event configuration mode.
Step 4	<code>event <i>event-name</i> match-all</code>	Specifies that the session-started event triggers actions in a control policy if conditions are met. match-all is the default behavior.
Step 5	<code><i>priority-number</i> class always do-until-failure</code>	Associates a priority with an action in the control policy.
Step 6	<code><i>action-number</i> authenticate using dot1x both</code>	Initiates the authentication of a subscriber session using the IEEE 802.1x method as both a supplicant and an authenticator.
Step 7	<code>event authentication-failure match-all</code>	Specifies that the authentication-failure event triggers actions in a control policy if conditions are met. match-all is the default behavior.

	Command	Purpose
Step 8	<code>priority-number class always do-until-failure</code>	Associates a priority with an action in the control policy.
Step 9	<code>action-number terminate dot1x</code>	Terminates the authentication of a subscriber session using the IEEE 802.1x method
Step 10	<code>action-number authentication-restart seconds</code>	Sets a timer to restart the authentication process after an authentication or authorization failure.
Step 11	<code>exit</code>	Exits control policy-map event configuration mode and returns to global configuration mode.
Step 12	<code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

Applying the 802.1x and MKA MACsec Configuration on Interfaces

To apply 801.1x and MKA MACsec using EAP-TLS to interfaces, perform the following task:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>interface interface-id</code>	Identifies the MACsec interface, and enter interface configuration mode. The interface must be a physical interface.
Step 3	<code>macsec network-link</code>	Enables MKA MACsec using EAP-TLS, on the interface.
Step 4	<code>authentication periodic</code>	Enables reauthentication for this port.
Step 5	<code>authentication timer reauthenticate interval</code>	Sets the reauthentication interval.
Step 6	<code>access-session host-mode multi-host</code>	Allows hosts to gain access to the interface.
Step 7	<code>access-session closed</code>	Prevents preauthentication access on the interface.
Step 8	<code>access-session port-control auto</code>	Sets the authorization state of a port.
Step 9	<code>dot1x pae both</code>	Configures the port as an 802.1X port access entity (PAE) supplicant and authenticator.
Step 10	<code>dot1x credentials profile</code>	Assigns a 802.1x credentials profile to the interface.
Step 11	<code>dot1x supplicant eap profile name</code>	Assigns the EAP-TLS profile to the interface.
Step 12	<code>service-policy type control subscriber control-policy name</code>	Applies a subscriber control policy to the interface.
Step 13	<code>exit</code>	Returns to privileged EXEC mode.
Step 14	<code>show access-session interface interface-id</code>	(Optional) Displays the active MKA sessions for the interface, and verifies your MKA MACsec configuration.
Step 15	<code>show mka session interface interface-id</code>	
Step 16	<code>show macsec interface interface-id</code>	
Step 17	<code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

Example: MKA MACsec Switch-to-Switch Configuration

```
Switch# configure terminal
Switch(config)# crypto key generate rsa label mkaioscarsa mod 2048
The name for the keys will be: mkaioscarsa
```

```
% The key modulus size is 2048 bits
% Generating 2048 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 2 seconds)
```

```
Switch(config)# crypto pki trustpoint POLESTAR-IOS-CA
Switch(ca-trustpoint)# subject-name CN=catdevice@polestar.com, C=IN, ST=KA,
OU=ENG,O=Polestar
Switch(ca-trustpoint)# revocation-check none
Switch(ca-trustpoint)# rsakeypair mkaioscarsa
Switch(ca-trustpoint)# storage nvram:
Switch(ca-trustpoint)# end
Switch# configure terminal
Switch(config)# crypto pki authenticate POLESTAR-IOS-CA
```

```
Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself
```

```
!!PASTE THE CERTIFICATE CONTENT HERE AND END WITH ENTER!!
```

```
% Do you accept this certificate? [yes/no]: Yes
Trustpoint CA certificate accepted.
% Certificate successfully imported
Switch(config)# end
Switch# show crypto pki certificate POLESTAR-IOS-CA
```

```
CA Certificate
Status: Available
Certificate Serial Number (hex): 01
Certificate Usage: Signature
Issuer:
  cn=POLESTAR-DHCP-CM.polestar.com
  ou=ENG
Subject:
  cn=POLESTAR-DHCP-CM.polestar.com
  ou=ENG
Validity Date:
  start date: 09:39:53 IST Apr 13 2016
  end   date: 09:39:53 IST Apr 13 2017
Associated Trustpoints: POLESTAR-IOS-CA
```

```
Switch# configure terminal
Switch(config)# crypto pki enroll POLESTAR-IOS-CA
% Start certificate enrollment ..
```

```
% The subject name in the certificate will include: CN=catdevice@polestar.com, C=IN,
ST=KA, OU=ENG,O=Polestar
% The subject name in the certificate will include: Device.polestar.com
% Include the router serial number in the subject name? [yes/no]: no
% Include an IP address in the subject name? [no]: no
Display Certificate Request to terminal? [yes/no]: yes
Certificate Request follows:
```

```
MIIC8TCCAdkCAQAwgYoxETAPBgNVBAoTCFBvbGVzdGFyMQwwCgYDVQQLEWNFtKcx
CzAJBgNVBAGTAktBMQswCQYDVQQGEwJJTjEfmB0GA1UEAwWY2F0NDUwMHgxQHBv
bGVzdGFyLmNvbTEsMCoGCSqGSIb3DQEJAhYdUE9MRVNUQVItNDUwMFgtMS5wb2x1
c3Rhci5jb20wggeiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC9gJrSiouv
YRr37Wf/i0MwWKR7VreQvQwSD3/Vr9YcVJ+8bULHcaB89wqF67Tlyuwt1UgO/q9
Jr+qfP3anhvj5H0VRA10wt/tvv66LLOH8y92/yb0dEY9h+DSRpFXQbUdQkIoYbkS
DnWZnuzv1ok4yKUAbslC3uZsrlnhIQz27bc/x1E0oigQxd0PjC82eZft5EgwL081
TzmaGsTnAUzWGQyhN6U97yDt0JXCmXIuND6uUXzZp1MDiRpNYjXwSWBZHAVEAFnF
tLBrtqA46eUj2b7iywqQ5WzqzSluPnw7ATo6sprj3TpQj2hLKihqfOPs8JK86Ow+
BGrp97P05p11AgMBAAGITAfBgqhkiG9w0BCQ4xEjAQMMA4GA1UdDwEB/wQEAWIF
oDANBgqhkiG9w0BAQUFAAOCAQEAtOqEekXckNKyl+1Wjy38AY7LHIiT0amQJ0SY
abw6P+SALe+Ro6EbyS4gMPildjkDZSgaH/q9IdtmdG3GGz25CNxb0imK+2NroV+f
a0JZ6A19nqvdtz/OQ5LREcRlzfaeuNMnA2mzCpzyC0/kLQ6r040Uvz/GzPdmjWQh
sXRb57UFWffrOb11C/7SsCo7HUCG1yCiYRFTKccHbLL/0+Q7yNHapWEZ0jaKAaj6
```

```

NhKR9WNrP0onZoHIivDm44CYc3iKS96XSsz7cu4J4HLimhB36tGk6M8jPGyNl4dc
eYyh4H2RSQqJLqy2D9q01uQFecHE5D79byKvVDPd1uSyVLpExg==

Redisplay enrollment request? [yes/no]: No
Switch(config)# end

Switch# configure terminal
Switch(config)# crypto pki import POLESTAR-IOS-CA certificate

Enter the base 64 encoded certificate.
End with a blank line or the word "quit" on a line by itself

!!PASTE THE CERTIFICATE CONTENT AND END WITH ENTER!!

% Router Certificate successfully imported

Switch(config)# policy-map type control subscriber DOT1X_POLICY_RADIUS
Switch(config-event-control-policymap)# event session-started match-all
Switch(config-class-control-policymap)# 10 class always do-until-failure
Switch(config-action-control-policymap)# 10 authenticate using dot1x both
Switch(config-action-control-policymap)# event authentication-failure match-all
Switch(config-class-control-policymap)# 10 class always do-until-failure
Switch(config-action-control-policymap)# 10 terminate dot1x
Switch(config-action-control-policymap)# 20 authentication-restart 7
Switch(config-action-control-policymap)# end

Switch# configure terminal
Switch(config)# eap profile EAPTLS-PROF-IOSCA
Switch(config-eap-profile)# method tls
Switch(config-eap-profile)# pki-trustpoint POLESTAR-IOS-CA
Switch(config-eap-profile)# end

Switch# configure terminal
Switch(config)# dot1x credentials EAPTLSCRED-IOSCA
Switch(config-dot1x-creden)# username catdevice@polestar.cisco.com
Switch(config-dot1x-creden)# pki-trustpoint POLESTAR-IOS-CA
Switch(config-dot1x-creden)# end

Switch(config)# interface Tengigabitethernet 1/10
Switch(config-if)# shutdown
Switch(config-if)# macsec network-link
Switch(config-if)# authentication periodic
Switch(config-if)# authentication timer reauthenticate 43200
Switch(config-if)# access-session host-mode multi-host
Switch(config-if)# access-session closed
Switch(config-if)# access-session port-control auto
Switch(config-if)# dot1x pae both
Switch(config-if)# dot1x credentials EAPTLSCRED-IOSCA
Switch(config-if)# dot1x supplicant eap profile EAPTLS-PROF-IOSCA
Switch(config-if)# service-policy type control subscriber DOT1X_POLICY_RADIUS
Switch(config-if)# end

```

Understanding Cisco TrustSec MACsec

Table 48-2 summarizes the Cisco TrustSec features supported on the switch. For more detailed explanations, see the *Cisco TrustSec Switch Configuration Guide*:

http://www.cisco.com/en/US/docs/switches/lan/trustsec/configuration/guide/arch_over.html#wp1054561

Table 48-2 Cisco TrustSec Features

Cisco TrustSec Feature	Description
802.1AE Encryption (MACsec)	<p>Protocol for 802.1AE-based wire-rate hop-to-hop Layer 2 encryption.</p> <p>Between MACsec-capable devices, packets are encrypted on egress from the sending device, decrypted on ingress to the receiving device, and in the clear within the devices.</p> <p>This feature is only available between 802.1AE-capable devices.</p>
Network Device Admission Control (NDAC)	<p>NDAC is an authentication process by which each network device in the TrustSec domain can verify the credentials and trustworthiness of its peer device. NDAC uses an authentication framework based on IEEE 802.1X port-based authentication and uses Extensible Authentication Protocol Flexible Authentication via Secure Tunnel (EAP-FAST) as its EAP method. Authentication and authorization by NDAC results in Security Association Protocol negotiation for 802.1AE encryption.</p>
Security Association Protocol (SAP)	<p>SAP is a Cisco proprietary key exchange protocol between switches. After NDAC switch-to-switch authentication, SAP automatically negotiates keys and the cipher suite for subsequent switch-to-switch MACsec encryption between TrustSec peers. The protocol description is available under a nondisclosure agreement.</p>
Security Group Tag (SGT)	<p>An SGT is a 16-bit single label showing the security classification of a source in the TrustSec domain. It is appended to an Ethernet frame or an IP packet.</p>
Note SGT is not supported in this release.	
SGT Exchange Protocol (SXP), including SXPv2	<p>With SXP, devices that are not TrustSec-hardware capable can receive SGT attributes for authenticated users or devices from the Cisco Access Control System (ACS). The devices then forward the source IP-to-SGT binding to a TrustSec-hardware capable device for tagging and security group ACL (SGACL) enforcement.</p>

When both ends of a link support 802.1AE MACsec, SAP negotiation occurs. An EAPOL-key exchange occurs between the supplicant and the authenticator to negotiate a cipher suite, exchange security parameters, and manage keys. Successful completion of these tasks results in the establishment of a security association (SA).

Depending on your software version and licensing and link hardware support, SAP negotiation can use one of these modes of operation:

- Galois Counter Mode (GCM)—authentication and encryption
- GCM authentication (GMAC)— GCM authentication, no encryption
- No Encapsulation—no encapsulation (clear text)
- Null—encapsulation, no authentication or encryption

Cisco TrustSec uses AES-128 GCM and GMAC and is compliant with the 802.1AE standard. GCM is not supported on switches running the NPE or the LAN Base image.

Cisco TrustSec NDAC SAP is supported on trunk ports because it is intended only for network device to network device links, that is, switch-to-switch links. It is not supported on:

- Host facing access ports (these ports support MKA MACsec)
- Switch virtual interfaces (SVIs)
- SPAN destination ports

The switch also does not support security group ACLs.

You must set the Cisco TrustSec credentials to create the Cisco TrustSec network.

You can configure Cisco TrustSec link layer security in 802.1X mode or manual mode.

Configuring Cisco TrustSec MACsec

- Following topics are discussed:
- [Configuring Cisco TrustSec Credentials on the Switch, page 48-22](#)
 - [Configuring Cisco TrustSec Switch-to-Switch Link Security in 802.1X Mode, page 48-23](#)
 - [Configuring Cisco TrustSec Switch-to-Switch Link Security in Manual Mode, page 48-24](#)
 - [Cisco TrustSec Switch-to-Switch Link Security Configuration Example, page 48-25](#)



Note

The sample configuration in the last section shows the AAA and the RADIUS configuration. Use this example to configure RADIUS and AAA before configuring switch-to-switch security.

Configuring Cisco TrustSec Credentials on the Switch

To enable Cisco TrustSec features, you must create Cisco TrustSec credentials on the switch to use in other TrustSec configurations.

To configure Cisco TrustSec credentials, perform this task:

	Command	Purpose
Step 1	<code>cts credentials id device-id password cts-password</code>	Specifies the Cisco TrustSec credentials for this switch to use when authenticating with other Cisco TrustSec devices with EAP-FAST. <ul style="list-style-type: none">• id device-id—Specifies a Cisco TrustSec device ID for the switch. The device-id argument has a maximum length of 32 characters and is case sensitive.• password cts-password—Specifies the Cisco TrustSec password for the device.
Step 2	<code>show cts credentials</code>	(Optional) Displays Cisco TrustSec credentials configured on the switch.
Step 3	<code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

To delete the Cisco TrustSec credentials, enter the **clear cts credentials** privileged EXEC command.

This example shows how to create Cisco TrustSec credentials:

```
Switch# cts credentials id trustsec password mypassword
CTS device ID and password have been inserted in the local keystore. Please make
sure that the same ID and password are configured in the server database.

Switch# show cts credentials
CTS password is defined in keystore, device-id = trustsecchange-password  Initiate
password change with AAA server
```


**Note**

Before you configure Cisco TrustSec MACsec authentication, you should configure Cisco TrustSec seed and non-seed devices. For 802.1X mode, you must configure at least one seed device, that device closest to the access control system (ACS). See this section in the *Cisco TrustSec Switch Configuration Guide*:

<http://www.cisco.com/en/US/docs/switches/lan/trustsec/configuration/guide/trustsec.html>

Configuring Cisco TrustSec Switch-to-Switch Link Security in 802.1X Mode

You enable Cisco TrustSec link layer switch-to-switch security on an interface that connects to another Cisco TrustSec device. When configuring Cisco TrustSec in 802.1X mode on an interface, follow these guidelines:

- To use 802.1X mode, you must globally enable 802.1X on each device.
- If you select GCM as the SAP operating mode, you must have a MACsec encryption software license from Cisco.

**Note**

MACsec is supported on the Catalyst 4500 series switch universal k9 image. It is not supported with the NPE license or with a LAN Base service image.

If you select GCM without the required license, the interface is forced to a link-down state.

To configure Cisco TrustSec switch-to-switch link layer security with 802.1X, perform this task:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>interface interface-id</code>	Enters interface configuration mode.
Step 3	<code>cts dot1x</code>	Configures the interface to perform NDAC authentication.
Step 4	<code>sap mode-list mode1 [mode2 [mode3 [mode4]]]</code>	<p>(Optional) Configures the SAP operation mode on the interface. The interface negotiates with the peer for a mutually acceptable mode. Enter the acceptable modes in your order of preference.</p> <p>Choices for <i>mode</i> are:</p> <ul style="list-style-type: none"> • gcm-encrypt—Authentication and encryption <p>Note Select this mode for MACsec authentication and encryption if your software license supports MACsec encryption.</p> <ul style="list-style-type: none"> • gmac—Authentication, no encryption • no-encap—No encapsulation • null—Encapsulation, no authentication or encryption <p>Note If the interface is not capable of data link encryption, no-encap is the default and the only available SAP operating mode. SGT is not supported.</p>

**Note**

Although visible in the CLI help, the **timer reauthentication** and **propagate sgt** keywords are not supported. However, the **no propagate sgt** keyword is supported (refer to Step 5 in the next section).

	Command	Purpose
Step 5	exit	Exits Cisco TrustSec 802.1X interface configuration mode.
Step 6	end	Returns to privileged EXEC mode.
Step 7	show cts interface [<i>interface-id</i> / brief / summary]	(Optional) Verifies the configuration by displaying TrustSec-related interface characteristics.
Step 8	copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to enable Cisco TrustSec authentication in 802.1X mode on an interface using GCM as the preferred SAP mode:

```
Switch# configure terminal
Switch(config)# interface tengigabitethernet 1/1/2
Switch(config-if)# cts dot1x
Switch(config-if-cts-dot1x)# sap mode-list gcm-encrypt null no-encap
Switch(config-if-cts-dot1x)# exit
Switch(config-if)# end
```

Configuring Cisco TrustSec Switch-to-Switch Link Security in Manual Mode

If your switch does not have access to an authentication server or if 802.1X authentication is not needed, you can manually configure Cisco TrustSec on an interface. You must manually configure the interface on each end of the connection.

When manually configuring Cisco TrustSec on an interface, consider these usage guidelines and restrictions:

- If no SAP parameters are defined, neither encryption nor MACsec Encapsulation are performed.
- If you select GCM as the SAP operating mode, you must have a MACsec Encryption software license from Cisco. If you select GCM without the required license, the interface is forced to a link-down state.
- These protection levels are supported when you configure SAP pairwise master key (**sap pmk**):
 - SAP is not configured—no protection.
 - **sap mode-list gcm-encrypt gmac no-encap**—protection desirable but not mandatory.
 - **sap mode-list gcm-encrypt gmac**—confidentiality preferred and integrity required. The protection is selected by the supplicant according to supplicant preference.
 - **sap mode-list gmac**—integrity only.
 - **sap mode-list gcm-encrypt**—confidentiality required.
 - **sap mode-list gmac gcm-encrypt**—integrity required and preferred, confidentiality optional.

To manually configure Cisco TrustSec on an interface to another Cisco TrustSec device, perform this task:

	Command	Purpose
Step 1	configure terminal	Enters global configuration mode.
Step 2	interface <i>interface-id</i>	Enters interface configuration mode.
Step 3	cts manual	Enters Cisco TrustSec manual configuration mode.

	Command	Purpose
Step 4	<code>sap pmk key [mode-list mode1 [mode2 [mode3 [mode4]]]]</code>	<p>(Optional) Configures the SAP pairwise master key (PMK) and operation mode. SAP is disabled by default in Cisco TrustSec manual mode.</p> <ul style="list-style-type: none"> <i>key</i>—A hexadecimal value with an even number of characters and a maximum length of 32 characters. <p>The SAP operation <i>mode</i> options:</p> <ul style="list-style-type: none"> gcm-encrypt—Authentication and encryption <p>Note Select this mode for MACsec authentication and encryption if your software license supports MACsec encryption.</p> <ul style="list-style-type: none"> gmac—Authentication, no encryption no-encap—No encapsulation null—Encapsulation, no authentication or encryption <p>Note If the interface is not capable of data link encryption, no-encap is the default and the only available SAP operating mode. SGT is not supported.</p>
Step 5	<code>no propagate sgt</code>	<p>Prevents the interface from transmitting the SGT to the peer and is required in manual mode.</p> <p>Use the no form of this command when the peer is incapable of processing a SGT.</p>
Step 6	<code>exit</code>	Exits Cisco TrustSec 802.1X interface configuration mode.
Step 7	<code>end</code>	Returns to privileged EXEC mode.
Step 8	<code>show cts interface [interface-id brief summary]</code>	(Optional) Verifies the configuration by displaying TrustSec-related interface characteristics.
Step 9	<code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

This example shows how to configure Cisco TrustSec authentication in manual mode on an interface:

```
Switch# configure terminal
Switch(config)# interface tengigabitethernet 1/1/2
Switch(config-if)# cts manual
Switch(config-if-cts-manual)# sap pmk 1234abcdef mode-list gcm-encrypt null no-encap
Switch(config-if-cts-manual)# no propagate sgt
Switch(config-if-cts-manual)# exit
Switch(config-if)# end
```

Cisco TrustSec Switch-to-Switch Link Security Configuration Example

This example shows the configuration necessary for a seed and non-seed device for Cisco TrustSec switch-to-switch security. You must configure the AAA and RADIUS for link security. In this example, *ACS-1* through *ACS-3* can be any server names and *cts-radius* is the Cisco TrustSec server.

Seed Device Configuration:

```
Switch(config)# aaa new-model
Switch(config)# radius server ACS-1 address ipv4 10.5.120.12 auth-port 1812 acct-port 1813
pac key cisco123
```

```

Switch(config)# radius server ACS-2 address ipv4 10.5.120.14 auth-port 1812 acct-port 1813
pac key cisco123
Switch(config)# radius server ACS-3 address ipv4 10.5.120.15 auth-port 1812 acct-port 1813
pac key cisco123
Switch(config)# aaa group server radius cts-radius
Switch(config-sg-radius)# server name ACS-1
Switch(config-sg-radius)# server name ACS-2
Switch(config-sg-radius)# server name ACS-3
Switch(config-sg-radius)# exit
Switch(config)# aaa authentication login default none
Switch(config)# aaa authentication dot1x default group cts-radius
Switch(config)# aaa authentication network cts-radius group radius
Switch(config)# aaa session-id common
Switch(config)# cts authorization list cts-radius
Switch(config)# dot1x system-auth-control

Switch(config)# interface g1/1/2
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# cts dot1x
Switch(config-if-cts-dot1x)# sap mode-list gcm-encrypt gmac
Switch(config-if-cts-dot1x)# exit
Switch(config-if)# exit

Switch(config)# interface g1/1/4
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# cts manual
Switch(config-if-cts-dot1x)# sap pmk 033445AABBCCDDEEFF mode-list gcm-encrypt gmac
Switch(config-if-cts-dot1x)# no propagate sgt
Switch(config-if-cts-dot1x)# exit
Switch(config-if)# exit

Switch(config)# radius-server vsa send authentication
Switch(config)# end
Switch# cts credentials id cts-36 password trustsec123

```

Non-Seed Device:

```

Switch(config)# aaa new-model
Switch(config)# aaa session-id common
Switch(config)# dot1x system-auth-control

Switch(config)# interface g1/1/2
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# shutdown
Switch(config-if)# cts dot1x
Switch(config-if-cts-dot1x)# sap mode-list gcm-encrypt gmac
Switch(config-if-cts-dot1x)# exit
Switch(config-if)# exit

Switch(config)# interface g1/1/4
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# shutdown
Switch(config-if)# cts manual
Switch(config-if-cts-dot1x)# sap pmk 033445AABBCCDDEEFF mode-list gcm-encrypt gmac
Switch(config-if-cts-dot1x)# no propagate sgt
Switch(config-if-cts-dot1x)# exit
Switch(config-if)# exit
Switch(config)# radius-server vsa send authentication
Switch(config)# end

```

```
Switch# cts credentials id cts-72 password trustsec123
```




Configuring 802.1X Port-Based Authentication

This chapter describes how to configure IEEE 802.1X port-based authentication on the Catalyst 4500 series switch to prevent unauthorized client devices from gaining access to the network.

This chapter includes the following major sections:

- [About 802.1X Port-Based Authentication, page 49-1](#)
- [Configuring 802.1X Port-Based Authentication, page 49-26](#)
- [Controlling Switch Access with RADIUS, page 49-97](#)
- [Configuring Device Sensor, page 49-117](#)
- [Displaying 802.1X Statistics and Status, page 49-126](#)
- [Displaying Authentication Details, page 49-126](#)
- [Cisco IOS Security Features, page 49-131](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About 802.1X Port-Based Authentication

802.1X defines 802.1X port-based authentication as a client-server based access control and authentication protocol that restricts unauthorized clients from connecting to a LAN through publicly accessible ports. An authentication server validates each supplicant (client) connected to an authenticator (network access switch) port before making available any services offered by the switch or the LAN.



Note

802.1X support requires an authentication server that is configured for Remote Authentication Dial-In User Service (RADIUS). 802.1X authentication does not work unless the network access switch can route packets to the configured RADIUS server. To verify that the switch can route packets, you must ping the server from the switch.

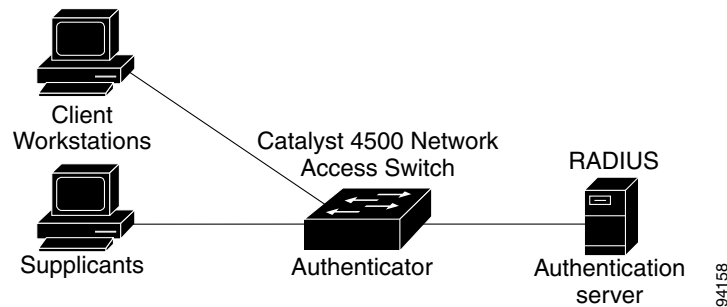
Until a client is authenticated, only Extensible Authentication Protocol over LAN (EAPOL) traffic is allowed using the port to which the client is connected. After authentication succeeds, normal traffic can pass using the port.

To configure 802.1X port-based authentication, you need to understand the concepts in these sections:

- [Device Roles, page 49-2](#)
- [802.1X and Network Access Control, page 49-3](#)
- [Authentication Initiation and Message Exchange, page 49-4](#)
- [Ports in Authorized and Unauthorized States, page 49-5](#)
- [802.1X Host Mode, page 49-6](#)
- [802.1X Violation Mode, page 49-9](#)
- [Using MAC Move, page 49-9](#)
- [Using MAC Replace, page 49-9](#)
- [Using 802.1X with VLAN Assignment, page 49-10](#)
- [Using 802.1X for Guest VLANs, page 49-11](#)
- [Using 802.1X with MAC Authentication Bypass, page 49-12](#)
- [Using 802.1X with Web-Based Authentication, page 49-14](#)
- [Using 802.1X with Inaccessible Authentication Bypass, page 49-14](#)
- [Using 802.1X with Unidirectional Controlled Port, page 49-15](#)
- [Using 802.1X with VLAN User Distribution, page 49-16](#)
- [Using 802.1X with Authentication Failed VLAN Assignment, page 49-17](#)
- [Using 802.1X with Port Security, page 49-19](#)
- [Using 802.1X Authentication with ACL Assignments and Redirect URLs, page 49-19](#)
- [Using 802.1X with RADIUS-Provided Session Timeouts, page 49-20](#)
- [Using 802.1X with Voice VLAN Ports, page 49-21](#)
- [Using Voice Aware 802.1x Security, page 49-21](#)
- [Using Multiple Domain Authentication and Multiple Authentication, page 49-22](#)
- [Limiting Login for Users, page 49-23](#)
- [802.1X Supplicant and Authenticator Switches with Network Edge Access Topology, page 49-23](#)
- [How 802.1X Fails on a Port, page 49-24](#)
- [Supported Topologies, page 49-25](#)

Device Roles

With 802.1X port-based authentication, network devices have specific roles. [Figure 49-1](#) shows the role of each device, which is described below.

Figure 49-1 802.1X Device Roles

- **Client**—The workstation that requests access to the LAN, and responds to requests from the switch. The workstation must be running 802.1X-compliant client software.
- **Authenticator**—Controls physical access to the network based on the authentication status of the client. The Catalyst 4500 series switch acts as an intermediary between the client and the authentication server, requesting identity information from the client, verifying that information with the authentication server, and relaying a response to the client. The switch encapsulates and decapsulates the Extensible Authentication Protocol (EAP) frames and interacts with the RADIUS authentication server.

When the switch receives EAPOL frames and relays them to the authentication server, the Ethernet header is stripped and the remaining EAP frame is reencapsulated in the RADIUS format. The EAP frames are not modified or examined during encapsulation, and the authentication server must support EAP within the native frame format. When the switch receives frames from the authentication server, the frame header is removed from the server, leaving the EAP frame, which is then encapsulated for Ethernet and sent to the client.

**Note**

The Catalyst 4500 series switches must be running software that supports the RADIUS client and 802.1X.

- **Authentication server**—Performs the actual authentication of the client. The authentication server validates the identity of the client and notifies the switch that the client is authorized to access the LAN and switch services. (The only supported authentication server is the RADIUS authentication server with EAP extensions; it is available in Cisco Secure Access Control Server version 3.2 and later releases.)

802.1X and Network Access Control

Network Access Control is a feature that allows port access policies to be influenced by the antivirus posture of the authenticating device.

Antivirus posture includes such elements as the operating system running on the device, the operating system version, whether antivirus software is installed and what version of antivirus signatures is available. If the authenticating device has a NAC-aware 802.1X supplicant and the authentication server is configured to support NAC using 802.1X, antivirus posture information is automatically included as part of the 802.1X authentication exchange.

For information on NAC, refer to the URL:

<http://www.cisco.com/en/US/products/ps6128/index.html>

Authentication Initiation and Message Exchange

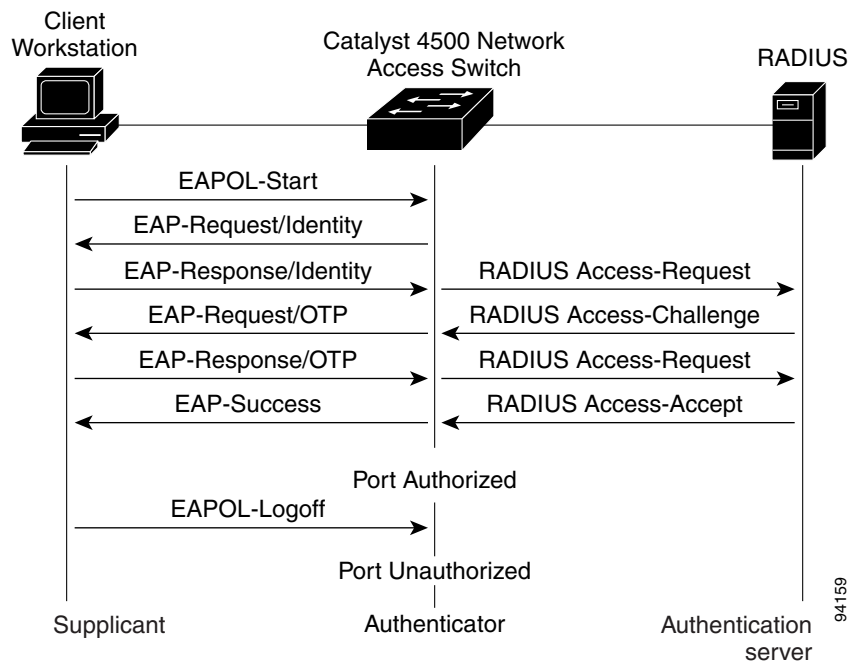
The switch or the client can initiate authentication. If you enable authentication on a port by using the **authentication port-control auto** interface configuration command (**dot1x port-control auto** command in Cisco IOS Release 12.2(46)SG and earlier releases), the switch must initiate authentication when it determines that the port link state has changed. It then sends an EAP-request/identity frame to the client to request its identity (typically, the switch sends an initial identity/request frame followed by one or more requests for authentication information). Upon receipt of the frame, the client responds with an EAP-response/identity frame.

However, if during bootup, the client does not receive an EAP-request/identity frame from the switch, the client can initiate authentication by sending an EAPOL-start frame, which prompts the switch to request the client's identity.

If 802.1X is not enabled or supported on the network access switch, any EAPOL frames from the client are dropped. If the client does not receive an EAP-request/identity frame after three attempts to start authentication, the client transmits frames as if the port is in the authorized state. A port in the authorized state means that the client was successfully authenticated. When the client supplies its identity, the switch begins its role as the intermediary, passing EAP frames between the client and the authentication server until authentication succeeds or fails. If the authentication succeeds, the switch port becomes authorized.

The specific exchange of EAP frames depends on the authentication method being used. [Figure 49-2](#) shows a message exchange that is initiated by the client using the One-Time Password (OTP) authentication method with an authentication server.

Figure 49-2 Message Exchange



Ports in Authorized and Unauthorized States

The switch port state determines whether the client is granted access to the network. The port starts in the unauthorized state. While in this state, the port disallows all ingress and egress traffic except for 802.1X protocol packets. When a client is successfully authenticated, the port transitions to the authorized state, allowing all traffic for the client to flow normally.

If a non-802.1X capable client is connected to an unauthorized 802.1X port, the switch requests the client's identity. In this situation, the client does not respond to the request, the port remains in the unauthorized state, and the client is not granted access to the network. If a guest VLAN is configured on a port that connects to a client that does not support 802.1X, the port is placed in the configured guest VLAN and in the authorized state. For more information, see the [“Using 802.1X for Guest VLANs” section on page 49-11](#).

In contrast, when an 802.1X-enabled client connects to a port that is not running the 802.1X protocol, the client initiates the authentication process by sending the EAPOL-start frame. When no response is received, the client sends the request a fixed number of times. Because no response is received, the client begins sending frames as if the port is in the authorized state.

You can control the port authorization state by using the **authentication port-control** interface configuration command (**dot1x port-control auto** command in Cisco IOS Release 12.2(46)SG and earlier releases) and these keywords:

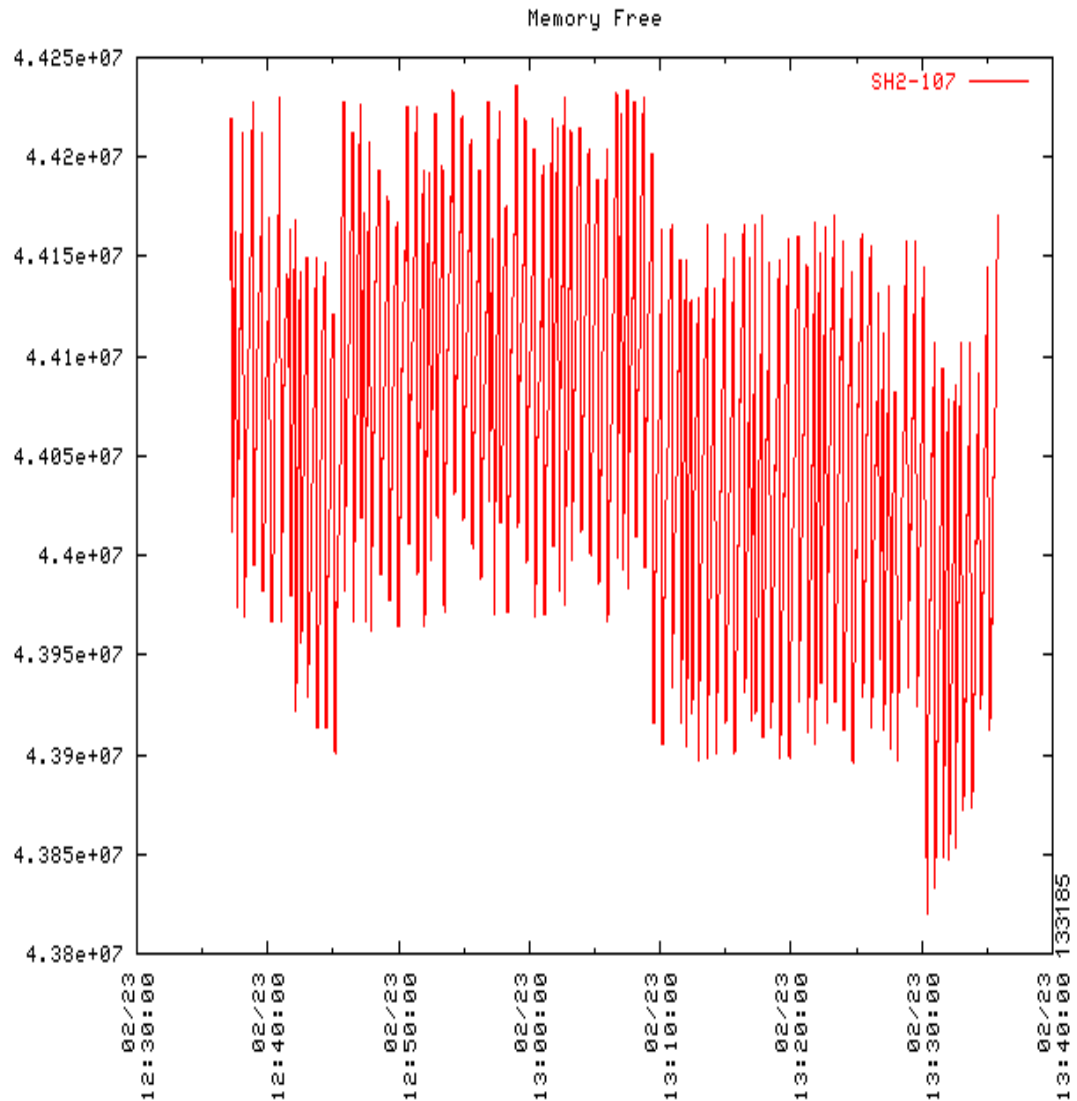
- **force-authorized**—Disables 802.1X authentication and causes the port to transition to the authorized state without requiring authentication exchange. The port transmits and receives normal traffic without 802.1X-based authentication of the client. This setting is the default.
- **force-unauthorized**—Causes the port to remain in the unauthorized state, ignoring all attempts by the client to authenticate. The switch cannot provide authentication services to the client using the interface.
- **auto**—Allows 802.1X authentication and causes the port to begin in the unauthorized state, allowing only EAPOL frames to be sent and received using the port. The authentication process begins when the link state of the port transitions from down to up or when an EAPOL-start frame is received. The switch requests the identity of the client and begins relaying authentication messages between the client and the authentication server. The switch can uniquely identify each client attempting to access the network by the client's MAC address.

If the client is successfully authenticated (receives an Accept frame from the authentication server), the port state changes to authorized, and all frames from the authenticated client are allowed using the port. If authentication fails, the port remains in the unauthorized state, but authentication can be retried. If the authentication server cannot be reached, the switch can retransmit the request. If no response is received from the server after the specified number of attempts, authentication fails and network access is not granted.

If the link state of a port transitions from up to down, or if an EAPOL-logoff frame is received by the port, the port returns to the unauthorized state.

If Multidomain Authentication (MDA) is enabled on a port, this flow can be used with some exceptions that are applicable to voice authorization. For more information on MDA, see the [“Using Multiple Domain Authentication and Multiple Authentication” section on page 49-22](#).

Figure 49-3 shows the authentication process.

Figure 49-3 Authentication Flowchart

802.1X Host Mode

The 802.1X port's host mode determines whether more than one client can be authenticated on the port and how authentication is enforced. You can configure an 802.1X port to use any of the five host modes described in the following sections. In addition, each mode can be modified to allow preauthentication open access:

- [Single-Host Mode, page 49-7](#)
- [Multiple-Hosts Mode, page 49-7](#)
- [Multidomain Authentication Mode, page 49-7](#)
- [Multiauthentication Mode, page 49-8](#)

- [Pre-authentication Open Access, page 49-8](#)

Single-Host Mode

You can configure an 802.1X port for single-host or multiple-hosts mode. In single-host mode (see [Figure 49-1 on page 49-3](#)), only one client can be connected to the 802.1X-enabled switch port. The switch detects the client by sending an EAPOL frame when the port link state changes to the up state. If a client leaves or is replaced with another client, the switch changes the port link state to down, and the port returns to the unauthorized state.

Multiple-Hosts Mode

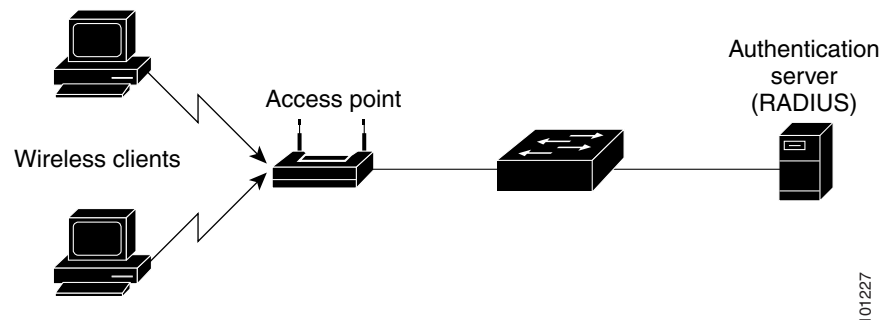
In multiple-hosts mode, you can attach multiple hosts to a single 802.1X-enabled port. [Figure 49-4 on page 49-7](#) shows 802.1X port-based authentication in a wireless LAN. In this mode, only one of the attached clients must be authorized for all clients to be granted network access. If the port becomes unauthorized (reauthentication fails or an EAPOL-logout message is received), the switch denies network access to all of the attached clients. In this topology, the wireless access point is responsible for authenticating the clients attached to it, and it also acts as a client to the switch.



Note

Wired guest access does not work on Supervisor Engine 8-E, 9-E, in multiple-host mode or in multi-authentication mode.

Figure 49-4 Multiple Host Mode Example



Multidomain Authentication Mode

Beginning with Cisco IOS Release 12.2(37)SG, Catalyst 4500 series switches support Multidomain Authentication (MDA), which allows an IP phone (Cisco or third-party) and a single host behind the IP phone to authenticate independently, using 802.1X, MAC authentication bypass (MAB) or (for the host only) web-based authentication. In this application, *multidomain* refers to two domains — data and voice — and only two MAC addresses are allowed per-port. A switch can place the host in the data VLAN and the IP phone in the voice VLAN, even though they appear on the same switch port. The data VLAN and the voice VLAN can be specified in the CLI configuration. The devices are identified as either data or voice depending on the vendor-specific-attributes (VSAs) received from the authentication, authorization, and accounting (AAA) server. The data and voice VLANs can also be obtained from the VSAs received from the (AAA) server during authentication.

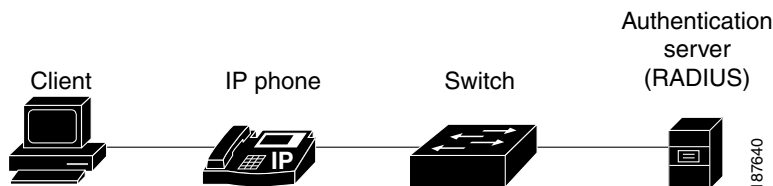
Figure 49-5 Multidomain Authentication Mode Example

Figure 49-5 shows a typical MDA application with a single host behind an IP phone connected to the 802.1X-enabled port. Because the client is not directly connected to the switch, the switch cannot detect a loss of port link if the client is disconnected. To prevent another device from using the established authentication of the disconnected client later, Cisco IP phones send a Cisco Discovery Protocol (CDP) host presence type length value (TLV) to notify the switch of changes in the attached client's port link state.

**Note**

Cisco IP phones that are connected to an authenticated port do not work in voice domain if the port is configured in either single-host mode or multi-host mode.

For details on how to configure MDA, see the [“Using Multiple Domain Authentication and Multiple Authentication”](#) section on page 49-22.

Multiauthentication Mode

Available starting in Cisco IOS Release 12.2(50)SG, multiauthentication mode allows one client on the voice VLAN and multiple authenticated clients on the data VLAN. When a hub or access point is connected to an 802.1X port, multiauthentication mode provides enhanced security over multiple-hosts mode by requiring authentication of each connected client. For non-802.1X devices, you can use MAB or web-based authentication as the fallback method for individual host authentications, allowing you to authenticate different hosts through different methods on a single port.

Multiauthentication also supports MDA functionality on the voice VLAN by assigning authenticated devices to either a data or voice VLAN depending on the VSAs received from the authentication server.

Pre-authentication Open Access

Beginning with Cisco IOS Release 12.2(50)SG, any of the four host modes can be additionally configured to allow a device to gain network access before authentication. This preauthentication open access is useful in an application such as the Pre-boot eXecution Environment (PXE), where a device must access the network to download a bootable image containing an authentication client.

Enable preauthentication open access by entering the **authentication open** command after host mode configuration. It acts as an extension to the configured host mode. For example, if preauthentication open access is enabled with single-host mode, then the port allows only one MAC address. When preauthentication open access is enabled, initial traffic on the port is restricted only by whatever other access restriction, independent of 802.1X, is configured on the port. If no access restriction other than 802.1X is configured on the port, then a client device has full access on the configured VLAN.

802.1X Violation Mode

You can use the **authentication violation** interface configuration command to configure the violation mode: restrict, shutdown, and replace.

In single-host mode, a security violation is triggered when more than one device are detected on the data vlan. In multidomain authentication mode, a security violation is triggered when more than one device are detected on the data or voice VLAN.

Security violation cannot be triggered in multiple-host mode or multiauthentication mode.

When security violation occurs, the port is protected depending on the configured violation action:

Shutdown—Errdisables the port; the default behavior on a port.

Restrict—The port state is unaffected. However the platform is notified to restrict the traffic from offending MAC-address.

Replace—Replaces existing host with the new host, instead of error-disabling or restricting the port.

For more information see [“Configuring Violation Action” section on page 49-58](#).

Using MAC Move

Hosts should be able to move across ports within a switch on the same or different VLAN without restriction, as if they had moved to a port on another switch.

Prior to Cisco IOS Release 12.2(54)SG, when a MAC address is authenticated on one switch port, that address is not allowed on another 802.1X switch port. If the switch detects that same MAC address on another 802.1X port, the address is not allowed.

Beginning with Cisco IOS Release 12.2(54)SG, you can move a MAC address to another port on the same switch. it is not pertinent for directly connected hosts or for hosts behind Cisco phones, where a port-down event or proxy EAPoL-Logoff/CDP TLV is received when the initial host disconnects. It is pertinent for hosts that disconnect from behind a hub, third party phone, or legacy Cisco phone, causing the session to remain up. With MAC move you can disconnect the host from such a device and connect it directly to another port on the same switch.

You can globally enable MAC move so that the device is reauthenticated on the new port. When a host moves to a second port, the session on the first port is deleted, and the host is reauthenticated on the new port.

MAC move is supported on all host modes. (The authenticated host can move to any port on the switch, for any host mode enabled on that port.)

For more information see [“Configuring MAC Move” section on page 49-56](#).

Using MAC Replace

Beginning with Cisco IOS Release 12.2(54)SG, you can allow new hosts to connect to abandoned ports. If the configured violation action is *replace*, the existing host is replaced by the new host, instead of err-disabling or restricting the port (as happens for single-host and MDA modes).

it is not an issue for directly connected hosts or for hosts behind Cisco phones, where a port-down event or proxy EAPoL-Logoff/CDP TLV is received when the initial host disconnects. It is an issue where a host disconnects from behind a hub, third party phone, or legacy Cisco phone, causing the session to remain up. New hosts connecting to this port violate the host-mode, triggering a violation. When the

violation action is *replace*, the NAD (switch) terminates the initial session and resets the authentication sequence based on the new MAC. This applies to single-host and MDA host modes. In multiple-auth mode, no attempt is made to remove an existing session on the same port.

For more information see the [“Configuring MAC Replace” section on page 49-57](#).

Using 802.1X with VLAN Assignment

You can use the VLAN assignment to limit network access for certain users. With the VLAN assignment, 802.1X-authenticated ports are assigned to a VLAN based on the username of the client connected to that port. The RADIUS server database maintains the username-to-VLAN mappings. After successful 802.1X authentication of the port, the RADIUS server sends the VLAN assignment to the switch. The VLAN can be a standard VLAN or a PVLAN.

On platforms that support PVLANS, you can isolate hosts by assigning ports into PVLANS.

When configured on the switch and the RADIUS server, 802.1X with VLAN assignment has these characteristics:

- If no VLAN is supplied by the RADIUS server, the port is configured in its access VLAN or isolated PVLAN when authentication succeeds.
- If the authentication server provides invalid VLAN information, the port remains unauthorized. This situation prevents ports from appearing unexpectedly in an inappropriate VLAN due to a configuration error.
- Starting with Cisco IOS Release 15.0(2)SG, if multi-authentication mode is enabled on an 802.1X port, VLAN Assignment occurs successfully for the first authenticated host. Subsequent authorized (based on user credentials) data hosts, are considered successfully authenticated, provided either they have no VLAN assignment or have a VLAN assignment matching the first successfully authenticated host on the port. This ensures that all successfully authenticated hosts on a port are members of the same VLAN. Flexibility of VLAN assignment is only provided to the first authenticated host.
- If the authentication server provides valid VLAN information, the port is authorized and placed in the specified VLAN when authentication succeeds.
- If the multiple-hosts mode is enabled, all hosts are in the same VLAN as the first authenticated user.
- If 802.1X is disabled on the port, the port is returned to the configured access VLAN.
- A port must be configured as an access port (which can be assigned only into “regular” VLANs), or as a PVLAN host port (which can be assigned only into PVLANS). Configuring a port as a PVLAN host port implies that all hosts on the port are assigned into PVLANS, whether their posture is compliant or non-compliant. If the type of the VLAN named in the Access-Accept does not match the type of VLAN expected to be assigned to the port (regular VLAN to access port, secondary PVLAN to PVLAN host port), the VLAN assignment fails.
- If a guest VLAN is configured to handle non-responsive hosts, the type of VLAN configured as the guest VLAN must match the port type (that is, guest VLANs configured on access ports must be standard VLANs, and guest VLANs configured on PVLAN host ports must be PVLANS). If the guest VLAN’s type does not match the port type, non-responsive hosts are treated as if no guest VLAN is configured (that is, they are denied network access).
- To assign a port into a PVLAN, the named VLAN must be a secondary PVLAN. The switch determines the implied primary VLAN from the locally configured secondary-primary association.

**Note**

If you change the access VLAN or PVLAN host VLAN mapping on a port that is already authorized in a RADIUS assigned VLAN, the port remains in the RADIUS assigned VLAN.

To configure VLAN assignment you need to perform these tasks:

- Enable AAA authorization by using the **network** keyword to allow interface configuration from the RADIUS server. For an illustration of how to apply the **aaa authorization network group radius** command, refer to the section “Enabling 802.1X Authentication” on page 28.
- Enable 802.1X. (The VLAN assignment feature is automatically enabled when you configure 802.1X on an access port.)
- Assign vendor-specific tunnel attributes in the RADIUS server. To ensure proper VLAN assignment, the RADIUS server must return these attributes to the switch:
 - Tunnel-Type = VLAN
 - Tunnel-Medium-Type = 802
 - Tunnel-Private-Group-ID = VLAN NAME

Using 802.1X for Guest VLANs

You can use guest VLANs to enable non-802.1X-capable hosts to access networks that use 802.1X authentication. For example, you can use guest VLANs while you are upgrading your system to support 802.1X authentication.

Guest VLANs are supported on a per-port basis, and you can use any VLAN as a guest VLAN as long as its type matches the type of the port. If a port is already forwarding on the guest VLAN and you enable 802.1X support on the network interface of the host, the port is immediately moved out of the guest VLAN and the authenticator waits for authentication to occur.

Enabling 802.1X authentication on a port starts the 802.1X protocol. If the host fails to respond to packets from the authenticator within a certain amount of time, the authenticator brings the port up in the configured guest VLAN.

If the port is configured as a PVLAN host port, the guest VLAN must be a secondary PVLAN. If the port is configured as an access port, the guest VLAN must be a regular VLAN. If the guest VLAN configured on a port is not appropriate for the type of the port, the switch behaves as if no guest VLAN is configured (that is, non-responsive hosts are denied network access).

For details on how to configure guest VLANs, see the [“Configuring 802.1X with Guest VLANs” section on page 49-58](#).

Usage Guidelines for Using 802.1X Authentication with Guest VLANs

When using 802.1X authentication with guest VLANs, consider these guidelines:

- When you reconfigure a guest VLAN to a different VLAN, any authentication failed ports are also moved and the ports stay in their current authorized state.
- When you shut down or remove a guest VLAN from the VLAN database, any authentication failed ports are immediately moved to an unauthorized state and the authentication process is restarted.

**Note**

No periodic reauthentication is allowed with guest VLANs.

Usage Guidelines for Using 802.1X Authentication with Guest VLANs on Windows-XP Hosts

When using 802.1X authentication with guest VLANs on Windows-XP hosts, consider these guidelines:

- If the host fails to respond to the authenticator, the port attempts to connect three times (with a 30 second timeout between each attempt). After this time, the login/password window does not appear on the host, so you must unplug and reconnect the network interface cable.
- Hosts responding with an incorrect login/password fail authentication. Hosts failing authentication are not put in the guest VLAN. The first time that a host fails authentication, the quiet-period timer starts, and no activity occurs for the duration of the quiet-period timer. When the quiet-period timer expires, the host is presented with the login and password window. If the host fails authentication for the second time, the quiet-period timer starts again, and no activity occurs for the duration of the quiet-period timer. The host is presented with the login and password window a third time. If the host fails authentication the third time, the port is placed in the unauthorized state, and you must disconnect and reconnect the network interface cable.

Using 802.1X with MAC Authentication Bypass

The 802.1X protocol has 3 entities: client (supplicant), authenticator, and authentication server. Typically, the host PC runs the supplicant software and tries to authenticate itself by sending its credentials to the authenticator which in turn relays that info to the authentication server for authentication.

However, not all hosts may have supplicant functionality. Devices that cannot authenticate themselves using 802.1X but still need network access can use MAC Authentication Bypass (MAB), which uses the connecting device's MAC address to grant or deny network access.

Typically, you use this feature on ports where devices such as printers are connected. Such devices do not have 802.1X supplicant functionality.

In a typical deployment, the RADIUS server maintains a database of MAC addresses that require access. When this feature detects a new MAC address on a port, it generates a RADIUS request with both username and password as the device's MAC address. After authorization succeeds, the port is accessible to the particular device using the same code path that 802.1X authentication would take when processing an 802.1X supplicant. If authentication fails, the port moves to the guest VLAN if configured, or it remains unauthorized.

The Catalyst 4500 series switch also supports reauthentication of MACs on a per-port level. Be aware that the reauthentication functionality is provided by 802.1X and is not MAB specific. In the reauthentication mode, a port stays in the previous RADIUS-sent VLAN and tries to re-authenticate itself. If the reauthentication succeeds, the port stays in the RADIUS-sent VLAN. Otherwise, the port becomes unauthorized and moves to the guest VLAN if one is configured.

For details on how to configure MAB, see the [“Configuring 802.1X with MAC Authentication Bypass” section on page 49-62](#).

Feature Interaction

This section lists feature interactions and restrictions when MAB is enabled. If a feature is not listed, assume that it interacts seamlessly with MAB (such as Unidirectional Controlled Port).

- MAB can only be enabled if 802.1X is configured on a port. MAB functions as a fall back mechanism for authorizing MACs. If you configure both MAB and 802.1X on a port, the port attempts to authenticate using 802.1X. If the host fails to respond to EAPOL requests and MAB is configured, the 802.1X port is opened up to listen to packets and to grab a MAC address, rather than attempt to authenticate endlessly.

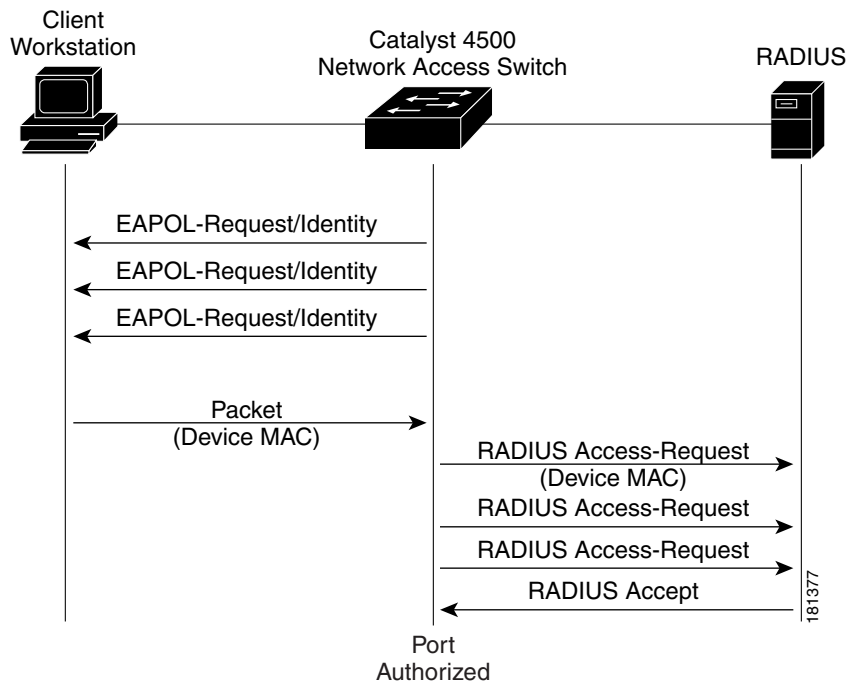
Based on the default 802.1X timer values, the transition between mechanisms takes approximately 90 seconds. You can shorten the time by reducing the value of the transmission period time, which affects the frequency of EAPOL transmission. A smaller timer value results in sending EAPOLs during a shorter time interval. With MAB enabled, after 802.1X performs one full set of EAPOLs, the learned MAC address is forwarded to the authentication server for processing.

The MAB module performs authorization for the first MAC address detected on the wire. The port is considered authorized once a valid MAC address is received that RADIUS approves of.

802.1X authentication can re-start if an EAPOL packet is received on a port that was initially authorized as a result of MAB.

Figure 49-6 shows the message exchange during MAB.

Figure 49-6 Message Exchange during MAC Authentication Bypass



- The authentication-failed VLAN is used only with dot1x-authentication-failed users. MAB is not attempted with dot1x-authentication-failed users. If 802.1X authentication fails, a port moves to the authentication-failed VLAN (if configured) whether MAB is configured or not.

- When both MAB and guest VLAN are configured and no EAPOL packets are received on a port, the 802.1X state-machine is moved to a MAB state where it opens the port to listen to traffic and grab MAC addresses. The port remains in this state forever waiting to see a MAC on the port. A detected MAC address that fails authorization causes the port to be moved to the guest VLAN if configured.

While in a guest VLAN, a port is open to all traffic on the specified guest VLAN. Non-802.1X supplicants that normally would be authorized but are in guest VLAN due to the earlier detection of a device that failed authorization, would remain in the guest VLAN indefinitely. However, loss of link or the detection of an EAPOL on the wire causes a transition out of the guest VLAN and back to the default 802.1X mode.

- Catalyst 4500 series switch supports MAB with VVID, with the restriction that the MAC address appears on a port data VLAN only. All IP phone MACs learned using CDP are allowed on voice VLANs.
- MAB and VMPS are mutually exclusive because their functionality overlaps.

Using 802.1X with Web-Based Authentication

The web-based authentication feature, known as Web Authentication Proxy, allows you to authenticate end users on host systems that do not run the IEEE 802.1X supplicant.

When configuring web-based authentication, consider these guidelines:

- Fallback to web-based authentication is configured on switch ports in access mode. Ports in trunk mode are not supported.
- Fallback to web-based authentication is not supported on EtherChannels or EtherChannel members.
- Although fallback to web-based authentication is an interface-specific configuration, the web-based authentication fallback behavior is defined in a global fallback profile. If the global fallback configuration changes, the new profile is not used until the next instance of authentication fallback.

For detailed information on configuring web-based authentication, see [Chapter 53, “Configuring Web-Based Authentication.”](#)

Using 802.1X with Inaccessible Authentication Bypass

When a switch cannot reach the configured RADIUS servers and clients (supplicants) cannot be authenticated, you can configure a switch to allow network access to hosts connected to *critical* ports that are enabled for Inaccessible Authentication Bypass.

When Inaccessible Authentication Bypass is enabled, a switch monitors the status of the configured RADIUS servers. If no RADIUS servers are available, clients that fail authentication due to server unavailability are authorized. Inaccessible Authentication Bypass can be enabled for data clients and voice clients. For data clients, you can specify an Inaccessible Authentication Bypass VLAN on a per-port basis. For voice clients they are authorized in the configured voice vlan. Inaccessible Authentication Bypass for voice clients can activate in Multiple Domain Authentication and Multiple Authentication modes, in which authentication is enforced for voice devices.



Note

Inaccessible Authentication Bypass allows a voice client to access configured voice VLAN when RADIUS becomes unavailable. For the voice device to operate properly, it must learn the voice VLAN ID through other protocols such as CDP, LLDP, or DHCP, wherever appropriate. When a RADIUS server is unavailable, it may not be possible for a switch to recognize a MAC address as that of a voice device.

Therefore, when Inaccessible Authentication Bypass is configured for voice devices, it should also be configured for data. Voice devices may be authorized on both critical data and voice VLANs. If port security is enabled, this may affect the maximum port security entries enforced on the port.

By default, data clients that were already authorized when RADIUS becomes unavailable are unaffected by Inaccessible Authentication Bypass. To reauthenticate all authorized data clients on the port when RADIUS becomes unavailable, use the **authentication server dead action reinitialize vlan** interface configuration command. This command is intended for multiauthentication mode and is mutually exclusive with the **authentication server dead action authorize vlan** command.

**Note**

In multiauthentication mode, you cannot use the **authentication server dead action authorize vlan** command to enable Inaccessible Authentication Bypass for data clients; it has no effect. Instead, use the **authentication server dead action reinitialize vlan** *vlan-id* command.

When RADIUS becomes available, critically authorized ports can be configured to automatically reauthenticate themselves.

**Note**

To properly detect RADIUS server availability, the **test username** *name* option should be enabled in the **radius server host** command. For details on how to configure RADIUS server, see the [“Configuring Switch-to-RADIUS-Server Communication”](#) section on page 49-32.

Inaccessible Authentication Bypass cannot activate after a port falls back to Web-based authentication. For details on how to configure Web-based authentication, see [Chapter 53, “Configuring Web-Based Authentication.”](#)

For details on how to configure Inaccessible Authentication Bypass, see [Chapter 53, “Configuring Web-Based Authentication”](#).

Using 802.1X with Unidirectional Controlled Port

Unidirectional Controlled Port is a combined hardware and software feature that allows dormant PCs to be powered on based on the receipt of a specific Ethernet frame, known as the *magic packet*. Generally, Unidirectional Controlled Port is used in environments where administrators plan to manage remote systems during off-hours, when the systems usually have been powered down.

Use of Unidirectional Controlled Port with hosts attached through 802.1X ports presents a unique problem: when the host powers down, a 802.1X port becomes unauthorized. In this state, the port allows the receipt and transmission of EAPoL packets only. The Unidirectional Controlled Port magic packet cannot reach the host; without powering up, the PC cannot authenticate and open the port.

Unidirectional Controlled Port solves this problem by allowing packets to be transmitted on unauthorized 802.1X ports.

**Note**

Unidirectional Controlled Port only works when Spanning Tree PortFast is enabled on the port.

For details on how to configure 802.1X with Unidirectional Controlled Port, see the [“Configuring 802.1X with Unidirectional Controlled Port”](#) section on page 49-68.

Unidirectional State

A unidirectional controlled port is typically configured when a connected host might enter a sleeping mode or power-down state. When either occurs, the host does not exchange traffic with other devices in the network. A host connected to the unidirectional port cannot send traffic to the network; it can only receive traffic from other devices in the network.

When you configure a port as unidirectional (with the **authentication control-direction in** interface configuration command), the port will receive traffic in VLANs on that port, but it is not put into a spanning-tree forwarding state. If a VLAN contains only unauthenticated ports, any SVI on that VLAN will be in a down state, during which packets will not be routed into the VLAN. For the SVI to be up, and so enable packets to be routed into the VLAN, at least one port in the VLAN must either be authenticated or in the spanning-tree forwarding state.

Bidirectional State

When you configure a port as bidirectional by using the **authentication control-direction both** interface configuration command (or the **dot1x control-direction both** interface configuration command for Cisco IOS Release 12.2(46) or earlier), the port is access-controlled in both directions. In this state, except for EAPOL packets, a switch port does not receive or send packets.

Using 802.1X with VLAN User Distribution

An alternative to dynamically assigning a VLAN ID or a VLAN name is to assign a VLAN group name. The 802.1X VLAN User Distribution feature allows you to distribute users belonging to the same group (and characterized by a common VLAN group name) across multiple VLANs. You usually do this to avoid creating an overly large broadcast domain.

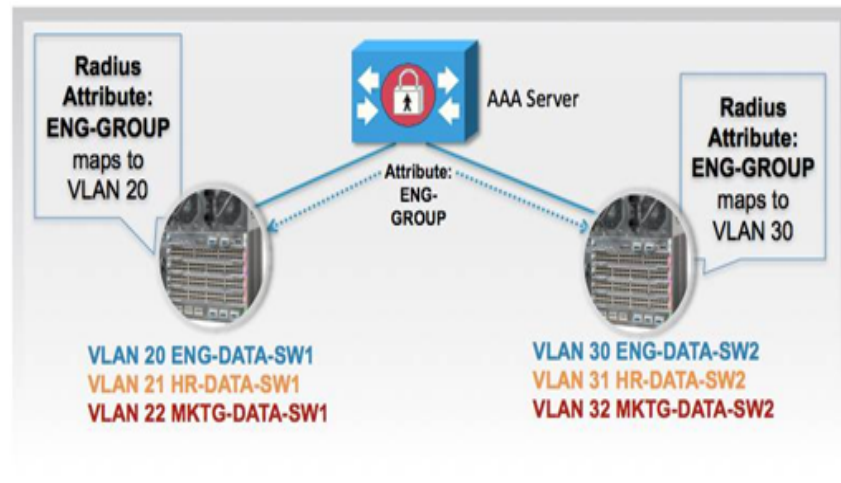
For example, with this feature, you can download a common VLAN group name (similar to ENG-Group, for all the users belonging to the engineering organization) from the authentication server to all the access-layer switches. The VLAN group name is then individually mapped to a different VLAN on each access-layer switch. The same VLAN number need not be spanned across separate switches. Similarly, the VLANs does not need to be renamed at the edge devices.

When the authentication server returns more than one VLAN group name or VLANs, this feature attempts to distribute users evenly across those groups. It internally maintains the count of users assigned to each VLAN on that switch by authentication or port security. Based on this information, this feature assigns a newly authenticated user to the least loaded VLAN on that switch among all the VLANs or VLAN group names obtained from the RADIUS server.

This VLAN distribution considers the load of all the valid VLANs only during initial user authentication, and not during reassignment. When some of the existing authenticated users are removed, the feature does not attempt to redistribute the remaining authenticated users. Group distribution does not guarantee perfect load distribution all the time.

Deployment Example

In a large campus LAN design, you might want to design the VLAN infrastructure without large Layer 2 domain. For the same employee VLAN, customers might have different VLANs at different campus access switches. When you deploy 802.1X with VLAN assignment, it does not assign one employee VLAN to all employees. You have to know the real VLANs configured on the switch. User distribution allows you to send a list of VLAN or VLAN group name(s) to the switch. Your switch can then do a local mapping to the corresponding VLAN. ([Figure 49-7](#)).

Figure 49-7 802.1X with VLAN User Distribution

For details on how to configure VLAN User Distribution, see the “[Configuring 802.1X with VLAN User Distribution](#)” section on page 49-69.

Using 802.1X with Authentication Failed VLAN Assignment

You can use authentication-failed VLAN assignment on a per-port basis to provide access for authentication failed users. Authentication failed users are end hosts that are 802.1X- capable but do not have valid credentials in an authentication server or end hosts that do not give any username and password combination in the authentication pop-up window on the user side.

If a user fails the authentication process, that port is placed in the authentication-failed VLAN. The port remains in the authentication-failed VLAN until the reauthentication timer expires. When the reauthentication timer expires the switch starts sending the port reauthentication requests. If the port fails reauthentication it remains in the authentication-failed VLAN. If the port is successfully reauthenticated, the port is moved either to the VLAN sent by RADIUS server or to the newly authenticated ports configured VLAN; the location depends on whether RADIUS is configured to send VLAN information.



Note

When enabling periodic reauthentication (see the “[Enabling Periodic Reauthentication](#)” section on page 49-83), only local reauthentication timer values are allowed. You cannot use a RADIUS server to assign the reauthentication timer value.

You can set the maximum number of authentication attempts that the authenticator sends before moving a port into the authentication-failed VLAN. The authenticator keeps a count of the failed authentication attempts for each port. A failed authentication attempt is either an empty response or an EAP failure. The authenticator tracks any mix of failed authentication attempts towards the authentication attempt count. After the maximum number of attempts is reached the port is placed in the authentication-failed VLAN until the reauthentication timer expires again.



Note

RADIUS can send a response without an EAP packet in it when it does not support EAP, and sometimes third-party RADIUS servers also send empty responses. When this behavior occurs, the authentication attempt counter is incremented.

For details on how to configure Authentication Failed VLAN Assignment, see the [“Configuring 802.1X with Authentication Failed” section on page 49-72](#).

Usage Guidelines for Using Authentication Failed VLAN Assignment

Usage guidelines include the following:

- You should enable reauthentication. The ports in authentication-failed VLANs do not receive reauthentication attempts if reauthentication is disabled. To start the reauthentication process the authentication-failed VLAN must receive a link-down event or an EAP logoff event from the port. If the host is behind a hub, you may never get a link-down event and may not detect the new host until the next reauthentication occurs.
- EAP failure messages are not sent to the user. If the user fails authentication the port is moved to an authentication-failed VLAN and a EAP success message is sent to the user. Because the user is not notified of the authentication failure there may be confusion as to why there is restricted access to the network. A EAP Success message is sent for the following reasons:
 - If the EAP Success message is not sent, the user tries to authenticate every 60 seconds (by default) by sending an EAP-start message.
 - In some cases, users have configured DHCP to EAP-Success and unless the user sees a success, DHCP does not work on the port.
- Sometimes a user caches an incorrect username and password combination after receiving a EAP success message from the authenticator and reuses that information in every reauthentication. Until the user passes the correct username and password combination the port remains in the authentication-failed VLAN.
- When an authentication failed port is moved to an unauthorized state the authentication process is restarted. If you should fail the authentication process again the authenticator waits in the held state. After you have correctly reauthenticated all 802.1X ports are reinitialized and treated as normal 802.1X ports.
- When you reconfigure an authentication-failed VLAN to a different VLAN, any authentication failed ports are also moved and the ports stay in their current authorized state.
- When you shut down or remove an authentication-failed VLAN from the VLAN database, any authentication failed ports are immediately moved to an unauthorized state and the authentication process is restarted. The authenticator does not wait in a held state because the authentication-failed VLAN configuration still exists. While the authentication-failed VLAN is inactive, all authentication attempts are counted, and as soon as the VLAN becomes active the port is placed in the authentication-failed VLAN.
- If you reconfigure the maximum number of authentication failures allowed by the VLAN, the change takes affect after the reauthentication timer expires.
- Internal VLANs that are used for Layer 3 ports cannot be configured as authentication-failed VLANs.
- The authentication-failed VLAN is supported only in single-host mode (the default port mode).
- When a port is placed in an authentication-failed VLAN the user's MAC address is added to the mac-address-table. If a new MAC address appears on the port, it is treated as a security violation.
- When an authentication failed port is moved to an authentication-failed VLAN, the Catalyst 4500 series switch does not transmit a RADIUS-Account Start Message as it does for standard 802.1X authentication.

Using 802.1X with Port Security

We do not recommend enabling port security when IEEE 802.1x is enabled. IEEE 802.1x enforces a single MAC address per port (or per VLAN when MDA is configured for IP telephony). Therefore port security is redundant and in some cases may interfere with the expected IEEE 802.1x operations.

Using 802.1X Authentication with ACL Assignments and Redirect URLs

Beginning with Cisco IOS Release 12.2(50)SG, you can download per-host policies such as ACLs and redirect URLs to the switch from the RADIUS server during 802.1X or MAB authentication of the host. ACL download is also supported with web authentication after a fallback from 802.1X or MAB.

When the 802.1X host mode of the port is either single-host, MDA, or multiple authentication, the downloaded ACLs (DACLS) are modified to use the authenticated hosts' IP address as the source address. When the host mode is multiple-hosts, the source address is configured as ANY, and the downloaded ACLs or redirects apply to all devices on the port.

If no ACLs are provided during the authentication of a host, the static default ACL configured on the port is applied to the host. On a voice VLAN port, only the static default ACL of the port is applied to the phone.

This section includes these topics:

- [Cisco Secure ACS and AV Pairs for URL-Redirect, page 49-19](#)
- [ACLs, page 49-20](#)

For details on how to configure downloadable ACL and URL redirect, refer to the [“Configuring 802.1X Authentication with ACL Assignments and Redirect URLs” section on page 49-37](#).

Cisco Secure ACS and AV Pairs for URL-Redirect

When downloadable ACL is enabled, Cisco Secure ACS provides AAA services through RADIUS.

You can set these Attribute-Value (AV) pairs on the Cisco Secure ACS with RADIUS *cisco-av-pair* vendor-specific attributes (VSAs):

- CiscoSecure-Defined-ACL specifies the names of the DACLS on the Cisco Secure ACS. The switch receives the ACL name using the CiscoSecure-Defined-ACL AV pair in the format:

#ACL#-IP-name-number

name is the ACL name and *number* is the version number (similar to 3f783768).

The Auth-Manager code verifies whether the access control entries (ACEs) of the specified downloadable ACL were previously downloaded. If not, the Auth-Manager code sends an AAA request with the downloadable ACL name as the username so that the ACEs are downloaded. The downloadable ACL is then created as a named ACL on the switch. This ACL has ACEs with a source address of any and does not have an implicit deny statement at the end. When the downloadable ACL is applied to an interface after authentication completes, the source address changes from any to the host source IP address depending on the host mode of the interface. The ACEs are prepended to the downloadable ACL applied to the switch interface to which the endpoint device is connected. If traffic matches the CiscoSecure-Defined-ACL ACEs, the appropriate actions are taken.

- url-redirect and url-redirect-acl specify the local URL policy on the switch. The switches use these cisco-av-pair VSAs as follows:

- url-redirect = <HTTP or HTTPS URL>
- url-redirect-acl = switch ACL name or number

These AV pairs enable the switch to intercept an HTTP or HTTPS request from the endpoint device and forward the client web browser to the specified redirect address from which the latest antivirus files can be downloaded. The url-redirect AV pair on the Cisco Secure ACS contains the URL to which the web browser is redirected. The url-redirect-acl AV pair contains the name or number of an ACL that specifies the HTTP or HTTPS traffic to be redirected. Traffic that matches a permit entry in the redirect ACL is redirected.

**Note**

The redirect or default ACL must be defined on the switch.

When redirect ACLs are used, we recommend that you configure a dynamic ACL that has an explicit permit statement for the IP address to which the traffic should be redirected.

ACLs

If downloadable ACL is configured for a particular client on the authentication server, you must configure a default port ACL on a client-facing switch port.

If the default ACL is configured on the switch and the Cisco Secure ACS sends a host access policy to the switch, it applies the policy to traffic from the host connected to a switch port. If the policy does not apply, the switch applies the default ACL. If the Cisco Secure ACS sends the switch a downloadable ACL, this ACL takes precedence over the default ACL already configured on the switch port. However, if the switch receives a host access policy from the Cisco Secure ACS, but the default ACL is not configured, the authorization failure is declared.

For details on how to configure a downloadable policy, refer to the [“Configuring a Downloadable Policy” section on page 49-45](#).

Using 802.1X with RADIUS-Provided Session Timeouts

You can specify whether a switch uses a locally configured or a RADIUS-provided reauthentication timeout. If the switch is configured to use the local timeout, it reauthenticates the host when the timer expires.

If the switch is configured to use the RADIUS-provided timeout, it scans the RADIUS Access-Accept message for the Session-Timeout and optional Termination-Action attributes. The switch uses the value of the Session-Timeout attribute to determine the duration of the session, and it uses the value of the Termination-Action attribute to determine the switch action when the session's timer expires.

If the Termination-Action attribute is present and its value is RADIUS-Request, the switch reauthenticates the host. If the Termination-Action attribute is not present, or its value is Default, the switch terminates the session.

**Note**

The supplicant on the port detects that its session was terminated and attempts to initiate a new session. Unless the authentication server treats this new session differently, the client may see only a brief interruption in network connectivity as the switch sets up a new session.

If the switch is configured to use the RADIUS-supplied timeout, but the Access-Accept message does not include a Session-Timeout attribute, the switch never reauthenticates the supplicant. This behavior is consistent with Cisco's wireless access points.

For details on how to configure RADIUS-provided session timeouts, see the [“Configuring RADIUS-Provided Session Timeouts” section on page 49-55](#).

Using 802.1X with Voice VLAN Ports

A voice VLAN port is a special access port associated with two VLAN identifiers:

- Voice VLAN ID (VVID) to carry voice traffic to and from the IP phone. The VVID is used to configure the IP phone connected to the port.
- Port VLAN ID (PVID) to carry the data traffic to and from the workstation connected to the switch using the IP phone. The PVID is the native VLAN of the port.

Each port that you configure for a voice VLAN is associated with a VVID and a PVID. This configuration allows voice traffic and data traffic to be separated onto different VLANs.

A voice VLAN port becomes active when a link exists whether the port is AUTHORIZED or UNAUTHORIZED. All traffic exiting the voice VLAN is obtained correctly and appears in the MAC address table. Cisco IP phones do not relay CDP messages from other devices. If several Cisco IP phones are connected in a series, the switch recognizes only the one directly connected to it. When 802.1X is enabled on a voice VLAN port, the switch drops packets from unrecognized Cisco IP phones more than one hop away.

When 802.1X is enabled on a port, you cannot configure a PVID that is equal to a VVID. For more information about voice VLANs, see [Chapter 46, “Configuring Voice Interfaces.”](#)

Observe the following feature interactions:

- 802.1X VLAN assignment cannot assign to the port the same VLAN as the voice VLAN; otherwise, the 802.1X authentication fails. The same holds true for dynamic VLAN assignment.
- 802.1X guest VLAN works with the 802.1X voice VLAN port feature. However, the guest VLAN cannot be the same as the voice VLAN.
- You cannot use the 802.1X voice VLAN port feature with 802.1X port security’s sticky MAC address configuration and statically configured MAC address configuration.
- 802.1X accounting is unaffected by the 802.1X voice VLAN port feature.
- When 802.1X is configured on a port, you cannot connect multiple IP phones to a Catalyst 4500 series switch through a hub.
- Because voice VLANs cannot be configured as PVLAN host ports, and because only PVLANs can be assigned to PVLAN host ports, VLAN assignment cannot assign a PVLAN to a port with a voice VLAN configured.

For details on how to configure 802.1X with voice VLANs, see the [“Configuring 802.1X with Voice VLAN” section on page 49-74](#).

Using Voice Aware 802.1x Security

You use the voice aware 802.1x security feature to configure the switch to disable only the VLAN on which a security violation occurs, whether it is a data or voice VLAN. In previous releases, when an attempt to authenticate the data client caused a security violation, the entire port shut down, resulting in a complete loss of connectivity.

You can use this feature in IP phone deployments where a PC is connected to the IP phone. A security violation found on the data VLAN results in the shutdown of only the data VLAN. The traffic on the voice VLAN flows through the switch without interruption.

For information on configuring voice aware 802.1x security, see the [“Configuring Voice Aware 802.1x Security” section on page 49-75](#)

Using Multiple Domain Authentication and Multiple Authentication

Multiple Domain Authentication (MDA) allows both a data device and a voice device, such as an IP phone (Cisco or third party non-Cisco), to authenticate on the same switch port, which is divided into a data domain and a voice domain.

Multi Auth allows multiple data devices and a voice device. When a voice VLAN is configured on a multiple- authentication port, the port can perform authentication in the voice domain as on an MDA port.

MDA does not enforce the order of device authentication. For best results, however, you should authenticate a voice device before you authenticate a data device on an MDA-enabled port.

When configuring MDA, consider the following guidelines.



Note

The same guidelines also apply for Multiple Authentication when voice VLAN is configured.

- We recommend that you enable CoPP on an MDA-enabled port to protect against a DoS attack. Refer to [Chapter 57, “Configuring Control Plane Policing and Layer 2 Control Packet QoS.”](#)
- To configure a switch port for MDA or Multiple Authentication, see the [“Configuring Multiple Domain Authentication and Multiple Authorization” section on page 49-33](#).
- You must configure the voice VLAN for the IP phone when the host mode is set to multidomain. For more information, see [Chapter 46, “Configuring Voice Interfaces.”](#)
- To authorize a voice device, the AAA server must be configured to send a Cisco Attribute-Value (AV) pair attribute with a value of device-traffic-class=voice. Without this value, the switch treats the voice device as a data device.
- You must configure the attribute device-traffic-class=voice on all authenticated phones. If not configured, authenticated phones may not work correctly.
- The guest VLAN and restricted VLAN features only apply to the data devices on an MDA-enabled port. The switch treats a voice device that fails authorization as a data device.
- If more than one device attempts authorization on either the voice or the data domain of a port, it is error-disabled.
- Until a device is authorized, the port drops its traffic. Non-Cisco IP phones or voice devices are allowed into both the data and voice VLANs. The data VLAN allows the voice device to contact a DHCP server to obtain an IP address and acquire the voice VLAN information. After the voice device starts sending on the voice VLAN, its access to the data VLAN is blocked. A security violation may occur in MDA if the voice device continues to send traffic on the data VLAN.
- MDA can use MAC authentication bypass as a fallback mechanism to allow the switch port to connect to devices that do not support 802.1X authentication. It is especially useful for third party phones without 802.1X supplicant. For more information, see the [“Using 802.1X with MAC Authentication Bypass” section on page 49-12](#).
- When a data or a **voice** device is detected on a port, its MAC address is blocked until authorization succeeds. If the authorization fails, the MAC address remains blocked for 5 minutes.
- If more than one device is detected on the data VLAN or more than one voice device is detected on the voice VLAN while a port is unauthorized, the port is error-disabled.

- When a port host mode is changed from single- or multihost to multidomain mode, an authorized data device remains authorized on the port. However, a Cisco IP phone that was allowed on the port in the voice VLAN is automatically removed and must be reauthenticated on that port.
- Active fallback mechanisms such as guest VLAN and restricted VLAN remain configured after a port changes from single- or multihost mode to multidomain mode.
- Switching a port host mode from multidomain to single- or multihost mode removes all authorized devices from the port.
- If a data domain is authorized first and placed in the guest VLAN, non-802.1X-capable voice devices need to tag their packets on the voice VLAN to trigger authentication.
- We do not recommend per-user ACLs with an MDA-enabled port. An authorized device with a per-user ACL policy might impact traffic on both the voice and data VLANs of the port. If used, only one device on the port should enforce per-user ACLs.

**Note**

Multi-Authentication per user VLAN is not supported on Catalyst 4500 Series Switch.

Limiting Login for Users

The Limiting Login feature helps Network administrators to limit the login attempt of users to a network. When a user fails to successfully login to a network within a configurable number of attempts within a configurable time limit, the user can be blocked. This feature is enabled only for local users and not for remote users. You need to configure the **aaa authentication rejected** command in global configuration mode to enable this feature.

802.1X Supplicant and Authenticator Switches with Network Edge Access Topology

The Network Edge Access Topology (NEAT) feature extends identity to areas outside the wiring closet (such as conference rooms).

You can enable any authentication host mode on the authenticator switch interface that connects to a supplicant switch. Once the supplicant switch authenticates successfully, the port mode changes from access to trunk. To ensure that NEAT works on all host modes, use the **dot1x supplicant force-multicast** global configuration command on the supplicant switch. If the access VLAN is configured on the authenticator switch, it becomes the native VLAN for the trunk port after successful authentication.

**Note**

MAB is not supported or recommended for use with NEAT. Only use 802.1X to authenticate the supplicant switch.

**Note**

The Catalyst 4500 series switch only supports authenticator ports.

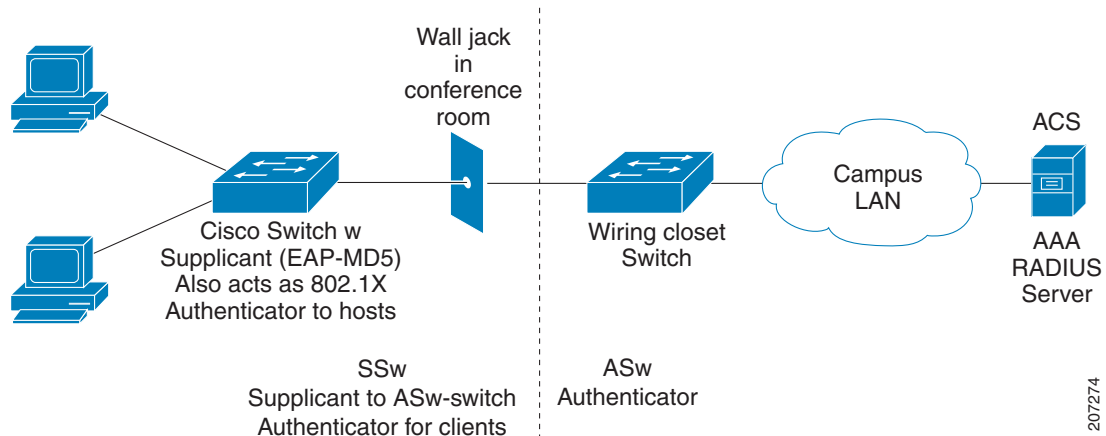
Deployment

NEAT is intended for deployment scenarios where a switch acting as 802.1X authenticator to end-hosts (PC or Cisco IP-phones) is placed in an unsecured location (outside wiring closet).

Because of this topology, the authenticator switch cannot always be trusted. For example, compact switches (8-port Catalyst 3560 and Catalyst 2960) are generally deployed outside the wiring closet. This enables hacker devices to swamp them to gain access to the network, compromising security. An edge switch must be able to authenticate itself against another switch, referred to as Network Edge Authentication Topology (NEAT).

Figure 49-8 illustrates a typical NEAT topology.

Figure 49-8 Typical NEAT Topology



NEAT facilitates the following functionality in such scenarios:

Host Authorization— Ensures that only traffic from authorized hosts (connecting to the switch with a supplicant) is allowed on the network. The switches use Client Information Signalling Protocol (CISP) to send the MAC addresses connecting the supplicant switch to the authenticator switch.

Auto enablement—Automatically enables trunk configuration on the authenticator switch, allowing user traffic from multiple VLANs arising from supplicant switches. At the ACS, you must configure the Cisco AV pair as device-traffic-class=switch. For details on how to do this, see the [“Configuring an Authenticator and a Supplicant Switch with NEAT”](#) section on page 49-89.

How 802.1X Fails on a Port

802.1X may fail on a port in three ways: timeout, explicit failure, and protocol timeout.

Timeout—A switch attempts 802.1X at link up but the attached endpoint is not 802.1X-capable. After the configured number of retries and timeouts, the switch attempts the next authentication method if one is configured (like MAB). If MAB fails, the switch deploys the Guest VLAN (also called the no-response VLAN), if configured. The Guest VLAN is configured with the **authentication event no-response** interface command.

Explicit Failure—A switch and the endpoint perform the entire 802.1X authentication sequence and the result is an explicit failure (usually indicated by an Access-Reject from the RADIUS server to the switch and an EAP-Failure sent from the switch to the endpoint). In this case, the switch attempts MAB (if “authentication event failure action next-method” is configured) or deploy the AuthFail VLAN (if “authentication event failure action authorize vlan” is configured).

Protocol Timeout—A switch and the endpoint start the 802.1X authentication process but do not complete it. For example, the endpoint may send an 802.1X EAPoL-Start message and then stop responding to the switch (perhaps, because the endpoint lacks a credential or because it is waiting for end user to enter some information). In this case, the switch knows that the connected device is EAPoL-capable, so it will not deploy the Guest VLAN after timing out. Instead, it restarts authentication after a timeout. The switch continues to label the port as EAPoL-capable until a physical link down event is detected. To force the switch to deploy the Guest VLAN in the case of a protocol timeout, configure **dot1x guest-vlan supplicant** globally. If the port is configured for hostmode multi-domain authentication, the switch behaves as if **dot1x guest-vlan supplicant** is configured.

Supported Topologies

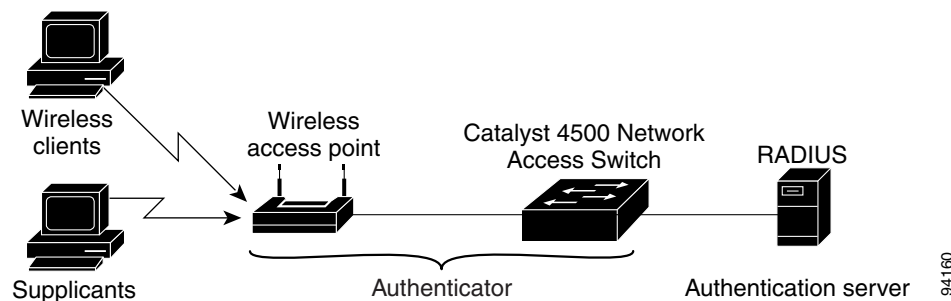
The 802.1X port-based authentication supports two topologies:

- Point-to-point
- Wireless LAN

In a point-to-point configuration (see [Figure 49-1 on page 49-3](#)), only one client can be connected to the 802.1X-enabled switch port when the multiple-host mode is not enabled (the default). The switch detects the client when the port link state changes to the up state. If a client leaves or is replaced with another client, the switch changes the port link state to down, and the port returns to the unauthorized state.

For 802.1X port-based authentication in a wireless LAN ([Figure 49-9](#)), you must configure the 802.1X port as a multiple-host port that is authorized as a wireless access point once the client is authenticated. (See the [“Resetting the 802.1X Configuration to the Default Values”](#) section on page 49-96.) When the port is authorized, all other hosts that are indirectly attached to the port are granted access to the network. If the port becomes unauthorized (reauthentication fails or an EAPoL-logoff message is received), the switch denies access to the network for all wireless access point-attached clients. In this topology, the wireless access point is responsible for authenticating clients attached to it, and the wireless access point acts as a client to the switch.

Figure 49-9 Wireless LAN Example



Configuring 802.1X Port-Based Authentication

To configure 802.1X, follow this procedure:

-
- | | |
|---------------|---|
| Step 1 | Enable 802.1X authentication. See the “Enabling 802.1X Authentication” section on page 49-28. |
| Step 2 | Configure switch to RADIUS server communication. See the “Configuring Switch-to-RADIUS-Server Communication” section on page 49-32. |
| Step 3 | Adjust the 802.1X timer values. See the “Changing the Quiet Period” section on page 49-86. |
| Step 4 | Configure optional features. See the “Configuring RADIUS-Provided Session Timeouts” section on page 49-55. |
-

These sections describe how to configure 802.1X:

- [Default 802.1X Configuration](#), page 49-27
- [802.1X Configuration Guidelines](#), page 49-28
- [Enabling 802.1X Authentication](#), page 49-28 (required)
- [Configuring Switch-to-RADIUS-Server Communication](#), page 49-32 (required)
- [Configuring Multiple Domain Authentication and Multiple Authorization](#), page 49-33
- [Configuring Limiting Login for Users](#), page 49-37
- [Configuring 802.1X Authentication with ACL Assignments and Redirect URLs](#), page 49-37
- [Configuring 802.1X Authentication with Per-User ACL and Filter-ID ACL](#), page 49-46
- [Configuring RADIUS-Provided Session Timeouts](#), page 49-55 (optional)
- [Configuring MAC Move](#), page 49-56 (optional)
- [Configuring MAC Replace](#), page 49-57 (optional)
- [Configuring Violation Action](#), page 49-58 (optional)
- [Configuring 802.1X with Guest VLANs](#), page 49-58 (optional)
- [Configuring 802.1X with MAC Authentication Bypass](#), page 49-62 (optional)
- [Configuring 802.1X with Inaccessible Authentication Bypass](#), page 49-64 (optional)
- [Configuring 802.1X with Unidirectional Controlled Port](#), page 49-68 (optional)
- [Configuring 802.1X with VLAN User Distribution](#), page 49-69
- [Configuring 802.1X with Authentication Failed](#), page 49-72 (optional)
- [Configuring 802.1X with Voice VLAN](#), page 49-74 (optional)
- [Configuring Voice Aware 802.1x Security](#), page 49-75
- [Configuring 802.1X with VLAN Assignment](#), page 49-77
- [Enabling Fallback Authentication](#), page 49-79
- [Enabling Periodic Reauthentication](#), page 49-83 (optional)
- [Enabling Multiple Hosts](#), page 49-84 (optional)
- [Changing the Quiet Period](#), page 49-86 (optional)
- [Changing the Switch-to-Client Retransmission Time](#), page 49-87 (optional)

- [Setting the Switch-to-Client Frame-Retransmission Number, page 49-88](#) (optional)
- [Configuring an Authenticator and a Supplicant Switch with NEAT, page 49-89](#)
- [Manually Reauthenticating a Client Connected to a Port, page 49-96](#) (optional)
- [Initializing the 802.1X Authentication State, page 49-96](#)
- [Removing 802.1X Client Information, page 49-96](#)
- [Resetting the 802.1X Configuration to the Default Values, page 49-96](#) (optional)

Default 802.1X Configuration

Table 49-1 shows the default 802.1X configuration.

Table 49-1 **Default 802.1X Configuration**

Feature	Default Setting
Authentication, authorization, and accounting (AAA)	Disabled
RADIUS server <ul style="list-style-type: none"> • IP address • UDP authentication port • Key 	<ul style="list-style-type: none"> • None specified • 1645 • None specified
Per-interface 802.1X protocol enable state	Force-authorized The port transmits and receives normal traffic without 802.1X-based authentication of the client.
Periodic reauthentication	Disabled
Time between reauthentication attempts	3600 sec
Quiet period	60 sec Number of seconds that the switch remains in the quiet state following a failed authentication exchange with the client.
Retransmission time	30 sec Number of seconds that the switch should wait for a response to an EAP request/identity frame from the client before retransmitting the request.
Maximum retransmission number	2 Number of times that the switch sends an EAP-request/identity frame before restarting the authentication process.
Multiple host support	Disabled

Table 49-1 **Default 802.1X Configuration (continued)**

Feature	Default Setting
Client timeout period	30 sec When relaying a request from the authentication server to the client, the amount of time that the switch waits for a response before retransmitting the request to the client.
Authentication server timeout period	30 sec When relaying a response from the client to the authentication server, the amount of time that the switch waits for a reply before retransmitting the response to the server. This setting is not configurable.

802.1X Configuration Guidelines

Guidelines for configuring 802.1X authentication include the following:

- The 802.1X protocol is supported only on Layer 2 static access, PVLAN host ports, and Layer 3 routed ports. You cannot configure 802.1X for any other port modes.
- If you are planning to use VLAN assignment, be aware that the features use general AAA commands. For information on how to configure AAA, refer to the “[Enabling 802.1X Authentication](#)” section on page 49-28. Alternatively, you can refer to the Cisco IOS security documentation at this location:

http://www.cisco.com/en/US/products/ps6586/products_ios_technology_home.html



Note

802.1x and MAB authentication session for a client installs only one IPv4 IP-to-MAC binding at a time (while an IP Device Tracking session can maintain multiple IP-to-MAC bindings). This limitation of the 802.1x and MAB authentication session can cause issues with traffic to devices that maintain multiple IP-to-MAC bindings.

Enabling 802.1X Authentication

To enable 802.1X port-based authentication, you first must enable 802.1X globally on your switch, then enable AAA and specify the authentication method list. A method list describes the sequence and authentication methods that must be queried to authenticate a user.

The software uses the first method listed in the method list to authenticate users; if that method fails to respond, the software selects the next authentication method in the list. This process continues until there is successful communication with a listed authentication method or until all defined methods are exhausted. If authentication fails at any point in this cycle, the authentication process stops, and no other authentication methods are attempted.



Note

To allow VLAN assignment, you must enable AAA authorization to configure the switch for all network-related service requests.

To configure 802.1X port-based authentication, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# dot1x system-auth-control	Enables 802.1X on your switch. To disable 802.1X globally on the switch, use the no dot1x system-auth-control command.
Step 3	Switch(config)# aaa new-model	Enables AAA. To disable AAA, use the no aaa new-model command.
Step 4	Switch(config)# aaa authentication dot1x {default} method1 [method2...]	Creates an 802.1X AAA authentication method list. To create a default list that is used when a named list is not specified in the authentication command, use the default keyword followed by the methods that are to be used in default situations. The default method list is automatically applied to all interfaces. Enter at least one of these keywords: <ul style="list-style-type: none"> • group radius—Use the list of all RADIUS servers for authentication. • none—Use no authentication. The client is automatically authenticated by the switch without using the information supplied by the client. To disable 802.1X AAA authentication, use the no aaa authentication dot1x {default list-name} method1 [method2...] global configuration command.
Step 5	Switch(config)# aaa authorization network {default} group radius	(Optional) Configures the switch for user RADIUS authorization for all network-related service requests, such as VLAN assignment.
Step 6	Switch(config)# interface interface-id	Enters interface configuration mode and specifies the interface to be enabled for 802.1X authentication.
Step 7	Switch(config-if)# switchport mode access	Specifies a nontrunking, nontagged single VLAN Layer 2 interface.
Step 8	Switch(config-if)# dot1x pae authenticator	Enables 802.1X authentication on the port with default parameters. Refer to the “Default 802.1X Configuration” section on page 49-27.
Step 9	Cisco IOS Release 12.2(50)SG and later Switch(config-if)# authentication port-control auto Cisco IOS Release 12.2(46)SG or earlier releases Switch(config-if)# dot1x port-control auto	Enables 802.1X authentication on the interface.
Step 10	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 11	Switch # show dot1x interface interface-id details	Verifies your entries. Check the PortControl row in the 802.1X port summary section of this display. The PortControl value is set to auto .
Step 12	Switch# show running-config	Verifies your entries.
Step 13	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

**Note**

Enabling Spanning Tree PortFast ensures that a port comes up immediately after authorization.

**Note**

Whenever you configure any 802.1X parameter on a port, a dot1x authenticator is automatically created on the port. As a result, **dot1x pae authenticator** appears in the configuration, ensuring that dot1x authentication still works on legacy configurations without manual intervention.

This example shows how to enable 802.1X and AAA on Fast Ethernet port 2/1, and how to verify the configuration:

Cisco IOS Release 12.2(50)SG and later

```
Switch# configure terminal
Switch(config)# dot1x system-auth-control
Switch(config)# aaa new-model
Switch(config)# aaa authentication dot1x default group radius
Switch(config)# interface fastethernet2/1
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# authentication port-control auto
Switch(config-if)# end
```

```
Switch# show authentication sessions interface f9/2
```

```
Interface: FastEthernet9/2
MAC Address: 0007.e95d.83c4
IP Address: Unknown
Status: Running
Domain: UNKNOWN
Oper host mode: single-host
Oper control dir: both
Session timeout: N/A
Idle timeout: N/A
Common Session ID: 0A050B160000009505106398
Acct Session ID: 0x0000009B
Handle: 0x0D000095
```

```
Runnable methods list:
```

Method	State
dot1x	Running
mab	Not run

Cisco IOS Release 12.2(46)SG or earlier

```
Switch# configure terminal
Switch(config)# dot1x system-auth-control
Switch(config)# aaa new-model
Switch(config)# aaa authentication dot1x default group radius
Switch(config)# interface fastethernet2/1
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# dot1x port-control auto
Switch(config-if)# end
```

```
Switch# show dot1x interface f9/2 details
```

```
Dot1x Info for FastEthernet9/2
```

```
-----
PAE = AUTHENTICATOR
PortControl = AUTO
ControlDirection = Both
```

```

HostMode                = SINGLE_HOST
QuietPeriod              = 60
ServerTimeout            = 0
SuppTimeout              = 30
ReAuthMax                = 2
MaxReq                   = 2
TxPeriod                 = 30

Dot1x Authenticator Client List
-----
Supplicant                = 0007.e95d.83c4
Session ID                = 0A050B160000009505106398
    Auth SM State          = AUTHENTICATING
    Auth BEND SM State      = REQUEST
Port Status               = UNAUTHORIZED

```

The following example illustrates when a port is authorized:

```

Switch# show authentication sessions int G4/5
      Interface: GigabitEthernet4/5
      MAC Address: 0015.e981.0531
      IP Address: Unknown
      User-Name: ctssxp
      Status: Authz Success
      Domain: DATA
      Oper host mode: single-host
      Oper control dir: both
      Session timeout: N/A
      Idle timeout: N/A
      Common Session ID: 0A053F0F00000004041E6B0C
      Acct Session ID: 0x00000021
      Handle: 0x2C000004

Runnable methods list:
      Method   State
      dot1x    Authc Success

```

```

Switch# show dot1x interface G4/5 details

```

```

Dot1x Info for GigabitEthernet4/5
-----
PAE                        = AUTHENTICATOR
PortControl                = AUTO
ControlDirection           = Both
HostMode                   = SINGLE_HOST
QuietPeriod                = 60
ServerTimeout              = 0
SuppTimeout                = 30
ReAuthMax                  = 2
MaxReq                     = 2
TxPeriod                   = 30

Dot1x Authenticator Client List
-----
Supplicant                = 0015.e981.0531
Session ID                = 0A053F0F00000004041E6B0C
    Auth SM State          = AUTHENTICATED
    Auth BEND SM State      = IDLE
Port Status               = AUTHORIZED

```

Configuring Switch-to-RADIUS-Server Communication

A RADIUS security server is identified by its host name or IP address, host name and specific UDP port number, or IP address and specific UDP port numbers. The combination of the IP address and UDP port number creates a unique identifier, which enables RADIUS requests to be sent to multiple UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service (for example, authentication), the second host entry configured acts as the failover backup to the first one. The RADIUS host entries are tried in the order they were configured.

To configure the RADIUS server parameters on the switch, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# radius server host { <i>hostname</i> <i>ip-address</i> } auth-port <i>port-number</i> [acct-port <i>port-number</i>] [test username <i>name</i>] [ignore-auth-port] [ignore-acct-port] [idle-time <i>min</i>] key <i>string</i>	<p>Configures the RADIUS server parameters on the switch.</p> <p>For <i>hostname</i> <i>ip-address</i>, specify the hostname or IP address of the remote RADIUS server.</p> <p>To delete the specified RADIUS server, use the no radius server host {<i>hostname</i> <i>ip-address</i>} global configuration command.</p> <p>auth-port <i>port-number</i>—Specifies the UDP destination port for authentication requests. The default is 1645.</p> <p>acct-port <i>port-number</i>—Specifies the UDP destination port for accounting requests. The default is 1646.</p> <p>Use test username <i>name</i> to enable automated RADIUS server testing, and to detect the RADIUS server going up and down. The name parameter is the username used in the test access request sent to the RADIUS server; it does not need to be a valid user configured on the server. The ignore-auth-port and ignore-acct-port options disable testing on the authentication and accounting ports respectively.</p> <p>The idle-time <i>min</i> parameter specifies the number of minutes before an idle RADIUS server is tested to verify that it is still up. The default is 60 minutes.</p> <p>The key <i>string</i> specifies the authentication and encryption key used between the switch and the RADIUS daemon running on the RADIUS server. The key is a text string that must match the encryption key used on the RADIUS server.</p> <p>Note Always configure the key as the last item in the radius server host command syntax because leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in the key, do not enclose the key in quotation marks unless the quotation marks are part of the key. This key must match the encryption used on the RADIUS daemon.</p> <p>If you want to use multiple RADIUS servers, use this command multiple times.</p>
Step 3	Switch(config)# radius-server deadtime <i>min</i>	(Optional) Configures the number of minutes before a dead RADIUS server is tested to check whether it has come back up. The default is 1 minute.

	Command	Purpose
Step 4	Switch(config)# radius-server dead-criteria <i>time seconds tries num</i>	(Optional) Configures the criteria used to decide whether a RADIUS server is dead. The time parameter specifies the number of seconds after which a request to the server is unanswered before it is considered dead. The tries parameter specifies the number of times a request to the server is unanswered before it is considered dead. The recommended values for these parameters are tries equal to radius-server retransmit and time equal to radius-server retransmit x radius-server timeout .
Step 5	Switch(config)# ip radius source-interface <i>m/p</i>	Establishes the IP address to be used as the source address for all outgoing RADIUS packets.
Step 6	Switch(config)# end	Returns to privileged EXEC mode.
Step 7	Switch# show running-config	Verifies your entries.
Step 8	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to specify the server with IP address 172.120.39.46 as the RADIUS server. The first command specifies port 1612 as the authorization port, sets the encryption key to rad123.

The second command dictates that key matches are performed on the RADIUS server:

```
Switch# configure terminal
Switch(config)# radius server host 172.120.39.46 auth-port 1612 key rad123
Switch(config)# ip radius source-interface g3/2
Switch(config)# end
Switch#
```

You can globally configure the timeout, retransmission, and encryption key values for all RADIUS servers by using the **radius server host** global configuration command. If you want to configure these options on a per-server basis, use the **radius-server timeout**, **radius-server retransmit**, and the **radius-server key** global configuration commands.

You also need to create a AAA client setting on the RADIUS server. These settings include the IP address of the switch and the key string to be shared by both the server and the switch.

Configuring Multiple Domain Authentication and Multiple Authorization



Note

Multiple Authorization requires Cisco IOS Release 12.2(50)SG and later releases.

To configure Multiple Domain Authentication (MDA) and Multiple Authorization, perform this task.

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# radius-server vsa send authentication	Configures the network access server to recognize and use vendor-specific attributes (VSAs).
Step 3	Switch(config)# interface interface-id	Specifies the port to which multiple hosts are indirectly attached, and enters interface configuration mode.

	Command	Purpose
Step 4	<p>Cisco IOS Release 12.2(50)SG and later</p> <pre>Switch(config-if)# [no] authentication host-mode {single-host multi-host multi-domain} multi-auth</pre> <p>Cisco IOS Release 12.2(46)SG or earlier releases</p> <pre>Switch(config-if)# [no] dot1x host-mode {single-host multi-host multi-domain}</pre>	<p>The keywords allow the following:</p> <ul style="list-style-type: none"> • single-host—Single-host (client) on an IEEE 802.1X-authorized port. • multi-host—Multiple-hosts on an 802.1X-authorized port after authenticating a single host. • multi-domain—Both a host and a voice device (such as an IP phone, Cisco or non-Cisco), to authenticate on an IEEE 802.1X-authorized port. <p>Note You must configure a voice VLAN for an IP phone when the host mode is set to multi-domain. For more information, see Chapter 46, “Configuring Voice Interfaces.”</p> <ul style="list-style-type: none"> • multi-auth—Allows multiple hosts and a voice device, such as an IP phone (Cisco or non-Cisco), to be authenticated on an IEEE 802.1x-authorized port. This keyword requires Cisco IOS Release 12.2(50)SG or a later release. <p>Ensure that the dot1x port-control interface configuration command is set to auto for the specified interface.</p> <p>To disable multiple hosts on the port, use the no authentication host-mode {multi-host multi-domain multi-auth} interface configuration command (for earlier releases, use the no dot1x host-mode {multi-host multi-domain} interface configuration command).</p>
Step 5	<pre>Switch(config-if)# switchport voice vlan vlan-id</pre>	(Optional) Configures the voice VLAN.
Step 6	<pre>Switch(config-if)# end</pre>	Returns to privileged EXEC mode.
Step 7	<pre>Switch# show dot1x interface interface-id [detail]</pre>	Verifies your entries.
Step 8	<pre>Switch# copy running-config startup-config</pre>	(Optional) Saves your entries in the configuration file.

This example shows how to enable 802.1X authentication and to allow multiple hosts:

Cisco IOS Release 12.2(50)SG and later

```
Switch(config)# interface gigabitethernet2/1
Switch(config-if)# authentication port-control auto
Switch(config-if)# authentication host-mode multi-host
Switch(config-if)# end
```

Cisco IOS Release 12.2(46)SG or earlier

```
Switch(config)# interface gigabitethernet2/1
Switch(config-if)# dot1x port-control auto
Switch(config-if)# dot1x host-mode multi-host
Switch(config-if)# end
```


This example shows how to enable MDA and to allow both a host and a 802.1X voice device (a Cisco or third-party phone with 802.1X supplicant) on the port:

Cisco IOS Release 12.2(50)SG and later

```
Switch# conf terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface FastEthernet3/1
Switch(config-if)# shut
Switch(config-if)# switchport access vlan 12
Switch(config-if)# switchport mode access
Switch(config-if)# switchport voice vlan 10
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# authentication port-control auto
Switch(config-if)# authentication host-mode multi-domain
Switch(config-if)# no shut
Switch(config-if)# end
```

Cisco IOS Release 12.2(46)SG or earlier

```
Switch# conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface FastEthernet3/1
Switch(config-if)# shut
Switch(config-if)# switchport access vlan 12
Switch(config-if)# switchport mode access
Switch(config-if)# switchport voice vlan 10
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# dot1x port-control auto
Switch(config-if)# dot1x host-mode multi-domain
Switch(config-if)# no shut
Switch(config-if)# end
```

This example shows how to enable MDA and to allow both a host and a non-802.1X voice device on the port:

Cisco IOS Release 12.2(50)SG and later

```
Switch# conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface FastEthernet3/1
Switch(config-if)# shut
Switch(config-if)# switchport access vlan 12
Switch(config-if)# switchport mode access
Switch(config-if)# switchport voice vlan 10
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# authentication port-control auto
Switch(config-if)# authentication host-mode multi-domain
Switch(config-if)# mab eap
Switch(config-if)# no shut
Switch(config-if)# end
```

Cisco IOS Release 12.2(46)SG or earlier

```
Switch# conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface FastEthernet3/1
Switch(config-if)# shut
Switch(config-if)# switchport access vlan 12
Switch(config-if)# switchport mode access
Switch(config-if)# switchport voice vlan 10
Switch(config-if)# dot1x pae authenticator
```

```
Switch(config-if)# dot1x port-control auto
Switch(config-if)# dot1x host-mode multi-domain
Switch(config-if)# dot1x mac-auth-bypass
Switch(config-if)# no shut
Switch(config-if)# end
```

This example shows how to verify the dot1x MDA settings on interface FastEthernet3/1:

```
Switch# show dot1x interface FastEthernet3/1 detail
```

```
Dot1x Info for FastEthernet3/1
-----
PAE = AUTHENTICATOR
PortControl = AUTO
ControlDirection = Both
HostMode = MULTI_DOMAIN
ReAuthentication = Disabled
QuietPeriod = 60
ServerTimeout = 30
SuppTimeout = 30
ReAuthPeriod = 3600 (Locally configured)
ReAuthMax = 2
MaxReq = 2
TxPeriod = 30
RateLimitPeriod = 0

Dot1x Authenticator Client List
-----
Domain = DATA
Supplicant = 0000.0000.ab01
    Auth SM State = AUTHENTICATED
    Auth BEND SM Stat = IDLE
Port Status = AUTHORIZED
Authentication Method = Dot1x
Authorized By = Authentication Server
Vlan Policy = 12

Domain = VOICE
Supplicant = 0060.b057.4687
    Auth SM State = AUTHENTICATED
    Auth BEND SM Stat = IDLE
Port Status = AUTHORIZED
Authentication Method = Dot1x
Authorized By = Authentication Server

Switch#
```

This example shows how to enable MDA and to authentication of multiple hosts and a voice device on an IEEE 802.1x-authorized port:



Note This example applies to Cisco IOS Release 12.2(50)SG and later releases.

```
Switch# conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface FastEthernet3/1
Switch(config-if)# shut
Switch(config-if)# switchport access vlan 12
Switch(config-if)# switchport mode access
Switch(config-if)# switchport voice vlan 10
Switch(config-if)# dot1x pae authenticator
```

```

Switch(config-if)# authentication port-control auto
Switch(config-if)# authentication host-mode multi-auth
Switch(config-if)# map eap
Switch(config-if)# no shut
Switch(config-if)# end

```

Configuring Limiting Login for Users

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# aaa new-model	Enables the authentication, authorization, and accounting (AAA) access control model.
Step 3	Switch(config)# aaa authentication login default local	Sets the authentication, authorization, and accounting (AAA) authentication by using the default authentication methods.
Step 4	Switch(config)# aaa authentication rejected n in m ban x	Configures the time period for which an user is blocked, if the user fails to successfully login within the specified time and login attempts. <ul style="list-style-type: none"> <i>n</i>—Specifies the number of times a user can try to login. <i>m</i>—Specifies the number of seconds within which an user can try to login. <i>x</i>—Specifies the time period an user is banned if the user fails to successfully login.
Step 5	Switch(config)# end	Exits from global configuration mode and returns to privileged EXEC mode.
Step 6	Switch# show aaa local user blocked	Displays the list of local users who were blocked.
Step 7	Switch# clear aaa local user blocked username username	Clears the information about the blocked local user.

Configuring 802.1X Authentication with ACL Assignments and Redirect URLs

This section includes these topics:

- [Downloadable ACL, page 49-37](#)
- [URL-Redirect, page 49-43](#)
- [Configuring a Downloadable Policy, page 49-45](#)

Downloadable ACL

The downloadable ACL (DACL) feature allows you to download device specific authorization policies from the authentication server. These policies activate after authentication succeeds for the respective client and the client's IP address was populated in the IP device tracking table. (Downloadable ACL is applied on the port, once the port is authenticated and the IP device tracking table has the host IP address entry).

Starting IOS XE Release 3.11.1E, IPv6 DACL is supported and DACL is defined on the ISE Server.

The following sections describe the configuration that is necessary to complement the related authentication (802.1X or MAB) configuration. (No unique configuration is required on the switch. All of the configuration is on the ACS.) After authentication succeeds, enter the **show ip access-list** command to display the downloadable ACLs.

Configuring the Switch for Downloadable ACL

To configure the switch for downloadable ACL, follow these steps:

Step 1 Configure the IP device tracking table.

```
Switch(config)# ip device tracking
```

Step 2 Configure RADIUS VSA to forward authentication.

```
Switch(config)# radius-server vsa send authentication
```

Step 3 Configure static ACL for the interface.

```
Switch(config)# int g2/9
Switch(config-if)# ip access-group pacl-4 in
```

For an IPv6 interface, add the following command:

```
Switch(config-if)# ipv6 traffic-filter PORT_IPV6_ACL in
```

Interface Configuration Example

```
Switch# show running-configuration interface g2/9
Building configuration...

Current configuration : 617 bytes
!
interface GigabitEthernet2/9
 switchport
 switchport access vlan 29
 switchport mode access
 switchport voice vlan 1234
 access-group mode prefer port
 ip access-group pacl-4 in
 ipv6 traffic-filter PORT_IPV6_ACL in
 speed 100
 duplex full
 authentication event fail action authorize vlan 111
 authentication event server dead action authorize vlan 333
 authentication event server alive action reinitialize
 authentication host-mode multi-auth
 authentication order dot1x
 authentication port-control auto
 authentication timer restart 100
 authentication timer reauthenticate 20
 authentication timer inactivity 200
 mab eap
 dot1x pae authenticator
end

Switch#
Switch# show ip access-list pacl-4
 10 permit ip host 10.1.1.1 host 2.2.2.2
```

```

20 permit icmp host 10.1.1.1 host 2.2.2.2
Switch#
Switch#show ipv6 access-list PORT_IPV6_ACL
deny ipv6 any 2110:10:10::/64 log
deny ipv6 any 2120:10:10::/64 log
deny ipv6 any 2130:10:10::/64 log
permit ipv6 any 2010:10:10::/64 log
permit ipv6 any 2020:10:10::/64 log
permit ipv6 any 2030:10:10::/64 log
permit ipv6 any 2001:10:10::/64 log
permit ipv6 any 2002:10:10::/64 log
permit ipv6 any 2003:10:10::/64 log

```

Debug Commands for DACL

The IP device tracking table contains the host IP address learned through ARP or DHCP.

The following command displays the constraints on the IP device tracking table:

```

Switch# show ip device tracking all
IP Device Tracking = Enabled
IP Device Tracking Probe Count = 3
IP Device Tracking Probe Interval = 30
-----
  IP Address      MAC Address      Interface      STATE
-----
50.0.0.12        0015.60a4.5e84   GigabitEthernet2/9   ACTIVE

```

The following **show authentication sessions** command displays the authentication sessions that contains the downloadable ACL obtained from ACS:



Note

The **show epm** command will be deprecated, displaying a warning message when used. Use the **show authentication sessions** command instead.

```

Switch-2033# show authentication sessions interface g2/9 details
Interface: GigabitEthernet2/9
MAC Address: 2c54.2d6a.0345
IPv6 Address: Unknown
IPv4 Address: 8.8.8.11
User-Name: 2C-54-2D-6A-03-45
Status: Authorized
Domain: DATA
Oper host mode: multi-auth
Oper control dir: both
Session timeout: N/A
Common Session ID: 0404040400000610081AA183
Acct Session ID: 0x000006F2
Handle: 0x760005B9
Current Policy: POLICY_Gi2/9

Server Policies:
ACS ACL: xACSAClX-IP-PERMIT_ALL_TRAFFIC-51de4498

Method status list:
Method      State
mab         Authc Success

```

The **show authentication sessions interface *interface-name* policy** displays session information in the form of Local Policies(features defined locally on the box), Server policies(features downloaded from radius) and Resultant Policies(the one with higher precedence when both local and server policies are present). By default, server policies have higher precedence than those defined locally.

AUTH# **show authentication sessions interface e0/0 policy**

```

Interface: Ethernet0/0
MAC Address: aabb.cc01.ff00
IPv6 Address: Unknown
IPv4 Address: Unknown
User-Name: gupn
Status: Authorized
Domain: DATA
Security Policy: Should Secure
Security Status: Unsecure
Oper host mode: multi-host
Oper control dir: both
Session timeout: N/A
Common Session ID: 0D0102330000000D0003329A
Acct Session ID: Unknown
Handle: 0x6F000002
Current Policy: POLICY_Et0/0

```

Local Policies:

```

Template: SVC_1 (priority 10)
Idle timeout: 500 sec
TAG: blue
URL Redirect: www.a.com
URL Redirect ACL: a

Template: SVC_3 (priority 20)
Idle timeout: 300 sec
TAG: red
URL_Redirect: www.b.com
URL-Redirect ACL: b

```

Server Policies:

```

Idle timeout: 800 sec

```

Resultant policies:

```

Idle timeout: 500 sec
TAG: blue
URL Redirect: www.a.com
URL Redirect ACL: a
TAG: red

```

Method status list:

```

Method      State
dot1x      Authc Success

```

The following command displays the contents of the downloadable ACL:

```

Switch# show ip access-lists xACSACLx-IP-auth-48b79b6e
Extended IP access list xACSACLx-IP-auth-48b79b6e (per-user)
  10 permit udp any any
Switch(config)#

```

Cisco ACS Configuration for DACL

To ensure correct functioning of the ACS configuration required for DACL, follow these steps:

- Step 1** Configure a downloadable IP ACL on the window that appears when you select **Radius Shared Profile > Downloadable IP ACL Content** (Figure 49-10).

Figure 49-10 Shared Profile Components

Shared Profile Components

Edit

Downloadable IP ACL Content

Name:

ACL Definitions
permit ip any host 10.10.10.10

- Step 2** Attach this downloadable ACL with the USER on the window that appears when you select **User > DACLs** (Figure 49-11).

Figure 49-11 Downloadable ACLs

Downloadable ACLs

☒ Assign IP ACL:

Cisco IOS/PIX 6.x RADIUS Attributes

205071

Cisco ISE Configuration for DACL



Note

Starting IOS XE 3.11.1E, DACL can be configured on the Identity Services Engine (ISE) server. IPv6 DACL is supported starting ISE 2.6 release. IPv4 DACL is supported on all releases of ISE.

To ensure correct functioning of the ISE configuration required for DACL, follow these steps on the ISE server:

- Step 1** From the home page, select **Policy > Policy Elements > Results**.

- Step 2** Select **Authorization > Downloadable ACLs** on the left pane of the window. Click **Add** to create a downloadable IP ACL. Fill the required fields - **Name**, **Description**, **IPv6** or **IPv4**, and **DACL** content.
- Step 3** Click **Submit**.

Figure 49-12 Downloadable ACLs Configuration on ISE

The screenshot shows the Cisco Identity Services Engine (ISE) configuration interface. The left sidebar has a navigation menu with 'Authorization' selected. The main area displays the 'Downloadable ACL' configuration form. The form includes the following fields and options:

- Name:** IPv6_DACL_1
- Description:** (empty text box)
- IP version:** IPv6 (selected), IPv4, Agnostic
- DACL Content:** A list of IP addresses: 1234567, 8910111, 2131415, 1617181, 9202122, 2324252, 6272829, 3031323, 3343536.
- Buttons:** Submit, Cancel

- Step 4** Bind the created IPv6 downloadable ACL to an Authorization Profile on the window that appears when you select **Authorization > Authorization Profiles > auth_profile**.

Figure 49-13 Bind IPv6 DACL

The screenshot shows the Cisco Identity Services Engine (ISE) configuration interface for an Authorization Profile. The left sidebar has a navigation menu with 'Authorization' selected. The main area displays the 'Authorization Profile' configuration form. The form includes the following fields and options:

- Name:** CORP13
- Description:** (empty text box)
- Access Type:** ACCESS_ACCEPT
- Network Device Profile:** Cisco
- Service Template:** (checkbox, unchecked)
- Track Movement:** (checkbox, unchecked)
- Passive Identity Tracking:** (checkbox, unchecked)
- Common Tasks:**
 - DACL Name:** (checkbox, unchecked)
 - IPv6 DACL Name:** (checkbox, checked) - IPv6_DACL_1

URL-Redirect

To configure URL-direct, you need to configure it on the ACS, and on the switch.

Starting Cisco IOS XE 3.11.1E, you can use ISE server to configure URL-direct.

Configuring ACS

To configure two Cisco-AV pairs, add the following statements under the user or group Cisco IOS/PIX 6x RADIUS attributes:

```
url-redirect-acl=urlacl
url-redirect=http://www.cisco.com
```



Note A default port ACL must be configured on the interface.

Configuring the Switch

To configure the switch for URL redirect, follow these steps:

-
- Step 1** Configure the IP device tracking table.
Switch(config)# **ip device tracking**
 - Step 2** Configure RADIUS by using the **send authentication** command.
Switch(config)# **radius-server vsa send authentication**
 - Step 3** Configure the URL redirect ACL (URLACL).
Switch# **ip access-list urlacl**
10 permit tcp any any
Switch#
 - Step 4** Configure static ACL (PACL) for the interface.
Switch(config)# **int g2/9**
Switch(config-if)# **ip access-group pacl-4 in**

For an IPv6 interface, add the following command:

```
Switch(config-if)# ipv6 traffic-filter PORT_IPV6_ACL in
```

Interface Configuration Example

```
Switch# show running-configuration int g2/9  
Building configuration...
```

```
Current configuration : 617 bytes  
!  
interface GigabitEthernet2/9  
  switchport  
  switchport access vlan 29  
  switchport mode access  
  switchport voice vlan 1234  
  access-group mode prefer port  
  ip access-group pacl-4 in
```

```

ipv6 traffic-filter PORT_IPV6_ACL in
speed 100
duplex full
authentication event fail action authorize vlan 111
authentication event server dead action authorize vlan 333
authentication event server alive action reinitialize
authentication host-mode multi-auth
authentication order dot1x
authentication port-control auto
authentication timer restart 100
authentication timer reauthenticate 20
authentication timer inactivity 200
mab
dot1x pae authenticator
end

Switch#

Switch# show access-list pacl-4
 10 permit ip host 10.1.1.1 host 2.2.2.2
 20 permit icmp host 10.1.1.1 host 2.2.2.2
Switch#

```

Verify URL-redirect by using the following commands.

The **show ip device tracking** command displays the constraints on the IP device tracking table:

```

Switch(config)# show ip device tracking all
IP Device Tracking = Enabled
IP Device Tracking Probe Count = 3
IP Device Tracking Probe Interval = 30
-----
  IP Address      MAC Address      Interface      STATE
-----
50.0.0.12        0015.60a4.5e84   GigabitEthernet2/9    ACTIVE

```

The **show authentication sessions interface details** command displays the URL-redirect-acl and URL-redirect URL information that downloads from the ACS:

```

Switch-2033# show authentication sessions int G1/0/7 details
Interface: GigabitEthernet1/0/7
MAC Address: 2c54.2d6a.0344
IPv6 Address: Unknown
IPv4 Address: 7.7.7.17
User-Name: 2C-54-2D-6A-03-44
Status: Authorized
Domain: DATA
Oper host mode: multi-auth
Oper control dir: both
Session timeout: N/A
Common Session ID: 0A4046D50000009502F03C4B
Acct Session ID: 0x000000D9
Handle: 0x0700005A
Current Policy: POLICY_Et0/0

Local Policies:

Server Policies:
  URL Redirect: www.cisco.com
  URL Redirect ACL: urlacl

Method status list:

  Method      State

```

```
mab                               Authc Success
```

For more information about AV pairs that are supported by Cisco IOS software, see the ACS configuration and command reference documentation about the software releases running on the AAA clients.

Guideline for DACL and URL Redirect

For downloadable ACL or URL redirect, the ACL source must be ANY (permit TCP ANY host 10.1.1.1 eq 80 or permit TCP ANY host 10.1.1.1 eq 443).

Configuring a Downloadable Policy

To configure downloadable policies, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# access-list <i>access-list-number</i> { deny permit } <i>source</i> [<i>source-wildcard</i>] [log]	Defines the default port ACL through a source address and wildcard. The <i>access-list-number</i> is a decimal from 1 to 99 or 1300 to 1999. Enter deny or permit to specify whether to deny or permit access if conditions match. <i>source</i> is the address of the network or host from which the packet is sent, specified as follows: <ul style="list-style-type: none"> The 32-bit quantity in dotted-decimal format The keyword any as an abbreviation for source and source-wildcard value of 0.0.0.0 255.255.255.255 You do not need a source-wildcard value. The keyword host as an abbreviation for source and source-wildcard of source 0.0.0.0. (Optional) Applies the source-wildcard wildcard bits to the source. (Optional) Enters log to cause an informational logging message about the packet that matches the entry to be sent to the console.
Step 3	Switch(config-if)# interface <i>interface-id</i>	Enters interface configuration mode.
Step 4	Switch(config-if)# ip access-group { <i>access-list-number</i> <i>name</i> } in	Controls access to the specified interface. This step is mandatory for a functioning downloaded policy.
Step 5	Switch(config)# exit	Returns to global configuration mode.
Step 6	Switch(config)# aaa new-model	Enables AAA.
Step 7	Switch(config)# aaa authorization network default local	Sets the authorization method to local. To remove the authorization method, use the no aaa authorization network default local command.

	Command	Purpose
Step 8	Switch(config)# ip device tracking	Enables the IP device tracking table. To disable the IP device tracking table, use the no ip device tracking global configuration commands. Note Starting from Cisco IOS XE Release 3.10.1E, the following IPDT commands are deprecated; there are no replacement commands: [no] ip device tracking probe count [no] ip device tracking probe delay . For more related information, see the <i>Configuring SISF-Based Device Tracking</i> chapter in this guide.
Step 9	Switch(config)# radius-server vsa send authentication	Configures the network access server to recognize and use vendor-specific attributes. Note The downloadable ACL must be operational.
Step 10	Switch(config)# end	Returns to privileged EXEC mode.
Step 11	Switch# show ip device tracking { all interface interface-id ip ip-address mac mac-address }	Displays information about the entries in the IP device tracking table.
Step 12	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

The following example illustrates how to configure a switch for downloadable policy:

```
Switch# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# aaa new-model
Switch(config)# aaa authorization network default local
Switch(config)# ip device tracking
Switch(config)# ip access-list extended default_acl
Switch(config-ext-nacl)# permit ip any any
Switch(config-ext-nacl)# exit
Switch(config)# radius-server vsa send authentication
Switch(config)# int fastEthernet 2/13
Switch(config-if)# ip access-group default_acl in
Switch(config-if)# exit
```

Configuring 802.1X Authentication with Per-User ACL and Filter-ID ACL

This section includes the following topics:

- [Per-User ACL and Filter-ID ACL, page 49-46](#)
- [Configuring a Per-User ACL and Filter-ID ACL, page 49-54](#)

Per-User ACL and Filter-ID ACL

Prior to Cisco IOS Release 12.2(52)SG, the Catalyst 4500 platform only supported downloadable ACLs, which work with the Cisco ACS server but not with third-party AAA servers. With Cisco IOS Release 12.2(52)SG, the Catalyst 4500 switch offers the Filter-ID/Per-user-acl enhancement, which allows ACL policy enforcement using a third-party AAA server.

The Filter-ID feature provides the following capabilities:

Filter-ID option allows an administrator to define the ACL name on the AAA server using IETF standard RADIUS attribute. The ACL itself must be preconfigured locally on the switch.

The Per-user-acl feature provides the following capabilities:

Per-user ACL allows an administrator to define the per-user ACL on the AAA server using Cisco RADIUS AV pairs. This action allows a third-party AAA server to interoperate by loading the Cisco RADIUS dictionary, which has Cisco Radius AV pairs configured as a VSA.



Note The RADIUS vendor-specific attributes (VSAs) allow vendors to support their own proprietary RADIUS attributes that are not included in standard RADIUS attributes.

Configuring the Switch

To configure the switch for per-user ACL and filter-ID ACL:

Step 1 Configure the IP device tracking table.

```
Switch(config)# ip device tracking
```

Step 2 Configure static ACL for the interface.

```
Switch(config)# int g2/9
Switch(config-if)# ip access-group pacl-4 in
```

For an IPv6 interface, add the following command:

```
Switch(config-if)# ipv6 traffic-filter PORT_IPV6_ACL in
```

Interface Configuration Example

```
Switch# show running-configuration interface g2/9
Building configuration...

Current configuration : 617 bytes
!
interface GigabitEthernet2/9
 switchport
 switchport access vlan 29
 switchport mode access
 switchport voice vlan 1234
 access-group mode prefer port
 ip access-group pacl-4 in
 speed 100
 duplex full
 authentication event fail action authorize vlan 111
 authentication event server dead action authorize vlan 333
 authentication event server alive action reinitialize
 authentication host-mode multi-auth
 authentication order dot1x
 authentication port-control auto
 authentication timer restart 100
 authentication timer reauthenticate 20
 authentication timer inactivity 200
 mab eap
 dot1x pae authenticator
end
```

```
Switch#
Switch# show ip access-list pacl-4
    10 permit ip host 10.1.1.1 host 2.2.2.2
    20 permit icmp host 10.1.1.1 host 2.2.2.2
Switch#
```

Per-User ACL Configuration in ACS

In the Group/User Setting page, scroll down to the Cisco IOS/PIX 6.x RADIUS Attributes section. Select the box next to [009\001 cisco-av-pair] and enter the elements of the per-user ACL. Per-user ACLs take this format:

`protocol_#:inacl# sequence number=ACE`

protocol Either `ip` (for IP-based ACLs) or `mac` (for MAC-based ACLs)

Figure 49-14 shows how members of the group you are configuring are denied all access to the 10.100.60.0 subnet, are denied HTTP access to the server at 10.100.10.116, and are permitted everywhere else.

Figure 49-14 Define the ACEs for the Per-User ACL

The screenshot shows the Cisco ACS Group Setup page. On the left is a navigation pane with icons for User Setup, Group Setup, Shared Profile Components, Network Configuration, System Configuration, Interface Configuration, Administration Control, External User Databases, Posture Validation, Network Access Profiles, Reports and Activity, and Online Documentation. The main content area has a 'Jump To' dropdown menu set to 'Access Restrictions'. Below this are two sections: 'IP Assignment' and 'Cisco IOS/PIX 6.x RADIUS Attributes'. The 'IP Assignment' section has three radio buttons: 'No IP address assignment', 'Assigned by dialup client' (which is selected), and 'Assigned from AAA Client pool'. Below the radio buttons is an empty text box. The 'Cisco IOS/PIX 6.x RADIUS Attributes' section has a list of attributes. The first attribute, '[009\001] cisco-av-pair', is selected with a checkbox. Its value is entered in a text box as:


```
ip:inacl#10=deny ip any
10.100.60.0 0.0.0.255
ip:inacl#20=deny tcp any host
10.100.10.116 eq www
ip:inacl#30=permit ip any any
```

 Below this are four other attributes, each with an unchecked checkbox and an empty text box: '[009\101] cisco-h323-credit-amount', '[009\102] cisco-h323-credit-time', '[009\103] cisco-h323-return-code', and '[009\104] cisco-h323-prompt-id'. At the bottom of the page are three buttons: 'Submit', 'Submit + Restart', and 'Cancel'.

274490

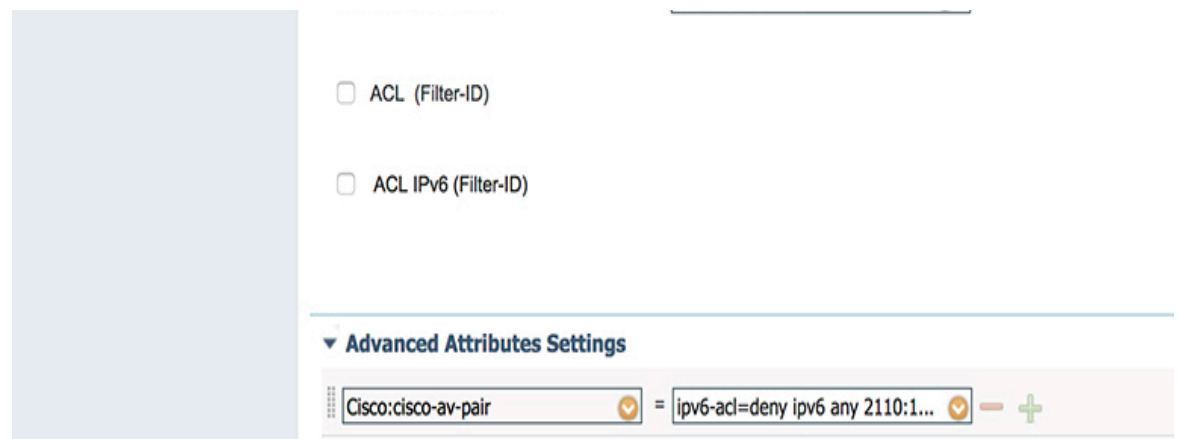
**Note**

Outbound ACLs (OUTACL) are not supported.

Per-User ACL Configuration in ISE

- Step 1** Select an authorization profile from the **Authorization > Authorization Profiles** page.
- Step 2** Select **cisco-av-pair** under **Advanced Attribute Settings** and enter the elements of Per-User ACL.
Cisco:cisco-av-pair = ipv6:inacl#1=deny ipv6 any 2110:10:10:10::64

Figure 49-15 Define ACEs for Per-User Configuration



Filter-Id Configuration in ACS

In the Group/User Setting page, scroll down to the IETF RADIUS Attributes section. Select the box next to Filter-Id and enter the ACL to apply for members of this group (Figure 49-16).

The Filter-Id ia in this format:

ACL_#.in

ACL Number of the ACL that was previously configured on the switch

Figure 49-16 Configuring the Filter-ID Attribute

The screenshot shows the Cisco Group Setup configuration window. On the left is a navigation pane with icons for User Setup, Group Setup, Shared Profile Components, Network Configuration, System Configuration, Interface Configuration, Administration Control, External User Databases, Posture Validation, Network Access Profiles, Reports and Activity, and Online. The main area is titled 'Group Setup' and has a 'Jump To' dropdown menu set to 'Access Restrictions'. Below this is a section titled 'IETF RADIUS Attributes' with a help icon. It contains several attributes with checkboxes and input fields:

- ☐ [006] Service-Type: Authenticate only (dropdown)
- ☐ [007] Framed-Protocol: Ascend MPP (dropdown)
- ☐ [009] Framed-IP-Netmask: 0.0.0.0 (text box)
- ☐ [010] Framed-Routing: None (dropdown)
- ☒ [011] Filter-ID: 100.in (text box)

At the bottom are three buttons: Submit, Submit + Restart, and Cancel. The number 274491 is visible in the bottom right corner.

**Note**

Outbound ACLs (for example, 100.out) are not supported.

Filter-Id Configuration in ISE

- Step 1** From the home page, select **Policy > Policy Elements > Results**.
- Step 2** Select **Authorization > Authorization Profiles > auth_profile**. Check the **ACL Filter-ID** checkbox and enter the name of the IPv6 Filter-ID.

Debug Commands for Per-User ACL and Filter-ID ACL

The IP device tracking table contains the host IP address learned through ARP or DHCP. The following command displays the constraints on the IP device tracking table:

```
Switch# show ip device tracking all
IP Device Tracking = Enabled
IP Device Tracking Probe Count = 3
IP Device Tracking Probe Interval = 30
-----
IP Address MAC Address Interface STATE
```



```
-----
50.0.0.12 0015.60a4.5e84 GigabitEthernet2/9 ACTIVE
```

The following command shows authentication sessions that contains the Filter-Id 100:

```
Switch-2033# show authentication sessions interface G2/9 details
      Interface: GigabitEthernet2/9
      MAC Address: 2c54.2d6a.0344
      IPv6 Address: Unknown
      IPv4 Address: 7.7.7.19
      User-Name: 2C-54-2D-6A-03-44
      Status: Authorized
      Domain: DATA
      Oper host mode: multi-auth
      Oper control dir: both
      Session timeout: N/A
      Common Session ID: 0A4046D50000009C0310AB47
      Acct Session ID: 0x000000E7
      Handle: 0xF3000061
      Current Policy: POLICY_Gi2/9

Server Policies:
      URL Redirect ACL: testacl
      ACS ACL: xACSACLx-IP-PERMIT_ALL_TRAFFIC-51def075
      Filter-ID: 100

Method status list:
      Method      State
      mab         Authc Success
```

The following command displays the contents of the per-user-acl (note that per-user-acl are shown above as the default port ACL configured on the interface, 151 is the default port ACL in the following example):

```
Switch# show access-list
151

      deny ip host 20.20.0.3 host 20.20.10.10

      10 permit ip any any (57 estimate matches)
```

The following command displays the number of sessions:

```
RouterRP# show authentication sessions

Interface    MAC Address    Method  Domain  Status Fg Session ID
Gi2/9        aabb.cc00.5600 mab      VOICE   Auth    0D0102340000000CEDF12589

Session count = 1

Key to Session Events Status Flags:

A - Applying Policy (multi-line status for details)
D - Awaiting Deletion
F - Final Removal in progress
I - Awaiting IIF ID allocation
P - Pushed Session (non-transient state)
R - Removing User Profile (multi-line status for details)
U - Applying User Profile (multi-line status for details)
X - Unknown Blocker
```

The following command displays authentication sessions that contains the per-user-acl:

```
S2049# show authentication sessions int gi 2/9 det
```

```

Interface: GigabitEthernet2/9
MAC Address: cdd.aabb.0001
IPv6 Address: Unknown
IPv4 Address: 6.6.65.66
User-Name: ccddaabb0001
Status: Authorized
Domain: DATA
Oper host mode: multi-auth
Oper control dir: both
Session timeout: N/A
Common Session ID: 0D0202010000003D04147B45
Acct Session ID: 0x0000004A
Handle: 0x7900002A
Current Policy: POLICY_Gi2/9

Local Policies:
  arp-probe-timeout: yes

Server Policies:
  Per-User ACL: GigabitEthernet1/0/23#v4#7C1C4AC
                : permit ip any host 10.1.1.20

Method status list:
  Method      State
  mab         Authc Success

```

The following command displays the contents of the per-user-acl (note that per-user-acl are shown above as the default port ACL configured on the interface, 151 is the default port ACL in the preceding example below):

```

Switch# show access-list
151

deny ip host 20.20.0.3 host 20.20.10.10

10 permit ip any any (57 estimate matches)
..
..
..(check for the mac access-list created)..
..
Extended MAC access list PerUser_MAC_ACL-589079192 (per-user)
  deny any host 0000.aaaa.aaaa
..

```

The following command shows that the Policy Enforced Module (EPM) session contains the Filter-Id 155 from ACS:



Note

The 156 IP extended ACL is to be preconfigured on the switch, so that the policy enforcement can happen.

```

Switch# show ip access-list 156
Extended IP access list 156
  10 deny ip any host 155.155.155.156
  20 deny ip any 156.100.60.0 0.0.0.255
  30 deny tcp any host 156.100.10.116 eq www

```

The following command shows authentication sessions that contains the Filter-Id TEST-ACL. TEST-ACL has been defined locally:

```

Switch-2033# show authentication sessions interface Gi2/9 details

```

```

Interface: GigabitEthernet2/9
MAC Address: 2c54.2d6a.0344
IPv6 Address: Unknown
IPv4 Address: 7.7.7.19
User-Name: 2C-54-2D-6A-03-44
Status: Authorized
Domain: DATA
Oper host mode: multi-auth
Oper control dir: both
Session timeout: N/A
Common Session ID: 0A4046D50000009C0310AB47
Acct Session ID: 0x000000E7
Handle: 0xF3000061
Current Policy: POLICY_Gi2/9

Local Policies:
Template: MYACL (priority 150)
Filter-ID: TEST-ACL

Server Policies:
URL Redirect ACL: testacl
ACS ACL: xACSACLx-IP-PERMIT_ALL_TRAFFIC-51def075

Method status list:
Method      State
mab         Authc Success

```

The following command displays the contents of the Filter-Id applied on the interface:

```

Switch# show ip access-list interface gi6/3
deny ip host 20.20.0.2 host 155.155.155.156
deny ip host 20.20.0.2 156.100.60.0 0.0.0.255
deny tcp host 20.20.0.2 host 156.100.10.116 eq www

```

Guidelines for Per-User ACL and Filter-ID ACL

- For per user ACL and Filter-ID ACL, the ACL source must be ANY (permit TCP ANY host 10.1.1.1 eq 80 or permit TCP ANY host 10.1.1.1 eq 443).
- Do not create per-user ACL in Cisco Identity Services Engine (ISE) with a duplicate sequence number. It results in a duplicate sequence number error on the console. It also results in partial policy configuration (only the last ACE/ACL of the same sequence number is applied).

Configuring a Per-User ACL and Filter-ID ACL

To configure per-user ACL and Filter-ID ACL, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# access-list <i>access-list-number</i> { deny permit } <i>source</i> [<i>source-wildcard</i>] [log]	<p>Defines the default port ACL through a source address and wildcard. The <i>access-list-number</i> is a decimal from 1 to 99 or 1300 to 1999.</p> <p>Enter deny or permit to specify whether to deny or permit access if conditions match.</p> <p><i>source</i> is the address of the network or host from which the packet is sent, specified as follows:</p> <ul style="list-style-type: none"> The 32-bit quantity in dotted-decimal format The keyword any as an abbreviation for source and source-wildcard value of 0.0.0.0 255.255.255.255 <p>You do not need a source-wildcard value.</p> <ul style="list-style-type: none"> The keyword host as an abbreviation for source and source-wildcard of source 0.0.0.0. <p>(Optional) Applies the source-wildcard wildcard bits to the source.</p> <p>(Optional) Enters log to cause an informational logging message about the packet that matches the entry to be sent to the console.</p>
Step 3	Switch(config-if)# interface <i>interface-id</i>	Enters interface configuration mode.
Step 4	Switch(config-if)# ip access-group { <i>access-list-number</i> <i>name</i> } in	<p>Controls access to the specified interface.</p> <p>This step is mandatory for a functioning downloaded policy.</p>
Step 5	Switch(config)# exit	Returns to global configuration mode.
Step 6	Switch(config)# aaa new-model	Enables AAA.
Step 7	Switch(config)# aaa authorization network default local	Sets the authorization method to local. To remove the authorization method, use the no aaa authorization network default local command.
Step 8	Switch(config)# ip device tracking	<p>Enables the IP device tracking table.</p> <p>To disable the IP device tracking table, use the no ip device tracking global configuration commands.</p> <p>Note Starting from Cisco IOS XE Release 3.10.1E, the following IPDT commands are deprecated; there are no replacement commands:</p> <p>[no] ip device tracking probe count</p> <p>[no] ip device tracking probe delay. For more related information, see the <i>Configuring SISF-Based Device Tracking</i> chapter in this guide.</p>
Step 9	Switch(config)# end	Returns to privileged EXEC mode.
Step 10	Switch# show ip device tracking { all interface <i>interface-id</i> ip <i>ip-address</i> mac <i>mac-address</i> }	Displays information about the entries in the IP device tracking table.
Step 11	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

The following example illustrates how to configure a switch for downloadable policy:

```
Switch# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# aaa new-model
Switch(config)# aaa authorization network default local
Switch(config)# ip device tracking
Switch(config)# ip access-list extended default_acl
Switch(config-ext-nacl)# permit ip any any
Switch(config-ext-nacl)# exit
Switch(config)# int fastEthernet 2/13
Switch(config-if)# ip access-group default_acl in
Switch(config-if)# exit
```

Configuring RADIUS-Provided Session Timeouts

You can configure the Catalyst 4500 series switch to use a RADIUS-provided reauthentication timeout.

To configure RADIUS-provided timeouts, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Enters interface configuration mode.
Step 3	Switch(config-if)# switchport mode access	Specifies a nontrunking, nontagged single VLAN Layer 2 interface.
Step 4	Switch(config-if)# dot1x pae authenticator	Enables 802.1X authentication on the port with default parameters. Refer to the “ Default 802.1X Configuration ” section on page 49-27.
Step 5	Cisco IOS Release 12.2(50)SG and later Switch(config-if)# authentication timer reauthenticate {interface server} Cisco IOS Release 12.2(46)SG or earlier releases Switch(config-if)# dot1x timeout reauth-attempts {interface server}	Sets the reauthentication period (seconds).
Step 6	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 7	Switch# show dot1x interface <i>interface-id details</i>	Verifies your entries.
Step 8	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to configure a switch to derive the reauthentication period from the server and to verify the configuration:

Cisco IOS Release 12.2(50):

```
Switch# configure terminal
Switch(config)# interface f7/1
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# authentication timer reauthenticate server
Switch(config-if)# end
Switch# show dot1x interface f7/1 det
```

```

Dot1x Info for FastEthernet7/11
-----
PAE                                = AUTHENTICATOR
PortControl                        = FORCE_AUTHORIZED
ControlDirection                  = Both
HostMode                          = SINGLE_HOST
ReAuthentication                  = Disabled
QuietPeriod                       = 60
ServerTimeout                     = 30
SuppTimeout                       = 30
ReAuthPeriod                      = (From Authentication Server)
ReAuthMax                         = 2
MaxReq                            = 2
TxPeriod                          = 30
RateLimitPeriod                   = 0

Dot1x Authenticator Client List Empty

Port Status                        = AUTHORIZED

Switch#

```

Cisco IOS Release 12.2(46) or earlier

```

Switch# configure terminal
Switch(config)# interface f7/1
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# dot1x timeout reauth-attempts server
Switch(config-if)# end
Switch# show dot1x interface f7/1 det

Dot1x Info for FastEthernet7/11
-----
PAE                                = AUTHENTICATOR
PortControl                        = FORCE_AUTHORIZED
ControlDirection                  = Both
HostMode                          = SINGLE_HOST
ReAuthentication                  = Disabled
QuietPeriod                       = 60
ServerTimeout                     = 30
SuppTimeout                       = 30
ReAuthPeriod                      = (From Authentication Server)
ReAuthMax                         = 2
MaxReq                            = 2
TxPeriod                          = 30
RateLimitPeriod                   = 0

Dot1x Authenticator Client List Empty

Port Status                        = AUTHORIZED

Switch#

```

Configuring MAC Move

MAC move allows an authenticated host to move from one switch port to another.



Note

You should remove port security before configuring MAC move.

To globally enable MAC move on the switch, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# authentication mac-move permit	Enable MAC move globally.
Step 3	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 4	Switch# show run	Verifies your entries.
Step 5	Switch # copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to globally enable MAC move on a switch:

```
Switch# configure terminal
Switch(config)# authentication mac-move permit
```

The following syslog messages displays when MAC-move happens:

```
%AUTHMGR-5-SECUREMACMOVE: <mac-addr> moved from <interface-name> to <interface-name>
```

Configuring MAC Replace

MAC replace allows new users to connect to abandoned ports.

If a user disconnects but the switch has not received the EAPoL-Logoff, the session will remain up. For single or multiple- domain modes, no new hosts can connect to that port. If a new host tries to connect, a violation is triggered on the port. Where the violation action is configured as replace, the desired behavior is for the NAD (switch) to terminate the initial session and reset the authentication sequence based on the new MAC.

To enable MAC replace on a switch, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface interface-id	Enters interface configuration mode.
Step 3	Switch(config-if)# authentication violation [restrict shutdown replace]	Tears down the old session and authenticates the new host, when a new host is seen in single or multiple- domain modes.
Step 4	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 5	Switch# show run	Verifies your entries.
Step 6	Switch # copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to globally enable MAC replace on a switch:

```
Switch# configure terminal
Switch(config)# interface f7/1
Switch(config-if)# authentication violation replace
```

The following syslog messages displays when MAC-replace occurs:

%AUTHMGR-5-SECUREMACREPLACE: <mac-addr> replaced <mac-addr> on <interface-name>

Configuring Violation Action

You can configure 802.1X security violation behavior as either shutdown, restrict, or replace mode, based on the response to the violation.

To configure the violation action, performing the following task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Enters interface configuration mode.
Step 3	Switch(config-if)# authentication violation [restrict shutdown replace]	(Optional) Configures the disposition of the port if a security violation occurs. The default action is to shut down the port. If the restrict keyword is configured, the port does not shut down. When a new host is seen in single or multiple- domain modes, replace mode tears down the old session and authenticates the new host.
Step 4	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 5	Switch# show run	Verifies your entries.
Step 6	Switch # copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to configure the violation mode shutdown on a switch:

```
Switch# configure terminal
Switch(config)# authentication violation shutdown
```

A port is error-disabled when a security violation triggers on shutdown mode. The following syslog messages displays:

```
%AUTHMGR-5-SECURITY_VIOLATION: Security violation on the interface <interface name>, new
MAC address <mac-address> is seen.
%PM-4-ERR_DISABLE: security-violation error detected on <interface name>, putting
<interface name> in err-disable state
```

Configuring 802.1X with Guest VLANs

You can configure a guest VLAN for each 802.1X port on the Catalyst 4500 series switch to provide limited services to clients, such as downloading the 802.1X client. These clients might be upgrading their system for 802.1X authentication, and some hosts, such as Windows 98 systems, might not be 802.1X-capable.

When you enable a guest VLAN on an 802.1X port, the Catalyst 4500 series switch assigns clients to a guest VLAN, provided one of the following apply:

- The authentication server does not receive a response to its EAPOL request or identity frame.
- The EAPOL packets are not sent by the client.

Beginning with Cisco IOS Release 12.2(25)EWA, the Catalyst 4500 series switch maintains the EAPOL packet history. If another EAPOL packet is detected on the interface during the lifetime of the link, network access is denied. The EAPOL history is reset upon loss of the link.

Any number of 802.1X-incapable clients are allowed access when the switch port is moved to the guest VLAN. If an 802.1X-capable client joins the same port on which the guest VLAN is configured, the port is put into the unauthorized state in the user-configured access VLAN, and authentication is restarted.

Guest VLANs are supported on 802.1X ports in single-host or multiple-hosts mode.

**Note**

When a port is put into a guest VLAN, it is automatically placed into multihost mode, and an unlimited number of hosts can connect using the port. Changing the multihost configuration does not effect a port in a guest VLAN.

**Note**

Except for an RSPAN VLAN or a voice VLAN, you can configure any active VLAN as an 802.1X guest VLAN.

To configure 802.1X with guest VLAN on a port, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Enters interface configuration mode and specifies the interface to be enabled for 802.1X authentication.
Step 3	Switch(config-if)# switchport mode access or Switch(config-if)# switchport mode private-vlan host	Specifies a nontrunking, nontagged single VLAN Layer 2 interface. Specifies that the ports with a valid PVLAN trunk association become active host PVLAN trunk ports.
Step 4	Switch(config-if)# dot1x pae authenticator	Enables 802.1X authentication on the port with default parameters. Refer to the “Default 802.1X Configuration” section on page 49-27.
Step 5	Cisco IOS Release 12.2(50)SG and later Switch(config-if)# authentication event no-response action authorize vlan <i>vlan-id</i> Cisco IOS Release 12.2(46)SG or earlier releases Switch(config-if)# dot1x guest-vlan <i>vlan-id</i>	Enables a guest VLAN on a particular interface. To disable the guest VLAN feature on a particular port, use the no authentication event no-response action authorize vlan interface configuration command (for earlier releases, use the no dot1x guest-vlan interface configuration command).
Step 6	Cisco IOS Release 12.2(50)SG and later Switch(config-if)# authentication port-control auto Cisco IOS Release 12.2(46)SG or earlier releases Switch(config-if)# dot1x port-control auto	Enables 802.1X authentication on the interface.
Step 7	Switch(config-if)# end	Returns to configuration mode.
Step 8	Switch(config)# end	Returns to privileged EXEC mode.

This example shows how to enable regular VLAN 50 on Fast Ethernet 4/3 as a guest VLAN on a static access port:

Cisco IOS Release 12.2(50)SG and later

```
Switch# configure terminal
Switch(config)# interface fa4/3
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# authentication event no-response action authorize vlan 50
Switch(config-if)# authentication port-control auto
Switch(config-if)# end
Switch#
```

Cisco IOS Release 12.2(46)SG or earlier

```
Switch# configure terminal
Switch(config)# interface fa4/3
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# dot1x guest-vlan 50
Switch(config-if)# dot1x port-control auto
Switch(config-if)# end
Switch#
```

This example shows how to enable a secondary PVLAN 100 as a guest VLAN on a PVLAN host port:

Cisco IOS Release 12.2(50)SG and later

```
Switch# configure terminal
Switch(config)# interface fa4/3
Switch(config-if)# switchport mode private-vlan host
Switch(config-if)# authentication port-control auto
Switch(config-if)# authentication event no-response action authorize vlan 100
Switch(config-if)# end
Switch#
```

Cisco IOS Release 12.2(46)SG or earlier

```
Switch# configure terminal
Switch(config)# interface fa4/3
Switch(config-if)# switchport mode private-vlan host
Switch(config-if)# dot1x port-control auto
Switch(config-if)# dot1x guest-vlan 100
Switch(config-if)# end
Switch#
```

To allow supplicants into a guest VLAN on a switch, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch# dot1x guest-vlan supplicant	(Optional) Enables supplicants to be allowed into the guest VLANs globally on the switch. Note Although not visible in the CLI for Cisco IOS Release 12.3(31)SG, legacy configurations that include the dot1x guest-vlan supplicant command still work. We do not recommend that you use this command. However, because the authentication failed VLAN option makes it unnecessary. To disable the supplicant guest VLAN feature on a switch, use the no dot1x guest-vlan supplicant global configuration command.
Step 3	Switch(config)# interface interface-id	Enters interface configuration mode and specifies the interface to be enabled for 802.1X authentication.
Step 4	Switch(config-if)# switchport mode access or Switch(config-if)# switchport mode private-vlan host	Specifies a nontrunking, nontagged single VLAN Layer 2 interface. Specifies that the ports with a valid PVLAN trunk association become active host PVLAN trunk ports.
Step 5	Switch(config-if)# dot1x pae authenticator	Enables 802.1X authentication on the port with default parameters. Refer to the “Default 802.1X Configuration” section on page 49-27 .
Step 6	Switch(config-if)# dot1x guest-vlan vlan-id	Specifies an active VLAN as an 802.1X guest VLAN. The range is 1 to 4094.
Step 7	Cisco IOS Release 12.2(50)SG and later Switch(config-if)# authentication port-control auto Cisco IOS Release 12.2(46)SG or earlier releases Switch(config-if)# dot1x port-control auto	Enables 802.1X authentication on the interface.
Step 8	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 9	Switch# show dot1x interface interface-id	Verifies your entries.
Step 10	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to enable the guest VLAN feature and to specify VLAN 5 as a guest VLAN:

Cisco IOS Release 12.2(50)SG and later

```
Switch# configure terminal
Switch(config)# dot1x guest-vlan supplicant
Switch(config)# interface gigabitethernet5/9
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# authentication event no-response action authorize vlan 5
Switch(config-if)# authentication port-control auto
Switch(config-if)# end
Switch#
```

Cisco IOS Release 12.2(46)SG or earlier

```

Switch# configure terminal
Switch(config)# dot1x guest-vlan supplicant
Switch(config)# interface gigabitethernet5/9
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# dot1x guest-vlan 5
Switch(config-if)# dot1x port-control auto
Switch(config-if)# end
Switch#

```

Configuring 802.1X with MAC Authentication Bypass

To enable MAC Authentication Bypass (MAB), perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Specifies the port to be configured, and enters interface configuration mode.
Step 3	Switch(config-if)# switchport mode access or Switch(config-if)# switchport mode private-vlan host	Specifies a nontrunking, nontagged single VLAN Layer 2 interface. Specifies that the ports with a valid PVLAN trunk association become active host PVLAN trunk ports.
Step 4	Switch(config-if)# dot1x pae authenticator	Enables 802.1X authentication on the port with default parameters. Refer to the “Default 802.1X Configuration” section on page 49-27.
Step 5	Cisco IOS Release 12.2(50)SG and later Switch(config-if)# authentication port-control auto Cisco IOS Release 12.2(46)SG or earlier releases Switch(config-if)# dot1x port-control auto	Enables 802.1X authentication on the interface.
Step 6	Cisco IOS Release 12.2(50)SG and later Switch(config-if)# mab [eap] Cisco IOS Release 12.2(46)SG or earlier releases Switch(config-if)# dot1x mac-auth-bypass [eap]	Enables MAB on a switch. The eap option specifies that a complete EAP conversation should be used, as opposed to standard RADIUS Access-Request, Access-Accept conversation. By default, the eap option is not enabled for MAB.
Step 7	Switch(config)# end	Returns to privileged EXEC mode.
Step 8	Switch# show mab interface <i>interface-id</i> details	(Optional) Verifies your entries.
Step 9	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.



Note

Removing a 802.1X MAB configuration from a port does not impact the authorized or authenticated state of the port. If the port is in an unauthenticated state, it remains in that state. If the port is in an authenticated state because of MAB, the switch reverts to the 802.1X Authenticator. If the port was

already authorized with a MAC address and the MAB configuration was removed, the port remains in an authorized state until reauthentication occurs. At that time, if an 802.1X supplicant is detected on the wire, the MAC address is removed.

This example shows how to enable MAB on Gigabit Ethernet interface 3/3 and to verify the configuration:

Cisco IOS Release 12.2(50)SG and later

```
Switch# configure terminal
Switch(config)# interface gigabitethernet3/3
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# authentication port-control auto
Switch(config-if)# mab
Switch(config-if)# end
Switch# show mab int g3/3 details
MAB details for GigabitEthernet3/3
-----
Mac-Auth-Bypass           = Enabled

MAB Client List
-----
Client MAC                = 0001.0001.0001
Session ID                = C0A8016F0000002304175914
MAB SM state              = TERMINATE
Auth Status               = AUTHORIZED
```

Cisco IOS Release 12.2(46)SG or earlier

```
Switch# configure terminal
Switch(config)# interface gigabitethernet3/3
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# dot1x port-control auto
Switch(config-if)# dot1x mac-auth-bypass
Switch(config-if)# end
Switch# show dot1x int g3/3 details
Dot1x Info for GigabitEthernet3/3
-----
PAE                        = AUTHENTICATOR
PortControl                = AUTO
ControlDirection          = Both
HostMode                   = SINGLE_HOST
ReAuthentication           = Disabled
QuietPeriod                = 60
ServerTimeout              = 30
SuppTimeout                = 30
ReAuthPeriod               = 3600 (Locally configured)
ReAuthMax                  = 2
MaxReq                     = 2
TxPeriod                   = 1
RateLimitPeriod            = 0
Mac-Auth-Bypass           = Enabled

Dot1x Authenticator Client List
-----
Supplicant                  = 0000.0000.0001
Auth SM State               = AUTHENTICATED
Auth BEND SM Stat          = IDLE
Port Status                 = AUTHORIZED
Authentication Method       = MAB
Authorized By                = Authentication Server
```

Vlan Policy = N/A

Switch#

Configuring 802.1X with Inaccessible Authentication Bypass



Caution

You must configure the switch to monitor the state of the RADIUS server as described in the section [Configuring Switch-to-RADIUS-Server Communication, page 49-32](#) for Inaccessible Authentication Bypass to work properly. Specifically, you must configure the RADIUS test username, idle-time, deadtime and dead-criteria. Failure to do so results in the switch failing to detect that the RADIUS server has gone down, or prematurely marking a dead RADIUS server as alive again.

To configure a port as a critical port and to enable the Inaccessible Authentication Bypass feature, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# dot1x critical eapol	(Optional) Configures whether to send an EAPOL-Success packet when a port is critically authorized partway through an EAP exchange. Note Some supplicants require this. The default is not to send EAPOL-Success packets when a port is critically authorized partway through an EAP exchange. If there is no ongoing EAP exchange at the time when a port is critically authorized, EAPOL-Success packet is always sent out regardless of this option.
Step 3	Supervisor Engine 6-E, 6L-E—Cisco IOS Release 12.2(50)SG and later; Supervisor Engine 7-E, 7L-E, 8-E—Cisco IOS Release 15.0(1)X and later; Supervisor Engine 9-E—Cisco IOS XE Release 3.10.0E Switch(config)# authentication critical recovery delay msec Cisco IOS Release 12.2(46)SG or earlier releases Switch(config)# dot1x critical recovery delay msec	(Optional) Specifies a throttle rate for the reinitialization of critically authorized ports when the RADIUS server becomes available. The default throttle rate is 100 milliseconds. This means that 10 ports reinitialize per second.
Step 4	Switch(config)# interface interface-id	Specifies the port to be configured and enters interface configuration mode.
Step 5	Switch(config-if)# switchport mode access or Switch(config-if)# switchport mode private-vlan host	Specifies a nontrunking, nontagged single VLAN Layer 2 interface. Specifies that the ports with a valid PVLAN trunk association become active host PVLAN trunk ports.
Step 6	Switch(config-if)# dot1x pae authenticator	Enables 802.1X authentication on the port with default parameters. Refer to the “Default 802.1X Configuration” section on page 49-27.

	Command	Purpose
Step 7	<code>Switch(config-if)# authentication port-control auto</code>	Enables 802.1X authentication on the interface.
Step 8	<p>Supervisor Engine 6-E, 6L-E—Cisco IOS Release 12.2(50)SG and later Supervisor Engine 7-E, 7L-E, 8-E—Cisco IOS Release 15.0(1)XO and later Supervisor Engine 9-E—Cisco IOS XE Release 3.10.0E and later</p> <p><code>Switch(config-if)# authentication event server dead action authorize [vlan vlan-id]</code></p> <p>Cisco IOS Release 12.2(46)SG or earlier releases</p> <p><code>Switch(config-if)# dot1x critical</code></p> <p>OR</p> <p>Supervisor Engine 6-E, 6L-E—Cisco IOS Release 15.0(2)SG and later Supervisor Engine 7-E, 7L-E, 8-E—Cisco IOS Release XE 3.2.0SG and later Supervisor Engine 9-E—Cisco IOS XE Release 3.10.0E and later</p> <p><code>Switch(config-if)# [no] authentication event server dead action reinitialize [vlan vlan-id]</code></p>	<p>Enables the Inaccessible Authentication Bypass feature for data clients on the port and specifies a VLAN into which data clients are assigned. If no VLAN is specified, data clients are assigned into the configured data VLAN on the port.</p> <p>To disable the feature, use the no authentication event server dead action authorize vlan interface configuration command (for earlier releases, use the no dot1x critical interface configuration command).</p> <p>Alternatively, starting with Cisco IOS Release 15.0(2)SG you can enable Inaccessible Authentication Bypass for data clients using the authentication event server dead action reinitialize vlan interface configuration command which forces all authorized data clients to be reauthenticated when RADIUS becomes unavailable and a client attempts to authenticate. This only applies to data devices. Voice devices are unaffected.</p> <p>To disable it, use the no authentication event server dead action reinitialize vlan interface configuration command.</p>
Step 9	<p>Supervisor Engine 6-E, 6L-E—Cisco IOS Release 15.0(2)SG and later Supervisor Engine 7-E, 7L-E, 8-E—Cisco IOS Release XE 3.2.0SG and later Supervisor Engine 9-E—Cisco IOS XE Release 3.10.0E and later</p> <p><code>Switch(config-if)# authentication event server dead action authorize voice</code></p>	<p>(Optional) Enables Inaccessible Authentication Bypass for voice clients on the port. This command applies to Multiple Domain Authentication and Multiple Authentication modes.</p> <p>To disable the feature, use the no authentication event server dead action authorize voice interface configuration command.</p>
Step 10	<p>Supervisor Engine 6-E, 6L-E—Cisco IOS Release 12.2(50)SG and later Supervisor Engine 7-E, 7L-E, 8-E—Cisco IOS Release 15.0(1)XO and later Supervisor Engine 9-E—Cisco IOS XE Release 3.10.0E and later</p> <p><code>Switch(config-if)# authentication event server alive action reinitialize</code></p> <p>Cisco IOS Release 12.2(46)SG or earlier releases</p> <p><code>Switch(config-if)# dot1x critical recovery action reinitialize</code></p>	<p>(Optional) Specifies that the port should be reinitialized if it is critically authorized and RADIUS becomes available.</p> <p>The default is not to reinitialize the port.</p>

	Command	Purpose
Step 11	Switch(config)# end	Returns to privileged EXEC mode.
Step 12	Switch# show dot1x interface interface-id details	(Optional) Verifies your entries.
Step 13	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

The following example shows a full configuration of 802.1X with Inaccessible Authentication Bypass, including required AAA and RADIUS configuration as specified in the [“Enabling 802.1X Authentication”](#) section on page 49-28 and [“Configuring Switch-to-RADIUS-Server Communication”](#) section on page 49-32.

The RADIUS server configured is at IP address 10.1.2.3, using port 1645 for authentication and 1646 for accounting. The RADIUS secret key is mykey. The username used for the test server probes is randomuser. The test probes for both living and dead servers are generated once per minute. The interface FastEthernet 3/1 is configured to critically authenticate into VLAN 17 when AAA becomes unresponsive, and to reinitialize automatically when AAA becomes available again.

Cisco IOS Release 12.2(50)SG and later

```
Switch# configure terminal
Switch(config)# aaa new-model
Switch(config)# aaa authentication dot1x default group radius
Switch(config)# dot1x system-auth-control
Switch(config)# radius server host 10.1.2.3 auth-port 1645 acct-port 1646 test username
randomuser idle-time 1 key mykey
Switch(config)# radius-server deadtime 1
Switch(config)# radius-server dead-criteria time 15 tries 3
Switch(config)# interface f3/1
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# authentication port-control auto
Switch(config-if)# authentication event server dead action authorize vlan 17
Switch(config-if)# end
Switch# show dot1x int fastethernet 3/1 details
```

```
Dot1x Info for FastEthernet3/1
-----
PAE                               = AUTHENTICATOR
PortControl                       = AUTO
ControlDirection                 = Both
HostMode                         = SINGLE_HOST
ReAuthentication                 = Disabled
QuietPeriod                      = 60
ServerTimeout                    = 30
SuppTimeout                     = 30
ReAuthPeriod                     = 3600 (Locally configured)
ReAuthMax                       = 2
MaxReq                           = 2
TxPeriod                         = 30
RateLimitPeriod                 = 0
Critical-Auth                    = Enabled
Critical Recovery Action         = Reinitialize
Critical-Auth VLAN               = 17

Dot1x Authenticator Client List
-----
Supplicant                       = 0000.0000.0001

Auth SM State                    = AUTHENTICATING
```



```
Auth BEND SM Stat = RESPONSE
Port Status          = AUTHORIZED
Authentication Method = Dot1x
Authorized By        = Critical-Auth
Operational HostMode = SINGLE_HOST
Vlan Policy          = 17
```

Switch#

Cisco IOS Release 12.2(46)SG or earlier

```
Switch# configure terminal
Switch(config)# aaa new-model
Switch(config)# aaa authentication dot1x default group radius
Switch(config)# dot1x system-auth-control
Switch(config)# radius server host 10.1.2.3 auth-port 1645 acct-port 1646 test username
randomuser idle-time 1 key mykey
Switch(config)# radius-server deadtime 1
Switch(config)# radius-server dead-criteria time 15 tries 3
Switch(config)# interface f3/1
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# dot1x port-control auto
Switch(config-if)# dot1x critical
Switch(config-if)# dot1x critical vlan 17
Switch(config-if)# dot1x critical recovery action reinitialize
Switch(config-if)# end
Switch# show dot1x int fastethernet 3/1 details
```

Dot1x Info for FastEthernet3/1

```
-----
PAE                                = AUTHENTICATOR
PortControl                        = AUTO
ControlDirection                  = Both
HostMode                          = SINGLE_HOST
ReAuthentication                   = Disabled
QuietPeriod                       = 60
ServerTimeout                     = 30
SuppTimeout                       = 30
ReAuthPeriod                      = 3600 (Locally configured)
ReAuthMax                         = 2
MaxReq                            = 2
TxPeriod                          = 30
RateLimitPeriod                   = 0
Critical-Auth                     = Enabled
Critical Recovery Action           = Reinitialize
Critical-Auth VLAN                = 17
```

Dot1x Authenticator Client List

```
-----
Supplicant                        = 0000.0000.0001
```

```
Auth SM State      = AUTHENTICATING
Auth BEND SM Stat = RESPONSE
Port Status        = AUTHORIZED
Authentication Method = Dot1x
Authorized By      = Critical-Auth
Operational HostMode = SINGLE_HOST
Vlan Policy        = 17
```

Switch#

Configuring 802.1X with Unidirectional Controlled Port

To configure unidirectional controlled port, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Specifies the port to be configured and enters interface configuration mode.
Step 3	Switch(config-if)# switchport mode access or Switch(config-if)# switchport mode private-vlan host	Specifies a nontrunking, nontagged single VLAN Layer 2 interface. Specifies that the ports with a valid PVLAN trunk association become active host PVLAN trunk ports.
Step 4	Switch(config-if)# dot1x pae authenticator	Enables 802.1X authentication on the port with default parameters. Refer to the “ Default 802.1X Configuration ” section on page 49-27.
Step 5	Switch(config-if)# authentication port-control auto	Enables 802.1X authentication on the interface.
Step 6	Cisco IOS Release 12.2(50)SG and later Switch(config-if)# authentication control-direction {in both} Cisco IOS Release 12.2(46)SG or earlier releases Switch(config-if)# dot1x control-direction {in both}	Enables unidirectional port control on each port.
Step 7	Switch(config)# end	Returns to privileged EXEC mode.
Step 8	Switch# show dot1x interface <i>interface-id details</i>	(Optional) Verifies your entries.
Step 9	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.



Note

Unidirectional controlled port only works when Spanning Tree PortFast is enabled on the port. Unidirectional controlled port and Spanning Tree PortFast should be configured on a switch port that connects to a host. If two such ports are connected together with an Ethernet cable, high CPU utilization may result because host learning will be flapping between the two ports.

This example shows how to enable unidirectional port control:

Cisco IOS Release 12.2(50)SG and later

```
Switch# configure terminal
Switch(config)# interface gigabitEthernet3/3
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# authentication port-control auto
Switch(config-if)# authentication control-direction in
Switch(config-if)# end
Switch# show dot1x int g3/3
Dot1x Info for GigabitEthernet3/3
-----
PAE                               = AUTHENTICATOR
PortControl                       = AUTO
```

```

ControlDirection      = In
HostMode               = SINGLE_HOST
ReAuthentication       = Disabled
QuietPeriod           = 60
ServerTimeout         = 30
SuppTimeout           = 30
ReAuthPeriod          = 3600 (Locally configured)
ReAuthMax              = 2
MaxReq                 = 2
TxPeriod              = 30
RateLimitPeriod       = 0

Switch#

```

Cisco IOS Release 12.2(46)SG or earlier

```

Switch# configure terminal
Switch(config)# interface gigabitethernet3/3
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# authentication port-control auto
Switch(config-if)# dot1x control-direction in
Switch(config-if)# end
Switch# show dot1x int g3/3
Dot1x Info for GigabitEthernet3/3
-----
PAE                      = AUTHENTICATOR
PortControl              = AUTO
ControlDirection         = In
HostMode                  = SINGLE_HOST
ReAuthentication         = Disabled
QuietPeriod              = 60
ServerTimeout            = 30
SuppTimeout              = 30
ReAuthPeriod             = 3600 (Locally configured)
ReAuthMax                 = 2
MaxReq                    = 2
TxPeriod                  = 30
RateLimitPeriod          = 0

Switch#

```

Configuring 802.1X with VLAN User Distribution

You will need to configure the switch and ACS to configure 802.1X with VLAN user distribution.

Configuring the Switch

To configure the switch, follow these steps:

Step 1 Create a VLAN group on the switch.

Enter the following commands to create a VLAN group and assign some VLANs to the VLAN group. The following example creates the VLAN group **eng-group** and maps VLANs 20 to 24 to that group:

```

Switch# configure terminal
Switch(config)# vlan group eng-group vlan-list 20-24
Switch(config)# end
Switch# show vlan group group-name eng-group

```

```
Group Name VLANs Mapped
-----
eng-group      20-24
```



Note

Ensure that the VLANs you specify as part of the VLAN group are enabled on the switch. Only specified VLANs are considered for assignment.

Step 2

Configure the individual ports for multidomain, single-host or multiple- host.
For details, refer to the [“Enabling 802.1X Authentication” section on page 49-28](#).

show commands

Use the following **show** commands to display the member VLANs in a VLAN group:

show command	Purpose
show vlan group all	Displays the member VLANs for all the VLAN groups configured on the device.
show vlan group group-name <i>vlan-group-name</i>	Displays the member VLANs in a VLAN group with the given VLAN group name.
show vlan group group-name <i>vlan-group-name</i> user-count	Displays the user count for each of the member VLANs of the specified VLAN group This feature counts only authenticated users and MAC addresses added through port security for distribution. It does not consider other learned MAC addresses. As of Cisco IOS Release 12.2(54)SG, the user count for a VLAN is incremented when a host is learned through port security, 802.1X, MAB, or fallback authentication on that VLAN.

The following examples show outputs of the **show vlan group** command:

```
Switch# show vlan group all
Group Name VLANs Mapped
-----
eng-dept      3-4

Switch# show vlan group group-name my_group user-count
      VLAN :    Count
-----
      3      :    1
      4      :    0
      5      :    2
      7      :    0
      9      :    0
Switch#
```

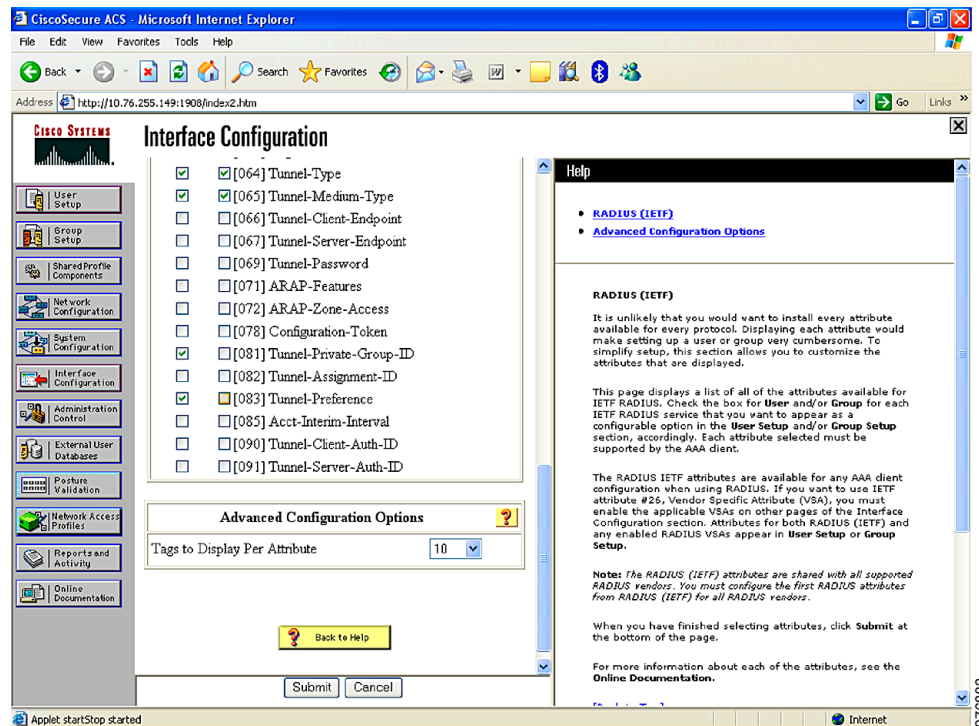
In this example, VLANs 3,4, 5, 7, and 9 are members of the VLAN group *my group*.

ACS Configuration

After configuring the switch, you must provide the VLAN group name in the ACS configuration.

By default, ACS sends only one VLAN name or group per user. However, you can configure ACS to send more than one tag per attribute. To do this, you must modify the configuration in ACS for user or group. (See the example shown in [Figure 49-17](#).)

Figure 49-17 VLAN User Distribution on ACS: Interface Configuration to Modify Tags per Attribute



After you add the number of tags required per attribute, the user or group set up presents multiple fields to be filled with values from the RADIUS server ([Figure 49-18](#)).

Figure 49-18 VLAN User Distribution on ACS: Multiple VLAN Numbers Configured per User

After you complete these two tasks and receive authorization, ACS sends the configured VLAN group to the switch. The switch is alerted to the list of VLANs configured under the VLAN group, and the least loaded valid VLAN in the group is assigned to the port.

Configuring 802.1X with Authentication Failed

By configuring authentication-failed VLAN alignment on any Layer 2 port on the Catalyst 4500 series switch, you can provide limited network services to clients that fail the authentication process.



Note

You can use authentication-failed VLAN assignment with other security features, such as Dynamic ARP Inspection (DAI), Dynamic Host Configuration Protocol (DHCP) snooping, and IP Source Guard. Each of these features can be enabled and disabled independently on the authentication-failed VLAN.

To configure 802.1X with authentication-failed VLAN assignment, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Enters interface configuration mode and specifies the interface to be enabled for 802.1X authentication.
Step 3	Switch(config-if)# switchport mode access	Specifies a nontrunking, nontagged single VLAN Layer 2 interface.
Step 4	Switch(config-if)# authentication port-control auto	Enables 802.1X authentication on the interface.

	Command	Purpose
Step 5	Cisco IOS Release 12.2(50)SG and later Switch(config-if)# authentication event fail action authorize vlan <i>vlan-id</i> Cisco IOS Release 12.2(46)SG or earlier releases Switch(config-if)# dot1x auth-fail vlan <i>vlan-id</i>	Enables authentication-failed VLAN on a particular interface. To disable the authentication-failed VLAN feature on a particular port, use the no authentication event fail action authorize vlan interface configuration command.
Step 6	Cisco IOS Release 12.2(50)SG and later Switch(config-if)# authentication event fail retry <i>max-attempts</i> action [authorize vlan <i>vlan-id</i> next-method] Cisco IOS Release 12.2(46)SG or earlier releases Switch(config-if)# dot1x auth-fail max-attempts <i>max-attempts</i>	Configure a maximum number of attempts before the port is moved to authentication-failed VLAN. Default is 3 attempts.
Step 7	Switch(config-if)# end	Returns to configuration mode.
Step 8	Switch(config)# end	Returns to privileged EXEC mode.
Step 9	Switch# show dot1x interface <i>interface-id</i> details	(Optional) Verifies your entries.
Step 10	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to enable a regular VLAN 40 on Fast Ethernet 4/3 as a authentication-failed VLAN on a static access port:

Cisco IOS Release 12.2(50)SG and later

```
Switch# configure terminal
Switch(config)# interface gigabitEthernet3/1
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# authentication port-control auto
Switch(config-if)# authentication event fail retry 5 action authorize vlan 40
Switch(config-if)# end
Switch# show dot1x all
Sysauthcontrol          Enabled
Dot1x Protocol Version    2

Dot1x Info for GigabitEthernet3/1
-----
PAE                      = AUTHENTICATOR
PortControl              = AUTO
ControlDirection        = Both
HostMode                 = SINGLE_HOST
QuietPeriod              = 60
ServerTimeout            = 0
SuppTimeout              = 30
ReAuthMax                = 2
MaxReq                   = 2
TxPeriod                 = 30
Switch#
```

Cisco IOS Release 12.2(46)SG or earlier

```

Switch# configure terminal
Switch(config)# interface gigabitEthernet3/1
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x port-control auto
Switch(config-if)# dot1x auth-fail vlan 40
Switch(config-if)# dot1x auth-fail max-attempts 3
Switch# show dot1x all
Sysauthcontrol          Enabled
Dot1x Protocol Version    2
Critical Recovery Delay   100
Critical EAPOL            Disabled

Dot1x Info for GigabitEthernet3/1
-----
PAE                      = AUTHENTICATOR
PortControl              = AUTO
ControlDirection        = Both
HostMode                 = SINGLE_HOST
ReAuthentication         = Disabled
QuietPeriod              = 60
ServerTimeout            = 0
SuppTimeout              = 30
ReAuthPeriod             = 3600 (Locally configured)
ReAuthMax                = 2
MaxReq                   = 2
TxPeriod                 = 5
RateLimitPeriod          = 0
Auth-Fail-Vlan           = 40
Auth-Fail-Max-attempts   = 3
Switch#

```

Configuring 802.1X with Voice VLAN



Note

You must configure 802.1X and voice VLAN simultaneously.



Note

You cannot configure an authentication-failed VLAN and a voice VLAN on the same port. When you try to configure these two features on the same port, a syslog message appears.

To enable 802.1X with voice VLAN, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Enters interface configuration mode.
Step 3	Switch(config-if)# switchport access vlan <i>vlan-id</i>	Sets the VLAN for a switched interface in access mode.
Step 4	Switch(config-if)# switchport mode access	Specifies a nontrunking, nontagged single VLAN Layer 2 interface.
Step 5	Switch(config-if)# switchport voice vlan <i>vlan-id</i>	Sets the voice VLAN for the interface.

	Command	Purpose
Step 6	Switch(config-if)# dot1x pae authenticator	Enables 802.1X authentication on the port with default parameters. Refer to the “ Default 802.1X Configuration ” section on page 49-27.
Step 7	Cisco IOS Release 12.2(50)SG and later and later Switch(config-if)# authentication port-control auto Cisco IOS Release 12.2(46)SG or earlier releases Switch(config-if)# dot1x port-control auto	Enables 802.1X authentication on the interface.
Step 8	Switch(config-if)# end	Returns to configuration mode.
Step 9	Switch(config)# end	Returns to privileged EXEC mode.
Step 10	Switch# show dot1x interface interface-id details	(Optional) Verifies your entries.
Step 11	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to enable 802.1X with voice VLAN feature on Fast Ethernet interface 5/9:

Cisco IOS Release 12.2(50)SG and later

```
Switch# configure terminal
Switch(config)# interface fastethernet5/9
Switch(config-if)# switchport access vlan 2
Switch(config-if)# switchport mode access
Switch(config-if)# switchport voice vlan 10
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# authentication port-control auto
Switch(config-if)# end
Switch(config)# end
Switch#
```

Cisco IOS Release 12.2(46)SG or earlier

```
Switch# configure terminal
Switch(config)# interface fastethernet5/9
Switch(config-if)# switchport access vlan 2
Switch(config-if)# switchport mode access
Switch(config-if)# switchport voice vlan 10
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# dot1x port-control auto
Switch(config-if)# end
Switch(config)# end
Switch#
```

Configuring Voice Aware 802.1x Security

You use the voice aware 802.1x security feature on the switch to disable only the VLAN on which a security violation occurs, whether it is a data or voice VLAN. You can use this feature in IP phone deployments where a PC is connected to the IP phone. A security violation found on the data VLAN results in the shutdown of only the data VLAN. The traffic on the voice VLAN flows through the switch without interruption.

Follow these guidelines to configure voice aware 802.1x voice security on the switch:

- You enable voice aware 802.1x security by entering the **errdisable detect cause security-violation shutdown vlan** global configuration command. You disable voice aware 802.1x security by entering the **no** version of this command. This command applies to all 802.1x-configured ports in the switch.



Note

If you do not include the **shutdown vlan** keywords, the entire port is shut down when it enters the error-disabled state.

- If you use the **errdisable recovery cause security-violation** global configuration command to configure error-disabled recovery, the port is automatically re-enabled. If error-disabled recovery is not configured for the port, you re-enable it with the **shutdown** and **no-shutdown** interface configuration commands.
- You can re-enable individual VLANs with the **clear errdisable interface interface-id vlan [vlan-list]** privileged EXEC command. If you do not specify a range, all VLANs on the port are enabled.

To enable voice aware 802.1x security, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# errdisable detect cause security-violation shutdown vlan	Shuts down any VLAN on which a security violation error occurs. Note If the shutdown vlan keywords are not included, the entire port enters the error-disabled state and shuts down.
Step 3	Switch(config)# errdisable recovery cause security-violation	(Optional) Enables automatic per-VLAN error recovery.
Step 4	Switch(config)# errdisable recovery interval interval	(Optional) Sets a recovery interval (in sec). The <i>interval</i> range is 30 to 86400. The default is 300 sec.
Step 5	Switch(config)# end	Enters exec mode.
Step 6	Switch# clear errdisable interface interface-id vlan [vlan-list]	(Optional) Reenables individual VLANs that have been error disabled. <ul style="list-style-type: none"> For <i>interface-id</i> specify the port on which to reenables individual VLANs. (Optional) For <i>vlan-list</i> specify a list of VLANs to be re-enabled. If <i>vlan-list</i> is not specified, all VLANs are re-enabled.
Step 7	Switch(config)# interface interface-id	Enters interface configuration mode.
Step 8	Switch(config-if)# shutdown no-shutdown	(Optional) Re-enables an error-disabled VLAN, and clears all error-disable indications.
Step 9	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 10	Switch# show errdisable detect	Verifies your settings.
Step 11	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

```
Switch# configure terminal
Switch(config)# errdisable detect cause security-violation shutdown vlan
Switch(config)# errdisable recovery cause security-violation
Switch(config)# errdisable recovery interval interval
Switch(config)# end
```

```
Switch# clear errdisable interface interface-id vlan [vlan-list]
Switch(config)# interface interface-id
Switch(config-if)# shutdown
Switch(config-if)# end
Switch# show errdisable detect
Switch# copy running-config startup-config
```

This example shows how to configure the switch to shut down any VLAN on which a security violation error occurs:

```
Switch(config)# errdisable detect cause security-violation shutdown vlan
```

This example shows how to re-enable all VLANs that were error disabled on port Gi4/0/2:

```
Switch# clear errdisable interface GigabitEthernet4/0/2 vlan
```

You can verify your settings by entering the **show errdisable detect** privileged EXEC command.

Configuring 802.1X with VLAN Assignment

For enabling dynamic VLAN assignment, no additional configuration is required in the switch. For information on configuring Multiple- authentication (MDA), refer to the [“Configuring Multiple Domain Authentication and Multiple Authorization”](#) section on page 49-33. To enable VLAN assignment, you must configure the Cisco ACS server.



Note

802.1x authentication with VLAN assignment is not supported on trunk ports, dynamic ports, or with dynamic-access port assignment through a VLAN Membership Policy Server (VMPS).

To enable 802.1X with VLAN assignment, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Enters interface configuration mode.
Step 3	Switch(config-if)# switchport access <i>vlan-id</i>	Sets the VLAN for a switched interface in access mode.
Step 4	Switch(config-if)# switchport mode access	Specifies a nontrunking, nontagged single VLAN Layer 2 interface.
Step 5	Switch(config-if)# switchport voice vlan <i>vlan-id</i>	Sets the voice VLAN for the interface.
Step 6	Switch(config-if)# authentication host-mode multi-domain	Enables MDA on the interface.
Step 7	Switch(config-if)# authentication port-control auto	Enables 802.1X authentication on the interface.
Step 8	Switch(config-if)# dot1x pae authenticator	Enables 802.1X authentication on the port with default parameters. Refer to the “Default 802.1X Configuration” section on page 49-27.
Step 9	Switch(config)# end	Returns to privileged EXEC mode.

	Command	Purpose
Step 10	Switch# show dot1x interface interface-id details	(Optional) Verifies your entries.
Step 11	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

The following example shows how to configure MDA on an interface and 802.1X as the authentication mechanism:

```
Switch(config)# interface FastEthernet3/3
Switch(config-if)# switchport access vlan 10
Switch(config-if)# switchport mode access
Switch(config-if)# switchport voice vlan 16
Switch(config-if)# authentication host-mode multi-domain
Switch(config-if)# authentication port-control auto
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# end
```



Note You must configure VLAN assignment in the ACS server. No configuration changes are required on the switch.

Cisco ACS Configuration for VLAN Assignment

The procedure for enabling MDA with voice VLAN assignment is the same as that for activating MDA except for one step: Configure a VLAN for dynamic VLAN assignment after selecting **User > IETF RADIUS Attributes** (Figure 49-19). This step ensures correct functioning of the ACS configuration required for dynamic VLAN assignment.

Figure 49-19 User Set Up

IETF RADIUS Attributes

☐ [011] Filter-Id

☒ [064] Tunnel-Type

Tag 1 Value VLAN
Tag 2 Value

☒ [065] Tunnel-Medium-Type

Tag 1 Value 802
Tag 2 Value

☒ [081] Tunnel-Private-Group-ID

Tag 1 Value VLAN0020
Tag 2 Value

**Note**

The procedure is the same for voice devices except that the AAA server must be configured to send a Cisco Attribute-Value (AV) pair attribute with a value of device-traffic-class=voice.

Enabling Fallback Authentication

On a port in multiauthentication mode, either or both of MAB and web-based authentication can be configured as fallback authentication methods for non-802.1X hosts (those that do not respond to EAPOL). You can configure the order and priority of the authentication methods.

For detailed configuration information for MAB, see the [“Configuring 802.1X with MAC Authentication Bypass”](#) section on page 49-62.

For detailed configuration information for web-based authentication, see [Chapter 53, “Configuring Web-Based Authentication.”](#)

**Note**

When web-based authentication and other authentication methods are configured on an MDA or multiauthentication port, downloadable ACL policies must be configured for all devices attached to that port.

To enable fallback authentication, perform this task:

	Command	Purpose
Step 1	Switch(config)# ip admission name <i>rule-name</i> proxy http	Configures an authentication rule for web-based authentication.
Step 2	Switch(config)# fallback profile <i>profile-name</i>	Creates a fallback profile for web-based authentication.
Step 3	Switch(config-fallback-profile)# ip access-group <i>rule-name</i> in	Specifies the default ACL to apply to network traffic before web-based authentication.
Step 4	Switch(config-fallback-profile)# ip admission name <i>rule-name</i>	Associates an IP admission rule with the profile and specifies that a client connecting by web-based authentication uses this rule.
Step 5	Switch(config-fallback-profile)# exit	Returns to global configuration mode.
Step 6	Switch(config)# interface <i>type slot/port</i>	Specifies the port to be configured and enters interface configuration mode. <i>type</i> = fastethernet , gigabitethernet , or tengigabitethernet
Step 7	Cisco IOS Release 12.2(50)SG and later Switch(config-if)# authentication port-control auto Cisco IOS Release 12.2(46)SG or earlier releases Switch(config-if)# dot1x port-control auto	Enables authentication on the port.
Step 8	Switch(config-if)# authentication order <i>method1</i> [<i>method2</i>] [<i>method3</i>]	(Optional) Specifies the fallback order of authentication methods to be used. The three values of <i>method</i> , in the default order, are dot1x , mab , and webauth . The specified order also determines the relative priority of the methods for reauthentication (highest to lowest).

	Command	Purpose
Step 9	Switch(config-if)# authentication priority <i>method1 [method2] [method3]</i>	(Optional) Overrides the relative priority of authentication methods to be used. The three values of <i>method</i> , in the default order of priority, are dot1x , mab , and webauth .
Step 10	Switch(config-if)# authentication event fail action next-method	Specifies that the next configured authentication method be applied if authentication fails.
Step 11	Cisco IOS Release 12.2(50)SG and later Switch(config-if)# mab [eap] Cisco IOS Release 12.2(46)SG or earlier releases Switch(config-if)# dot1x mac-auth-bypass [eap]	Enables MAC authentication bypass. The optional eap keyword specifies that the EAP extension be used during RADIUS authentication.
Step 12	Switch(config-if)# authentication fallback <i>profile-name</i>	Enables web-based authentication using the specified profile.
Step 13	Switch(config-if)# authentication violation [shutdown restrict]	(Optional) Configures the disposition of the port if a security violation occurs. The default action is to shut down the port. If the restrict keyword is configured, the port does not shut down, but trap entries are installed for the violating MAC address, and traffic from that MAC address is dropped.
Step 14	Switch(config-if)# authentication timer inactivity { <i>seconds</i> server }	(Optional) Configures the inactivity timeout value for MAB and 802.1X. By default, inactivity aging is disabled for a port. <ul style="list-style-type: none"> <i>seconds</i>—Specifies inactivity timeout period. The range is from 1 to 65535 seconds. server—Specifies that the inactivity timeout period value be obtained from the authentication server.
Step 15	Switch(config-if)# authentication timer restart <i>seconds</i>	(Optional) Specifies a period after which the authentication process restarts in an attempt to authenticate an unauthorized port. <ul style="list-style-type: none"> <i>seconds</i>—Specifies the restart period. The range is from 1 to 65535 seconds.
Step 16	Switch(config-if)# exit	Returns to global configuration mode.
Step 17	Switch(config)# ip device tracking	Enables the IP device tracking table, which is required for web-based authentication.
Step 18	Switch(config)# exit	Returns to privileged EXEC mode.
Step 19	Switch# show dot1x interface <i>type slot/port</i>	Verifies your entries.

This example shows how to enable 802.1X fallback to MAB, and then to enable web-based authentication, on an 802.1X-enabled port:

```
Switch(config)# ip admission name rule1 proxy http
Switch(config)# fallback profile fallback1
Switch(config-fallback-profile)# ip access-group default-policy in
Switch(config-fallback-profile)# ip admission rule1
Switch(config-fallback-profile)# exit
Switch(config)# interface gigabit5/9
```

```

Switch(config-if)# switchport mode access
Switch(config-if)# authentication port-control auto
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# authentication order dot1x mab webauth
Switch(config-if)# mab eap
Switch(config-if)# authentication fallback fallback1
Switch(config-if)# exit
Switch(config)# ip device tracking
Switch(config)# exit

```

To determine if a host was authenticated using 802.1X when fallback authentication is configured on the port, enter the following commands:

```
Switch# show authentication sessions interface g7/2
```

```

Interface: GigabitEthernet7/2
MAC Address: 0060.b057.4687
IP Address: Unknown
User-Name: test2
Status: Authz Success
Domain: DATA
Oper host mode: multi-auth
Oper control dir: both
Authorized By: Authentication Server
Vlan Policy: N/A
Session timeout: N/A
Idle timeout: N/A
Common Session ID: C0A8013F0000000901BAB560
Acct Session ID: 0x0000000B
Handle: 0xE8000009

```

```
Runnable methods list:
```

```

Method   State
dot1x    Authc Success
mab       Not run

```

```
Switch# show dot1x interfaces g7/2 detail
```

```
Dot1x Info for GigabitEthernet7/2
```

```

-----
PAE = AUTHENTICATOR
PortControl = AUTO
ControlDirection = Both
HostMode = MULTI_AUTH
QuietPeriod = 60
ServerTimeout = 0
SuppTimeout = 30
ReAuthMax = 2
MaxReq = 2
TxPeriod = 2

```

```
Dot1x Authenticator Client List
```

```

-----
Supplicant = 0060.b057.4687
Session ID = C0A8013F0000000901BAB560
Auth SM State = AUTHENTICATED
Auth BEND SM State = IDLE
Port Status = AUTHORIZED

```

To determine if a host was authenticated using MAB when fallback authentication is configured on the port, enter the following commands:

```
Switch# show authentication sessions interface g7/2
```

```

      Interface: GigabitEthernet7/2
      MAC Address: 0060.b057.4687
      IP Address: 192.168.22.22
      User-Name: 0060b0574687
      Status: Authz Success
      Domain: DATA
      Oper host mode: multi-auth
      Oper control dir: both
      Authorized By: Authentication Server
      Vlan Policy: N/A
      Session timeout: N/A
      Idle timeout: N/A
      Common Session ID: C0A8013F0000000B01BBD278
      Acct Session ID: 0x0000000D
      Handle: 0xF500000B

```

```
Runnable methods list:
```

```

      Method  State
      dot1x   Failed over
      mab      Authc Success

```

```
Switch# show mab interface g7/2 detail
```

```
MAB details for GigabitEthernet7/2
```

```
-----
Mac-Auth-Bypass           = Enabled

```

```
MAB Client List
```

```

-----
Client MAC                 = 0060.b057.4687
Session ID                 = C0A8013F0000000B01BBD278
MAB SM state               = TERMINATE
Auth Status                = AUTHORIZED

```

To determine if a host was authenticated using web authentication when fallback authentication is configured on the port, enter the following commands:

```
Switch# show authentication sessions interface G4/3
```

```

      Interface: GigabitEthernet4/3
      MAC Address: 0015.e981.0531
      IP Address: 10.5.63.13
      Status: Authz Success
      Domain: DATA
      Oper host mode: single-host
      Oper control dir: both
      Authorized By: Authentication Server
      Vlan Policy: N/A
      Session timeout: N/A
      Idle timeout: N/A
      Common Session ID: 0A053F0F0000000200112FFC
      Acct Session ID: 0x00000003
      Handle: 0x09000002

```

```
Runnable methods list:
```

```

      Method  State
      dot1x   Failed over
      mab      Failed over
      webauth  Authc Success

```



```
Switch# show ip admission cache
Authentication Proxy Cache
Total Sessions: 1 Init Sessions: 0
Client IP 10.5.63.13 Port 4643, timeout 1000, state ESTAB
```

Cisco IOS Release 12.2(46)SG or earlier

```
Switch(config)# ip admission name rule1 proxy http
Switch(config)# fallback profile fallback1
Switch(config-fallback-profile)# ip access-group default-policy in
Switch(config-fallback-profile)# ip admission rule1
Switch(config-fallback-profile)# exit
Switch(config)# interface gigabit5/9
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x port-control auto
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# authentication order dot1x mab webauth
Switch(config-if)# dot1x mac-auth-bypass eap
Switch(config-if)# adot1x fallback fallback1
Switch(config-if)# exit
Switch(config)# ip device tracking
Switch(config)# exit
```

Enabling Periodic Reauthentication

You can enable periodic 802.1X client reauthentication and specify how often it occurs. If you do not specify a time value before enabling reauthentication, the interval between reauthentication attempts is 3600 seconds.

Automatic 802.1X client reauthentication is a per-interface setting and can be set for clients connected to individual ports. To manually reauthenticate the client connected to a specific port, see the [“Changing the Quiet Period”](#) section on page 49-86.

To enable periodic reauthentication of the client and to configure the number of seconds between reauthentication attempts, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Enters interface configuration mode and specifies the interface to be enabled for periodic reauthentication.
Step 3	Switch(config-if)# switchport mode access	Specifies a nontrunking, nontagged single VLAN Layer 2 interface.
Step 4	Switch(config-if)# dot1x pae authenticator	Enables 802.1X authentication on the port with default parameters. Refer to the “Default 802.1X Configuration” section on page 49-27.
Step 5	Cisco IOS Release 12.2(50)SG and later Switch(config-if)# authentication periodic Cisco IOS Release 12.2(46)SG or earlier releases Switch(config-if)# dot1x reauthentication	Enables periodic reauthentication of the client, which is disabled by default. To disable periodic reauthentication, use the no authentication periodic interface configuration command (for earlier releases, use the no dot1x reauthentication interface configuration command).

	Command	Purpose
Step 6	Cisco IOS Release 12.2(50)SG and later Switch(config-if)# authentication timer reauthenticate {seconds / server}	Specifies the number of seconds between reauthentication attempts or have the switch use a RADIUS-provided session timeout. The range is 1 to 65,535; the default is 3600 seconds.
	Cisco IOS Release 12.2(46)SG or earlier releases Switch(config-if)# dot1x timeout reauth-period {seconds / server}	To return to the default number of seconds between reauthentication attempts, use the no authentication timer reauthenticate global configuration command (for earlier releases, use the dot1x timeout reauth-attempts command). This command affects the behavior of the switch only if periodic reauthentication is enabled.
Step 7	Cisco IOS Release 12.2(50)SG and later Switch(config-if)# authentication port-control auto Cisco IOS Release 12.2(46)SG or earlier releases Switch(config-if)# dot1x port-control auto	Enables 802.1X authentication on the interface.
Step 8	Switch(config-if)# end	Returns to privileged EXEC mode.

This example shows how to enable periodic reauthentication and set the number of seconds between reauthentication attempts to 4000:

Cisco IOS Release 12.2(50)SG and later

```
Switch# configure terminal
Switch(config)# interface fastethernet5/9
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# authentication periodic
Switch(config-if)# authentication timer reauthenticate 4000
Switch(config-if)# authentication port-control auto
Switch(config-if)# end
Switch#
```

Cisco IOS Release 12.2(46)SG or earlier

```
Switch# configure terminal
Switch(config)# interface fastethernet5/9
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# dot1x reauthentication
Switch(config-if)# dot1x timeout reauth-period 4000
Switch(config-if)# dot1x port-control auto
Switch(config-if)# end
Switch#
```

Enabling Multiple Hosts

You can attach multiple hosts (clients) to a single 802.1X-enabled port as shown in [Figure 49-9 on page 49-25](#). In this mode, when the port is authorized, all other hosts that are indirectly attached to the port are granted access to the network. If the port becomes unauthorized (reauthentication fails or an EAPOL-logoff message is received), the switch denies access to the network for all wireless access point-attached clients.

To allow multiple hosts (clients) on an 802.1X-authorized port that has the **dot1x port-control** interface configuration command set to **auto**, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Enters interface configuration mode and specifies the interface to which multiple hosts are indirectly attached.
Step 3	Switch(config-if)# switchport mode access	Specifies a nontrunking, nontagged single VLAN Layer 2 interface.
Step 4	Switch(config-if)# dot1x pae authenticator	Enables 802.1X authentication on the port with default parameters. Refer to the “ Default 802.1X Configuration ” section on page 49-27.
Step 5	Cisco IOS Release 12.2(50)SG and later Switch(config-if)# authentication host-mode multi-host Cisco IOS Release 12.2(46)SG or earlier releases Switch(config-if)# dot1x host-mode multi-host	Allows multiple hosts (clients) on an 802.1X-authorized port. Note Ensure that the dot1x port-control interface configuration command set is set to auto for the specified interface. To disable multiple hosts on the port, use the no authentication host-mode multi-host interface configuration command (for earlier releases, use the no dot1x host-mode multi-host interface configuration command).
Step 6	Cisco IOS Release 12.2(50)SG and later Switch(config-if)# authentication port-control auto Cisco IOS Release 12.2(46)SG or earlier releases Switch(config-if)# dot1x port-control auto	Enables 802.1X authentication on the interface.
Step 7	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 8	Switch# show dot1x all interface <i>interface-id</i>	Verifies your entries.
Step 9	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to enable 802.1X on Fast Ethernet interface 5/9 and to allow multiple hosts:

Cisco IOS Release 12.2(50)SG and later

```
Switch# configure terminal
Switch(config)# interface fastethernet5/9
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# authentication host-mode multi-host
Switch(config-if)# authentication port-control auto
Switch(config-if)# end
Switch#
```

Cisco IOS Release 12.2(46)SG or earlier

```
Switch# configure terminal
Switch(config)# interface fastethernet5/9
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
```

```
Switch(config-if)# dot1x host-mode multi-host
Switch(config-if)# dot1x port-control auto
Switch(config-if)# end
Switch#
```

Changing the Quiet Period

When the switch cannot authenticate the client, the switch remains idle for a set period of time, and then tries again. The idle time is determined by the **quiet-period** value. A failed authentication of the client might occur because the client provided an invalid password. You can provide a faster response time to the user by entering a number smaller than the default.

To change the quiet period, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Enters interface configuration mode and specifies the interface to be enabled for timeout quiet-period .
Step 3	Switch(config-if)# switchport mode access	Specifies a nontrunking, nontagged single VLAN Layer 2 interface.
Step 4	Switch(config-if)# dot1x pae authenticator	Enables 802.1X authentication on the port with default parameters. Refer to the “Default 802.1X Configuration” section on page 49-27.
Step 5	Switch(config-if)# dot1x timeout quiet-period <i>seconds</i>	Sets the number of seconds that the switch remains in the quiet-period following a failed authentication exchange with the client. To return to the default quiet-period, use the no dot1x timeout quiet-period configuration command. The range is 0 to 65,535 seconds; the default is 60.
Step 6	Switch(config-if)# authentication port-control auto	Enables 802.1X authentication on the interface.
Step 7	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 8	Switch# show dot1x all	Verifies your entries.
Step 9	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to set the quiet period on the switch to 30 seconds:

Cisco IOS Release 12.2(50)SG and later

```
Switch# configure terminal
Switch(config)# interface fastethernet4/1
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# dot1x timeout quiet-period 30
Switch(config-if)# authentication port-control auto
Switch(config-if)# end
Switch#
```

Cisco IOS Release 12.2(46)SG or earlier

```
Switch# configure terminal
Switch(config)# interface fastethernet4/1
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# dot1x timeout quiet-period 30
Switch(config-if)# dot1x port-control auto
Switch(config-if)# end
Switch#
```

Changing the Switch-to-Client Retransmission Time

The client responds to the EAP-request/identity frame from the switch with an EAP-response/identity frame. If the switch does not receive this response, it waits a set period of time (known as the retransmission time) and then retransmits the frame.

**Note**

You should change the default value of this command only to adjust for unusual circumstances, such as unreliable links or specific behavioral problems with certain clients and authentication servers.

To change the amount of time that the switch waits for client notification, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Enters interface configuration mode and specifies the interface to be enabled for timeout tx-period.
Step 3	Switch(config-if)# switchport mode access	Specifies a nontrunking, nontagged single VLAN Layer 2 interface.
Step 4	Switch(config-if)# dot1x pae authenticator	Enables 802.1X authentication on the port with default parameters. Refer to the “Default 802.1X Configuration” section on page 49-27.
Step 5	Switch(config-if)# dot1x timeout tx-period <i>seconds</i>	Sets the number of seconds that the switch waits for a response to an EAP-request/identity frame from the client before retransmitting the request. The range is 1 to 65,535 seconds; the default is 30. To return to the default retransmission time, use the no dot1x timeout tx-period interface configuration command.
Step 6	Cisco IOS Release 12.2(50)SG and later Switch(config-if)# authentication port-control auto Cisco IOS Release 12.2(46)SG or earlier releases Switch(config-if)# dot1x port-control auto	Enables 802.1X authentication on the interface.
Step 7	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 8	Switch# show dot1x all	Verifies your entries.
Step 9	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to set the retransmission time to 60 seconds:

Cisco IOS Release 12.2(50)SG and later

```
Switch# configure terminal
Switch(config)# interface fastethernet5/9
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# dot1x timeout tx-period 60
Switch(config-if)# authentication port-control auto
Switch(config-if)# end
Switch#
```

Cisco IOS Release 12.2(46)SG or earlier

```
Switch# configure terminal
Switch(config)# interface fastethernet5/9
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# dot1x timeout tx-period 60
Switch(config-if)# dot1x port-control auto
Switch(config-if)# end
Switch#
```

Setting the Switch-to-Client Frame-Retransmission Number

In addition to changing the switch-to-client retransmission times, you can change the number of times that the switch sends EAP-Request/Identity and other EAP-Request frames to the client before restarting the authentication process. The number of EAP-Request/Identity retransmissions is controlled by the **dot1x max-reauth-req** command; the number of retransmissions for other EAP-Request frames is controlled by the **dot1x max-req** command.



Note

You should change the default values of these commands only to adjust for unusual circumstances such as unreliable links or specific behavioral problems with certain clients and authentication servers.

To set the switch-to-client frame-retransmission numbers, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Enters interface configuration mode and specifies the interface to be enabled for max-reauth-req and/or max-req .
Step 3	Switch(config-if)# switchport mode access	Specifies a nontrunking, nontagged single VLAN Layer 2 interface.
Step 4	Switch(config-if)# dot1x pae authenticator	Enables 802.1X authentication on the port with default parameters. Refer to the “Default 802.1X Configuration” section on page 49-27.

	Command	Purpose
Step 5	Switch(config-if) # dot1x max-req count	Specifies the number of times EAPOL DATA packets are retransmitted (if lost or not replied to). For example, if you have a supplicant that is authenticating and it experiences a problem, the authenticator retransmits requests for data three times before abandoning the authentication request. The range for <i>count</i> is 1 to 10; the default is 2.
	or Switch(config-if) # dot1x max-reauth-req count	Specifies the timer for EAPOL-Identity-Request frames (only). If you plug in a device incapable of 802.1X, three EAPOL-Id-Req frames are sent before the state machine resets. Alternatively, if you have configured Guest-VLAN, three frames are sent before the port is enabled. This parameter has a default value of 2. To return to the default retransmission number, use the no dot1x max-req and no dot1x max-reauth-req global configuration command.
Step 6	Switch(config-if) # authentication port-control auto	Enables 802.1X authentication on the interface.
Step 7	Switch(config-if) # end	Returns to privileged EXEC mode.
Step 8	Switch# show dot1x all	Verifies your entries.
Step 9	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to set 5 as the number of times that the switch retransmits an EAP-request/identity request before restarting the authentication process:

Cisco IOS Release 12.2(50)SG and later

```
Switch# configure terminal
Switch(config)# interface fastethernet5/9
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# dot1x max-reauth-req 5
Switch(config-if)# authentication port-control auto
Switch(config-if)# end
Switch#
```

Cisco IOS Release 12.2(46)SG or earlier

```
Switch# configure terminal
Switch(config)# interface fastethernet5/9
Switch(config-if)# switchport mode access
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# dot1x max-reauth-req 5
Switch(config-if)# dot1x port-control auto
Switch(config-if)# end
Switch#
```

Configuring an Authenticator and a Supplicant Switch with NEAT

Configuring NEAT requires that one switch outside a wiring closet is configured as a supplicant and is connected to an authenticator switch.

This section includes these topics:

- [Configuring Switch as an Authenticator, page 49-90](#)
- [Configuring Switch as a Supplicant, page 49-93](#)

- [Configuring NEAT with ASP, page 49-94](#)
- [Configuration Guidelines, page 49-94](#)


Note

For overview information, see the “[802.1X Supplicant and Authenticator Switches with Network Edge Access Topology](#)” section on page 49-23.

Configuring Switch as an Authenticator

To configure a switch as an authenticator, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# cisp enable	Enables CISP.
Step 3	Switch(config)# interface <i>interface-id</i>	Specifies the port to be configured, and enter interface configuration mode.
Step 4	Switch(config-if)# switchport mode access	Sets the port mode to access.
Step 5	Switch(config-if)# authentication port-control auto	Sets the port-authentication mode to auto.
Step 6	Switch(config-if)# dot1x pae authenticator	Configures the interface as a port access entity (PAE) authenticator.
Step 7	Switch(config-if)# spanning-tree portfast	Enables Port Fast on an access port connected to a single workstation or server.
Step 8	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 9	Switch# show running-config interface interface-id	Verifies your configuration.
Step 10	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

When CISP is enabled on a trunk port, the following features are inert. When CISP is neither running nor configured, these features operate as expected:

- VLAN assignment
- Guest, Authentication Failure, voice, and critical VLANs
- Critical authentication
- Wake-on-LAN
- Web authentication
- Port security
- Violation modes (restrict, shut down, and shut down VLAN)

The following example shows how to enable CISP on a port. You must configure the following procedure in the Cisco ACS server. Configuring a user with Cisco AV Pair value, allows SSW to authenticate itself with the ASW. Because the user is attached with the AV pair value, upon successful authentication on ASW, the macro is executed on the interface on which SSW is authenticated:

```
Switch# configure terminal
Switch(config)# cisp enable
```



```
Switch(config)# interface GigabitEthernet5/23
Switch(config-if)# switchport mode access
Switch(config-if)# spanning-tree portfast
Switch(config-if)# dot1x pae authenticator
Switch(config-if)# authentication port-control auto
```

Cisco AV Pair Configuration

Next, you need to configure a Cisco AV pair value.

Log into ACS, and Select/Create a User. Go to User Setup and scroll down to the [009\001] cisco-av-pair Tab. Enter `device-traffic-class=switch` (Figure 49-20).

Figure 49-20 Specifying the Cisco AV Pair

Starting with Cisco IOS XE Release 3.2.0 SG (15.0(2)SG) the spanning-tree bpduguard feature is automatically disabled or enabled as part of a macro provided it was previously enabled in the port configuration. If the configuration did not have BPDU Guard enabled before the supplicant switch was authenticated, the spanning-tree bpduguard feature is not applied to the macro.



Note

Disabling spanning-tree bpduguard happens only if it was previously enabled through the **port level** command. Enabling it globally without a specific port level CLI prevents NEAT from disabling it on the port after the authenticator switch receives a `device-traffic-class=switch` AV Pair and applies the macro.

There are 2 scenarios:

Scenario 1: With Port Level BPDU Guard Configuration

Before Authorization

```

interface GigabitEthernet5/1
  switchport access vlan 81
  switchport mode access
  dot1x pae authenticator
  authentication port-control auto
  spanning-tree bpduguard enable
end

```

Post Authorization and Application of Internal Macro

```

interface GigabitEthernet5/1
  switchport trunk encapsulation dot1q
  switchport trunk native vlan 81
  switchport mode trunk
  dot1x pae authenticator
  authentication port-control auto
  spanning-tree portfast trunk
  no spanning-tree bpduguard
end

```

Scenario 2: Without port level BPDU Guard Configuration (with or without globally enabling BPDU Guard)

Before Authorization

```

interface GigabitEthernet5/1
  switchport access vlan 81
  switchport mode access
  dot1x pae authenticator
  authentication port-control auto
end

```

Post Authorization and Application of Internal Macro

```

interface GigabitEthernet5/1
  switchport trunk encapsulation dot1q
  switchport trunk native vlan 81
  switchport mode trunk
  dot1x pae authenticator
  authentication port-control auto
  spanning-tree portfast trunk
  no spanning-tree bpduguard
end

```

When the authenticator switch receives a device-traffic-class=switch AV pair, the following macro is applied to the authenticator switch port:

```

no switchport access vlan $AVID
no switchport nonegotiate
switchport mode trunk
switchport trunk native vlan $AVID
no spanning-tree bpduguard enable
spanning-tree portfast trunk

```

After the supplicant switch is authenticated as a switch device, the configuration will appear as follows:

```

interface GigabitEthernet5/23
  switchport mode trunk
  authentication port-control auto
  dot1x pae authenticator
  spanning-tree portfast trunk
end

```

Radius Config (Cisco AV Pair value)

```
-----
device-traffic-class=switch
```

show running-config interface is the only command that informs you that the smart macro has been applied after the supplicant switch is authenticated:

```
Switch# show authentication session
```

```
Interface  MAC Address      Method  Domain  Status      Session ID
Gi5/23     0024.9844.de23  dot1x   DATA   Authz Success  0909117A0000000000010561C
```

```
Switch# show running-configuration interface gi 5/23
```

```
Building configuration...
```

```
Current configuration : 149 bytes
```

```
!
```

```
interface GigabitEthernet5/23
 switchport mode trunk
 authentication port-control auto
 dot1x pae authenticator
 spanning-tree portfast trunk
end
```

```
Switch#
```

NEAT changes the port configuration on the authenticator switch. So, to perform ISSU from one version that supports NEAT to another that does not support NEAT, you must first deactivate NEAT on all switch ports for ISSU. Similarly, NEAT cannot activate when ISSU is in progress. If a supplicant switch tries to authenticate during ISSU, authorization would fail on the port.

Configuring Switch as a Supplicant



Note

The Catalyst 4500 series switch does not support supplicant switch functionality. The following supplicant specific commands are mentioned for a quick reference. For more details, see the *Catalyst 3750 Switch Software Configuration Guide*.

To configure a switch as a supplicant, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# cisp enable	Enables CISP.
Step 3	Switch(config)# dot1x credentials profile	Creates 802.1x credentials profile. This must be attached to the port that is configured as supplicant.
Step 4	Switch(config)# username suppswitch	Creates a username.
Step 5	Switch(config)# password password	Creates a password for the new username.
Step 6	Switch(config)# dot1x supplicant force-multicast	Forces the switch to send <i>only</i> multicast EAPOL packets when it receives either unicast or multicast packets. This also allows NEAT to work on the supplicant switch in all host modes.

	Command	Purpose
Step 7	Switch(config)# interface <i>interface-id</i>	Specifies the port to be configured, and enter interface configuration mode.
Step 8	Switch(config-if)# switchport trunk encapsulation dot1q	Sets the port to trunk mode.
Step 9	Switch(config-if)# switchport mode trunk	Configures the interface as a VLAN trunk port.
Step 10	Switch(config-if)# dot1x pae supplicant	Configures the interface as a port access entity (PAE) supplicant.
Step 11	Switch(config-if)# dot1x credentials profile-name	Attaches the 802.1x credentials profile to the interface.
Step 12	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 13	Switch# show running-config interface interface	Verifies your configuration. Note it is the only command that tells you that the smart macro has been applied after the supplicant switch has been authenticated.
Step 14	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to configure a switch as a supplicant:

```
Switch# configure terminal
Switch(config)# cisp enable
Switch(config)# dot1x credentials test
Switch(config)# username suppswitch
Switch(config)# password myswitch
Switch(config)# dot1x supplicant force-multicast
Switch(config)# interface gigabitethernet1/0/1
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# dot1x pae supplicant
Switch(config-if)# dot1x credentials test
Switch(config-if)# end
```

The following macro is applied to the authenticator switch port after the supplicant switch is deauthenticated due to a link-down or a reauthenticating event:

```
no switchport nonegotiate
switchport mode access
no switchport trunk native vlan $AVID
no spanning-tree portfast trunk
switchport access vlan $AVID
spanning-tree bpduguard enable
spanning-tree portfast
```

Configuring NEAT with ASP

You can also use an AutoSmart Ports user-defined macro rather than a switch VSA to configure the authenticator switch. For more information, see the [Chapter 22, “Configuring Cisco IOS Auto Smartport Macros.”](#)

Configuration Guidelines

- If BPDU Guard was enabled prior to supplicant switch authentication, it is re-enabled after the supplicant switch unauthenticates.

- You can configure NEAT ports and non-NEAT ports with the same configuration. When the supplicant switch authenticates, the port mode is changed from access to trunk based on the switch vendor-specific attributes (`device-traffic-class=switch`).
- To enable NEAT, you must configure the vendor-specific attributes (VSA) attribute as switch. Configuring the trunk with an 802.1X configuration and enabling CISP globally will not enable NEAT.
- VSA `device-traffic-class=switch` assists the authenticator switch in identifying the supplicant as a switch-device. This identification changes the authenticator switch port mode from access to trunk and enables 802.1X trunk encapsulation. The access VLAN, if any, is converted to a native trunk VLAN. VSA does not change any of the port configurations on the supplicant.
- Although modified trunk parameters are retained, when the trunk link is down or authentication is cleared, the interface is reconfigured to the following:
 - **spanning-tree portfast**
 - **switchport mode access**
 - **switchport access vlan** *access-vlan-id*



Note *access-vlan-id* is derived from the **switchport trunk native vlan** *x* command entered on the interface. If you have modified the trunk native VLAN, the configured native VLAN is used as the access-vlan-id when the port returns to access mode.

- We recommend using 802.1X authentication mode single-host for NEAT configuration on the interface.
- The `cisco-av-pairs` must be configured as `device-traffic-class=switch` on the ACS. This sets the interface as a trunk after the supplicant is successfully authenticated.
- You should not modify the trunk mode configurations that are based on *device-traffic-class* either manually or through features such as AutoSmart Ports. It is because 802.1X configuration is not supported for trunk ports.
- To change the host mode and apply a standard port configuration on the authenticator switch port, you can also use AutoSmart ports user-defined macros rather than the switch VSA. Doing this allows you to remove unsupported configurations on the authenticator switch port and to change the port mode from access to trunk. For details, see [Chapter 22, “Configuring Cisco IOS Auto Smartport Macros.”](#)



Note Configuring only the Auto SmartPorts macro does not identify the end host as a supplicant switch. The switch VSA is required to identify the supplicant switch. However, when Auto Smartports macro is configured, the internal macro that reconfigures the port from access to trunk is not executed and the Auto Smartports macro should ensure that the port reconfigures as a trunk port.

Manually Reauthenticating a Client Connected to a Port

You can manually reauthenticate a client connected to a specific port at any time by entering the **dot1x re-authenticate interface** privileged EXEC command. If you want to enable or disable periodic reauthentication, see the [“Enabling Periodic Reauthentication” section on page 49-83](#).

This example shows how to manually reauthenticate the client connected to Fast Ethernet port 1/1:

```
Switch# dot1x re-authenticate interface fastethernet1/1
Starting reauthentication on FastEthernet1/1
```

Initializing the 802.1X Authentication State

The **dot1x initialize** command causes the authentication process to be restarted regardless of its current state.

This example shows how to restart the authentication process on Fast Ethernet port 1/1:

```
Switch# dot1x initialize interface fastethernet1/1
```

This example shows how to restart the authentication process on all ports of the switch:

```
Switch# dot1x initialize
```

Removing 802.1X Client Information

The **clear dot1x** command causes all existing supplicants to be completely deleted from an interface or from all the interfaces on a switch.

This example shows how to remove 802.1X client information on Fast Ethernet port 1/1:

```
Switch# clear dot1x interface fastethernet1/1
```

This example shows how to remove 802.1X client information on all ports of the switch:

```
Switch# clear dot1x all
```

Resetting the 802.1X Configuration to the Default Values

To reset the 802.1X configuration to the default values, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# dot1x default	Resets the configurable 802.1X parameters to the default values.
Step 3	Switch(config)# end	Returns to privileged EXEC mode.
Step 4	Switch# show dot1x all	Verifies your entries.
Step 5	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Controlling Switch Access with RADIUS

This section describes how to enable and configure the RADIUS, which provides detailed accounting information and flexible administrative control over authentication and authorization processes. RADIUS is facilitated through AAA and can be enabled only through AAA commands.

**Note**

For complete syntax and usage information for the commands used in this section, see the *Cisco IOS Security Command Reference, Release 12.2*.

These sections contain this configuration information:

- [Understanding RADIUS, page 49-97](#)
- [RADIUS Operation, page 49-98](#)
- [RADIUS Change of Authorization, page 49-99](#)
- [Configuring RADIUS, page 49-104](#)
- [Displaying the RADIUS Configuration, page 49-117](#)

Understanding RADIUS

RADIUS is a distributed client/server system that secures networks against unauthorized access. RADIUS clients run on supported Cisco routers and switches. Clients send authentication requests to a central RADIUS server, which contains all user authentication and network service access information. The RADIUS host is normally a multiuser system running RADIUS server software from Cisco (Cisco Secure Access Control Server Version 3.0), Livingston, Merit, Microsoft, or another software provider. For more information, see the RADIUS server documentation.

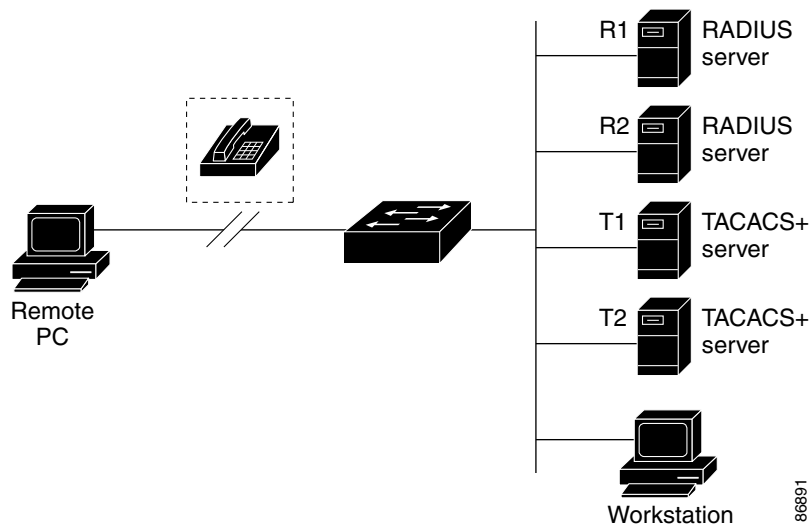
Use RADIUS in these network environments that require access security:

- Networks with multiple-vendor access servers, each supporting RADIUS. For example, access servers from several vendors use a single RADIUS server-based security database. In an IP-based network with multiple vendors' access servers, dial-in users are authenticated through a RADIUS server that has been customized to work with the Kerberos security system.
- Turnkey network security environments in which applications support the RADIUS protocol, such as in an access environment that uses a *smart card* access control system. In one case, RADIUS has been used with Enigma's security cards to validate users and to grant access to network resources.
- Networks already using RADIUS. You can add a Cisco switch containing a RADIUS client to the network. This might be the first step when you make a transition to a TACACS+ server. See [Figure 49-21 on page 49-98](#).
- Network in which the user must only access a single service. Using RADIUS, you can control user access to a single host, to a single utility such as Telnet, or to the network through a protocol such as IEEE 802.1X.
- Networks that require resource accounting. You can use RADIUS accounting independently of RADIUS authentication or authorization. The RADIUS accounting functions allow data to be sent at the start and end of services, showing the amount of resources (such as time, packets, bytes, and so forth) used during the session. An Internet service provider might use a freeware-based version of RADIUS access control and accounting software to meet special security and billing needs.

RADIUS is not suitable in these network security situations:

- Multiprotocol access environments. RADIUS does not support AppleTalk Remote Access (ARA), NetBIOS Frame Control Protocol (NBFCP), NetWare Asynchronous Services Interface (NASI), or X.25 PAD connections.
- Switch-to-switch or router-to-router situations. RADIUS does not provide two-way authentication. RADIUS can be used to authenticate from one device to a non-Cisco device if the non-Cisco device requires authentication.
- Networks using a variety of services. RADIUS generally binds a user to one service model.

Figure 49-21 Transitioning from RADIUS to TACACS+ Services



RADIUS Operation

When a user attempts to log in and authenticate to a switch that is access controlled by a RADIUS server, these events occur:

1. The user is prompted to enter a username and password.
2. The username and encrypted password are sent over the network to the RADIUS server.
3. The user receives one of these responses from the RADIUS server:
 - a. ACCEPT—The user is authenticated.
 - b. REJECT—The user is either not authenticated and is prompted to re-enter the username and password, or access is denied.
 - c. CHALLENGE—A challenge requires additional data from the user.
 - d. CHALLENGE PASSWORD—A response requests the user to select a new password.

The ACCEPT or REJECT response is bundled with additional data that is used for privileged EXEC or network authorization. Users must first successfully complete RADIUS authentication before proceeding to RADIUS authorization, if it is enabled. The additional data included with the ACCEPT or REJECT packets includes these items:

- Telnet, SSH, rlogin, or privileged EXEC services
- Connection parameters, including the host or client IP address, access list, and user timeouts

RADIUS Change of Authorization

This section provides an overview of the RADIUS interface including available primitives and how they are used during a Change of Authorization (CoA).

- [Overview, page 49-99](#)
- [Change-of-Authorization Requests, page 49-99](#)
- [CoA Request Response Code, page 49-100](#)
- [CoA Request Commands, page 49-101](#)
- [Session Reauthentication, page 49-102](#)
- [Displaying 802.1X Statistics and Status, page 49-126](#)

Overview

A standard RADIUS interface is typically used in a pulled model where the request originates from a network attached device and the response come from the queried servers. Catalyst switches support the RADIUS Change of Authorization (CoA) extensions defined in RFC 5176 that are typically used in a pushed model and allow for the dynamic reconfiguring of sessions from external authentication, authorization, and accounting (AAA) or policy servers.

The switch supports these per-session CoA requests:

- Session reauthentication
- Session termination
- Session termination with port shut down
- Session termination with port bounce

The RADIUS interface is enabled by default on Catalyst switches.

Change-of-Authorization Requests

Change of Authorization (CoA) requests, as described in RFC 5176, are used in a push model to allow for session identification, host reauthentication, and session termination. The model is comprised of one request (CoA-Request) and two possible response codes:

- CoA acknowledgement (ACK) [CoA-ACK]
- CoA non-acknowledgement (NAK) [CoA-NAK]

The request is initiated from a CoA client (typically a RADIUS or policy server) and directed to the switch that acts as a listener.

This section includes these topics:

- [CoA Request Response Code](#)
- [CoA Request Commands](#)
- [Session Reauthentication](#)

RFC 5176 Compliance

The Disconnect Request message, which is also referred to as Packet of Disconnect (POD), is supported by the switch for session termination.

Table 49-2 shows the IETF attributes are supported for this feature.

Table 49-2 Supported IETF Attributes

Attribute Number	Attribute Name
24	State
31	Calling-Station-ID
44	Acct-Session-ID
80	Message-Authenticator
101	Error-Cause

Table 49-3 shows the possible values for the Error-Cause attribute.

Table 49-3 Error-Cause Values

Value	Explanation
201	Residual Session Context Removed
202	Invalid EAP Packet (Ignored)
401	Unsupported Attribute
402	Missing Attribute
403	NAS Identification Mismatch
404	Invalid Request
405	Unsupported Service
406	Unsupported Extension
407	Invalid Attribute Value
501	Administratively Prohibited
502	Request Not Routable (Proxy)
503	Session Context Not Found
504	Session Context Not Removable
505	Other Proxy Processing Error
506	Resources Unavailable
507	Request Initiated
508	Multiple Session Selection Unsupported

Preconditions

To use the CoA interface, a session must already exist on the switch. CoA can be used to identify a session and enforce a disconnect request. The update affects only the specified session.

CoA Request Response Code

The CoA Request response code can be used to convey a command to the switch. The supported commands are listed in Table 49-4 on page 49-102.

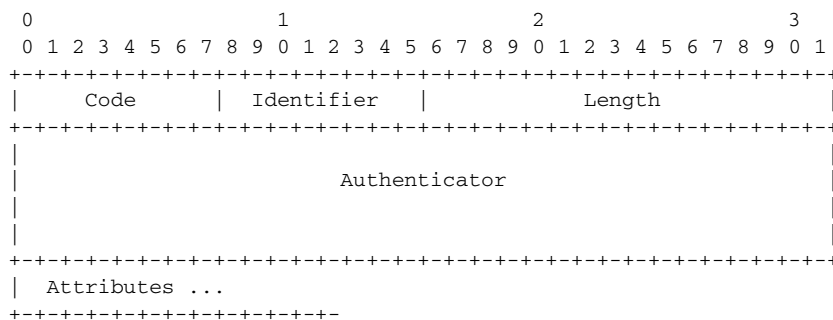
Session Identification

For disconnect and CoA requests targeted at a particular session, the switch locates the session based on one or more of the following attributes:

- Calling-Station-Id (IETF attribute #31 which contains the host MAC address)
- Audit-Session-Id (Cisco VSA)
- Acct-Session-Id (IETF attribute #44)

Unless all session identification attributes included in the CoA message match the session, the switch returns a Disconnect-NAK or CoA-NAK with the “Invalid Attribute Value” error-code attribute.

The packet format for a CoA Request code as defined in RFC 5176 consists of the fields: Code, Identifier, Length, Authenticator, and Attributes in Type:Length:Value (TLV) format.



The attributes field is used to carry Cisco VSAs.

CoA ACK Response Code

If the authorization state is changed successfully, a positive acknowledgement (ACK) is sent. The attributes returned within CoA ACK will vary based on the CoA Request and are discussed in individual CoA Commands.

CoA NAK Response Code

A negative acknowledgement (NAK) indicates a failure to change the authorization state and can include attributes that indicate the reason for the failure. Use **show** commands to verify a successful CoA.

CoA Request Commands

This section includes:

- [Session Reauthentication](#)
- [Session Termination](#)
- [CoA Disconnect-Request](#)
- [CoA Request: Disable Host Port](#)
- [CoA Request: Bounce-Port](#)

The switch supports the commands shown in [Table 49-4](#).

Table 49-4 CoA Commands Supported on the Switch

Command ¹	Cisco VSA
Reauthenticate host	Cisco:Avpair="subscriber:command=reauthenticate"
Terminate session	it is a standard disconnect request that does not require a VSA.
Bounce host port	Cisco:Avpair="subscriber:command=bounce-host-port"
Disable host port	Cisco:Avpair="subscriber:command=disable-host-port"

1. All CoA commands must include the session identifier between the switch and the CoA client.

Session Reauthentication

The AAA server typically generates a session reauthentication request when a host with an unknown identity or posture joins the network and is associated with a restricted access authorization profile (such as a guest VLAN). A reauthentication request allows the host to be placed in the appropriate authorization group when its credentials are known.

To initiate session authentication, the AAA server sends a standard CoA-Request message which contains a Cisco vendor-specific attribute (VSA) in this form:

Cisco:Avpair="subscriber:command=reauthenticate" and one or more session identification attributes.

The current session state determines the switch response to the message. If the session is currently authenticated by IEEE 802.1x, the switch responds by sending an EAPoL¹-RequestId message (see footnote 1 below) to the server.

If the session is currently authenticated by MAC authentication bypass (MAB), the switch sends an access-request to the server, passing the same identity attributes used for the initial successful authentication.

If session authentication is in progress when the switch receives the command, the switch terminates the process, and restarts the authentication sequence, starting with the method configured to be attempted first.

If the session is not yet authorized, or is authorized by using guest VLAN, or critical VLAN, or similar policies, the reauthentication message restarts the access control methods, beginning with the method configured to be attempted first. The current authorization of the session is maintained until the reauthentication leads to a different authorization result.

Session Termination

Three types of CoA requests can trigger session termination. A CoA Disconnect-Request terminates the session, without disabling the host port. This command causes re-initialization of the authenticator state machine for the specified host, but does not restrict that hosts' access to the network.

To restrict a hosts' access to the network, use a CoA Request with the *Cisco:Avpair="subscriber:command=disable-host-port"* VSA. This command is useful when a host is known to be causing problems on the network, and you need to immediately block network access for the host. When you want to restore network access on the port, reenale it using a non-RADIUS mechanism.

When a device with no supplicant, such as a printer, needs to acquire a new IP address (for example, after a VLAN change), terminate the session on the host port with port-bounce (temporarily disable and then re-enable the port).

1. Extensible Authentication Protocol over Lan

CoA Disconnect-Request

This command is a standard Disconnect-Request. Because this command is session-oriented, it must be accompanied by one or more of the session identification attributes described in the [“Session Identification” section on page 49-101](#). If the session cannot be located, the switch returns a Disconnect-NAK message with the “Session Context Not Found” error-code attribute. If the session *is* located, the switch terminates the session. After the session has been completely removed, the switch returns a Disconnect-ACK.

If the switch fails-over to a standby switch before returning a Disconnect-ACK to the client, the process is repeated on the new active switch when the request is re-sent from the client. If the session is not found following re-sending, a Disconnect-ACK is sent with the “Session Context Not Found” error-code attribute.

CoA Request: Disable Host Port

This command is carried in a standard CoA-Request message that has this new VSA:

```
Cisco:Avpair="subscriber:command=disable-host-port"
```

Because this command is session-oriented, it must be accompanied by one or more of the session identification attributes described in the [“Session Identification” section on page 49-101](#). If the session cannot be located, the switch returns a CoA-NAK message with the “Session Context Not Found” error-code attribute. If the session is located, the switch disables the hosting port and returns a CoA-ACK message.

If the switch fails before returning a CoA-ACK to the client, the process is repeated on the new active switch when the request is re-sent from the client. If the switch fails after returning a CoA-ACK message to the client but before the operation has completed, the operation is restarted on the new active switch.



Note

A Disconnect-Request failure following command re-sending could be the result of either a successful session termination before change-over (if the Disconnect-ACK was not sent) or a session termination by other means (for example, a link failure) that occurred after the original command was issued and before the standby switch became active.

CoA Request: Bounce-Port

This command is carried in a standard CoA-Request message that contains the following new VSA:

```
Cisco:Avpair="subscriber:command=bounce-host-port"
```

Because this command is session-oriented, it must be accompanied by one or more of the session identification attributes described in the [“Session Identification” section on page 49-101](#). If the session cannot be located, the switch returns a CoA-NAK message with the “Session Context Not Found” error-code attribute. If the session is located, the switch disables the hosting port for a period of 10 seconds, re-enables it (port-bounce), and returns a CoA-ACK.

If the switch fails before returning a CoA-ACK to the client, the process is repeated on the new active switch when the request is re-sent from the client. If the switch fails after returning a CoA-ACK message to the client but before the operation has completed, the operation is re-started on the new active switch.

Configuring RADIUS

This section describes how to configure your switch to support RADIUS. At a minimum, you must identify the host or hosts that run the RADIUS server software and define the method lists for RADIUS authentication. You can optionally define method lists for RADIUS authorization and accounting.

A method list defines the sequence and methods to be used to authenticate, to authorize, or to keep accounts on a user. You can use method lists to designate one or more security protocols to be used (such as TACACS+ or local username lookup), thus ensuring a backup system if the initial method fails. The software uses the first method listed to authenticate, to authorize, or to keep accounts on users. If that method does not respond, the software selects the next method in the list. This process continues until there is successful communication with a listed method or the method list is exhausted.

You should have access to and should configure a RADIUS server before configuring RADIUS features on your switch.

- [Default RADIUS Configuration, page 49-104](#)
- [Identifying the RADIUS Server Host, page 49-104](#) (required)
- [Configuring RADIUS Login Authentication, page 49-107](#) (required)
- [Defining AAA Server Groups, page 49-109](#) (optional)
- [Configuring RADIUS Authorization for User Privileged Access and Network Services, page 49-111](#) (optional)
- [Starting RADIUS Accounting, page 49-112](#) (optional)
- [Configuring Settings for All RADIUS Servers, page 49-113](#) (optional)
- [Configuring the Switch to Use Vendor-Specific RADIUS Attributes, page 49-113](#) (optional)
- [Configuring the Switch for Vendor-Proprietary RADIUS Server Communication, page 49-115](#) (optional)
- [Configuring CoA on the Switch, page 49-116](#)
- [Monitoring and Troubleshooting CoA Functionality, page 49-117](#)
- [Configuring RADIUS Server Load Balancing, page 49-117](#) (optional)

Default RADIUS Configuration

RADIUS and AAA are disabled by default.

To prevent a lapse in security, you cannot configure RADIUS through a network management application. When enabled, RADIUS can authenticate users accessing the switch using the CLI.

Identifying the RADIUS Server Host

Switch-to-RADIUS-server communication involves several components:

- Hostname or IP address
- Authentication destination port
- Accounting destination port
- Key string
- Timeout period
- Retransmission value

You identify RADIUS security servers by their hostname or IP address, hostname and specific UDP port numbers, or their IP address and specific UDP port numbers. The combination of the IP address and the UDP port number creates a unique identifier, allowing different ports to be individually defined as RADIUS hosts providing a specific AAA service. This unique identifier enables RADIUS requests to be sent to multiple UDP ports on a server at the same IP address.

If two different host entries on the same RADIUS server are configured for the same service—for example, accounting—the second host entry configured acts as a failover backup to the first one. Using this example, if the first host entry fails to provide accounting services, the `%RADIUS-4-RADIUS_DEAD` message appears, and then the switch tries the second host entry configured on the same device for accounting services. (The RADIUS host entries are tried in the order that they are configured.)

A RADIUS server and the switch use a shared secret text string to encrypt passwords and exchange responses. To configure RADIUS to use the AAA security commands, you must specify the host running the RADIUS server daemon and a secret text (key) string that it shares with the switch.

The timeout, retransmission, and encryption key values can be configured globally for all RADIUS servers, on a per-server basis, or in some combination of global and per-server settings. To apply these settings globally to all RADIUS servers communicating with the switch, use the three unique global configuration commands: **radius-server timeout**, **radius-server retransmit**, and **radius-server key**. To apply these values on a specific RADIUS server, use the **radius server host** global configuration command.

**Note**

- If you configure both global and per-server functions (timeout, retransmission, and key commands) on the switch, the per-server timer, retransmission, and key value commands override global timer, retransmission, and key value commands. For information on configuring these settings on all RADIUS servers, see the [“Configuring Settings for All RADIUS Servers” section on page 49-113](#).
- In Cisco IOS XE Release 3.8.7E, the legacy command **radius-server host** is deprecated. Use the **radius server host** command if the software running on your device is Cisco IOS XE Release 3.8.7E or later.

You can configure the switch to use AAA server groups to group existing server hosts for authentication. For more information, see the [“Defining AAA Server Groups” section on page 49-109](#).

To configure per-server RADIUS server communication, perform this task. This procedure is required.

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# radius server host {hostname ip-address} [auth-port port-number] [acct-port port-number] [timeout seconds] [retransmit retries] [key string]	<p>Specifies the IP address or hostname of the remote RADIUS server host.</p> <ul style="list-style-type: none"> • (Optional) For auth-port port-number, specify the UDP destination port for authentication requests. • (Optional) For acct-port port-number, specify the UDP destination port for accounting requests. • (Optional) For timeout seconds, specify the time interval that the switch waits for the RADIUS server to reply before resending. The range is 1 to 1000. This setting overrides the radius-server timeout global configuration command setting. If no timeout is set with the radius server host command, the setting of the radius-server timeout command is used. • (Optional) For retransmit retries, specify the number of times a RADIUS request is resent to a server if that server is not responding or responding slowly. The range is 1 to 1000. If no retransmit value is set with the radius server host command, the setting of the radius-server retransmit global configuration command is used. • (Optional) For key string, specify the authentication and encryption key used between the switch and the RADIUS daemon running on the RADIUS server. <p>Note The key is a text string that must match the encryption key used on the RADIUS server. Always configure the key as the last item in the radius server host command. Leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in your key, do not enclose the key in quotation marks unless the quotation marks are part of the key.</p> <p>To configure the switch to recognize more than one host entry associated with a single IP address, enter this command as many times as necessary, making sure that each UDP port number is different. The switch software searches for hosts in the order in which you specify them. Set the timeout, retransmit, and encryption key values to use with the specific RADIUS host.</p>
Step 3	Switch(config)# end	Returns to privileged EXEC mode.
Step 4	Switch# show running-config	Verifies your entries.
Step 5	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To remove the specified RADIUS server, use the **no radius server host** hostname | ip-address global configuration command.

This example shows how to configure one RADIUS server to be used for authentication and another to be used for accounting:

```
Switch(config)# radius server host 172.29.36.49 auth-port 1612 key rad1
Switch(config)# radius server host 172.20.36.50 acct-port 1618 key rad2
```


This example shows how to configure *host1* as the RADIUS server and to use the default ports for both authentication and accounting:

```
Switch(config)# radius server host host1
```

**Note**

You also need to configure some settings on the RADIUS server. These settings include the IP address of the switch and the key string to be shared by both the server and the switch. For more information, see the RADIUS server documentation.

Configuring RADIUS Login Authentication

To configure AAA authentication, you define a named list of authentication methods and then apply that list to various ports. The method list defines the types of authentication to be performed and the sequence in which they are performed; it must be applied to a specific port before any of the defined authentication methods are performed. The only exception is the default method list (which, by coincidence, is named *default*). The default method list is automatically applied to all ports except those that have a named method list explicitly defined.

A method list describes the sequence and authentication methods to be queried to authenticate a user. You can designate one or more security protocols to be used for authentication, thus ensuring a backup system for authentication in case the initial method fails. The software uses the first method listed to authenticate users; if that method fails to respond, the software selects the next authentication method in the method list. This process continues until there is successful communication with a listed authentication method or until all defined methods are exhausted. If authentication fails at any point in this cycle—meaning that the security server or local username database responds by denying the user access—the authentication process stops, and no other authentication methods are attempted.

To configure login authentication, perform this task. This procedure is required.

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# aaa new-model	Enables AAA.

	Command	Purpose
Step 3	Switch(config)# aaa authentication login { default <i>list-name</i> } <i>method1</i> [<i>method2...</i>]	<p>Creates a login authentication method list.</p> <ul style="list-style-type: none"> To create a default list that is used when a named list is <i>not</i> specified in the login authentication command, use the default keyword followed by the methods that are to be used in default situations. The default method list is automatically applied to all ports. For <i>list-name</i>, specify a character string to name the list you are creating. For <i>method1...</i>, specify the actual method the authentication algorithm tries. The additional methods of authentication are used only if the previous method returns an error, not if it fails. <p>Select one of these methods:</p> <ul style="list-style-type: none"> enable—Use the enable password for authentication. Before you can use this authentication method, you must define an enable password by using the enable password global configuration command. group radius—Use RADIUS authentication. Before you can use this authentication method, you must configure the RADIUS server. For more information, see the “Identifying the RADIUS Server Host” section on page 49-104. line—Use the line password for authentication. Before you can use this authentication method, you must define a line password. Use the password password line configuration command. local—Use the local username database for authentication. You must enter username information in the database. Use the username name password global configuration command. local-case—Use a case-sensitive local username database for authentication. You must enter username information in the database by using the username password global configuration command. none—Do not use any authentication for login.
Step 4	Switch(config)# line [console tty vty] <i>line-number</i> [<i>ending-line-number</i>]	Enters line configuration mode, and configure the lines to which you want to apply the authentication list.
Step 5	Switch(config)# login authentication { default <i>list-name</i> }	<p>Applies the authentication list to a line or set of lines.</p> <ul style="list-style-type: none"> If you specify default, use the default list created with the aaa authentication login command. For <i>list-name</i>, specify the list created with the aaa authentication login command.
Step 6	Switch(config)# end	Returns to privileged EXEC mode.
Step 7	Switch# show running-config	Verifies your entries.
Step 8	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To disable AAA, use the **no aaa new-model** global configuration command. To disable AAA authentication, use the **no aaa authentication login {default | list-name} method1 [method2...]** global configuration command. To either disable RADIUS authentication for logins or to return to the default value, use the **no login authentication {default | list-name}** line configuration command.

**Note**

To secure the switch for HTTP access by using AAA methods, you must configure the switch with the **ip http authentication aaa** global configuration command. Configuring AAA authentication does not secure the switch for HTTP access by using AAA methods.

For more information about the **ip http authentication** command, see the *Cisco IOS Security Command Reference, Release 12.2* from the Cisco.com page under **Documentation > Cisco IOS Software > 12.2 Mainline > Command References**.

Defining AAA Server Groups

You can configure the switch to use AAA server groups to group existing server hosts for authentication. You select a subset of the configured server hosts and use them for a particular service. The server group is used with a global server-host list, which lists the IP addresses of the selected server hosts.

Server groups also can include multiple host entries for the same server if each entry has a unique identifier (the combination of the IP address and UDP port number), allowing different ports to be individually defined as RADIUS hosts providing a specific AAA service. If you configure two different host entries on the same RADIUS server for the same service, (for example, accounting), the second configured host entry acts as a failover backup to the first one.

You use the **server** group server configuration command to associate a particular server with a defined group server. You can either identify the server by its IP address or identify multiple host instances or entries by using the optional **auth-port** and **acct-port** keywords.

To define the AAA server group and associate a particular RADIUS server with it, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# radius server host {hostname ip-address} [auth-port port-number] [acct-port port-number] [timeout seconds] [retransmit retries] [key string]	<p>Specifies the IP address or hostname of the remote RADIUS server host.</p> <ul style="list-style-type: none"> • (Optional) For auth-port port-number, specify the UDP destination port for authentication requests. • (Optional) For acct-port port-number, specify the UDP destination port for accounting requests. • (Optional) For timeout seconds, specify the time interval that the switch waits for the RADIUS server to reply before resending. The range is 1 to 1000. This setting overrides the radius-server timeout global configuration command setting. If no timeout is set with the radius server host command, the setting of the radius-server timeout command is used. • (Optional) For retransmit retries, specify the number of times a RADIUS request is resent to a server if that server is not responding or responding slowly. The range is 1 to 1000. If no retransmit value is set with the radius server host command, the setting of the radius-server retransmit global configuration command is used. • (Optional) For key string, specify the authentication and encryption key used between the switch and the RADIUS daemon running on the RADIUS server. <p>Note The key is a text string that must match the encryption key used on the RADIUS server. Always configure the key as the last item in the radius server host command. Leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in your key, do not enclose the key in quotation marks unless the quotation marks are part of the key.</p> <p>To configure the switch to recognize more than one host entry associated with a single IP address, enter this command as many times as necessary, making sure that each UDP port number is different. The switch software searches for hosts in the order in which you specify them. Set the timeout, retransmit, and encryption key values to use with the specific RADIUS host.</p>
Step 3	Switch(config)# aaa new-model	Enables AAA.
Step 4	Switch(config)# aaa group server radius group-name	<p>Defines the AAA server-group with a group name.</p> <p>This command puts the switch in a server group configuration mode.</p>
Step 5	Switch(config)# server ip-address	<p>Associates a particular RADIUS server with the defined server group. Repeat this step for each RADIUS server in the AAA server group.</p> <p>Each server in the group must be previously defined in Step 2.</p>
Step 6	Switch(config)# end	Returns to privileged EXEC mode.
Step 7	Switch# show running-config	Verifies your entries.

	Command	Purpose
Step 8	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.
Step 9		Enables RADIUS login authentication. See the “Configuring RADIUS Login Authentication” section on page 49-107.

To remove the specified RADIUS server, use the **no radius server host** *hostname | ip-address* global configuration command. To remove a server group from the configuration list, use the **no aaa group server radius** *group-name* global configuration command. To remove the IP address of a RADIUS server, use the **no server ip-address** server group configuration command.

In this example, the switch is configured to recognize two different RADIUS group servers (*group1* and *group2*). Group1 has two different host entries on the same RADIUS server configured for the same services. The second host entry acts as a failover backup to the first entry.

```
Switch(config)# radius server host 172.20.0.1 auth-port 1000 acct-port 1001
Switch(config)# radius server host 172.10.0.1 auth-port 1645 acct-port 1646
Switch(config)# aaa new-model
Switch(config)# aaa group server radius group1
Switch(config-sg-radius)# server 172.20.0.1 auth-port 1000 acct-port 1001
Switch(config-sg-radius)# exit
Switch(config)# aaa group server radius group2
Switch(config-sg-radius)# server 172.20.0.1 auth-port 2000 acct-port 2001
Switch(config-sg-radius)# exit
```

Configuring RADIUS Authorization for User Privileged Access and Network Services

AAA authorization limits the services available to a user. When AAA authorization is enabled, the switch uses information retrieved from the user's profile, which is in the local user database or on the security server, to configure the user's session. The user is granted access to a requested service only if the information in the user profile allows it.

You can use the **aaa authorization** global configuration command with the **radius** keyword to set parameters that restrict a user's network access to privileged EXEC mode.

The **aaa authorization exec radius local** command sets these authorization parameters:

- Use RADIUS for privileged EXEC access authorization if authentication was performed by using RADIUS.
- Use the local database if authentication was not performed by using RADIUS.



Note

Authorization is bypassed for authenticated users who log in using the CLI even if authorization has been configured.

We recommend that you use the **aaa authorization network default group radius local** command to configure RADIUS authentications. Custom authorization method of RADIUS authentication is not supported.

To specify RADIUS authorization for privileged EXEC access and network services, perform this task:

■ Controlling Switch Access with RADIUS

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# aaa authorization network radius	Configures the switch for user RADIUS authorization for all network-related service requests.
Step 3	Switch(config)# aaa authorization exec radius	Configures the switch for user RADIUS authorization if the user has privileged EXEC access. The exec keyword might return user profile information (such as autocommand information).
Step 4	Switch(config)# end	Returns to privileged EXEC mode.
Step 5	Switch# show running-config	Verifies your entries.
Step 6	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To disable authorization, use the **no aaa authorization {network | exec} method1** global configuration command.

Starting RADIUS Accounting

The AAA accounting feature tracks the services that users are accessing and the amount of network resources that they are consuming. When AAA accounting is enabled, the switch reports user activity to the RADIUS security server in the form of accounting records. Each accounting record contains accounting attribute-value (AV) pairs and is stored on the security server. This data can then be analyzed for network management, client billing, or auditing.

To enable RADIUS accounting for each Cisco IOS privilege level and for network services, perform these steps:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# aaa accounting network start-stop radius	Enables RADIUS accounting for all network-related service requests.
Step 3	Switch(config)# aaa accounting exec start-stop radius	Enables RADIUS accounting to send a start-record accounting notice at the beginning of a privileged EXEC process and a stop-record at the end.
Step 4	Switch(config)# end	Returns to privileged EXEC mode.
Step 5	Switch# show running-config	Verifies your entries.
Step 6	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To disable accounting, use the **no aaa accounting {network | exec} {start-stop} method1...** global configuration command.

Configuring Settings for All RADIUS Servers

To configure global communication settings between the switch and all RADIUS servers, perform these steps:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# radius-server key <i>string</i>	Specifies the shared secret text string used between the switch and all RADIUS servers. Note The key is a text string that must match the encryption key used on the RADIUS server. Leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in your key, do not enclose the key in quotation marks unless the quotation marks are part of the key.
Step 3	Switch(config)# radius-server retransmit <i>retries</i>	Specifies the number of times the switch sends each RADIUS request to the server before giving up. The default is 3; the range 1 to 1000.
Step 4	Switch(config)# radius-server timeout <i>seconds</i>	Specifies the number of seconds a switch waits for a reply to a RADIUS request before resending the request. The default is 5 seconds; the range is 1 to 1000.
Step 5	Switch(config)# radius-server deadtime <i>minutes</i>	Specifies the number of minutes a RADIUS server, which is not responding to authentication requests, to be skipped, thus avoiding the wait for the request to timeout before trying the next configured server. The default is 0; the range is 1 to 1440 minutes.
Step 6	Switch(config)# end	Returns to privileged EXEC mode.
Step 7	Switch# show running-config	Verifies your settings.
Step 8	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To return to the default setting for the retransmit, timeout, and deadtime, use the **no** forms of these commands.

Configuring the Switch to Use Vendor-Specific RADIUS Attributes

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating vendor-specific information between the switch and the RADIUS server by using the vendor-specific attribute (attribute 26). Vendor-specific attributes (VSAs) allow vendors to support their own extended attributes not suitable for general use. The Cisco RADIUS implementation supports one vendor-specific option by using the format recommended in the specification. Cisco's vendor-ID is 9, and the supported option has vendor-type 1, which is named *cisco-avpair*. The value is a string with this format:

```
protocol : attribute sep value *
```

Protocol is a value of the Cisco protocol attribute for a particular type of authorization. *Attribute* and *value* are an appropriate attribute-value (AV) pair defined in the Cisco TACACS+ specification, and *sep* is = for mandatory attributes and is * for optional attributes. The full set of features available for TACACS+ authorization can then be used for RADIUS.

For example, this AV pair activates Cisco’s *multiple named ip address pools* feature during IP authorization (during PPP IPCP address assignment):

```
cisco-avpair= "ip:addr-pool=first"
```

This example shows how to provide a user logging in from a switch with immediate access to privileged EXEC commands:

```
cisco-avpair= "shell:priv-lvl=15"
```

This example shows how to specify an authorized VLAN in the RADIUS server database:

```
cisco-avpair= "tunnel-type(#64)=VLAN(13)"
cisco-avpair= "tunnel-medium-type(#65)=802 media(6)"
cisco-avpair= "tunnel-private-group-ID(#81)=vlanid"
```

This example shows how to apply an input ACL in ASCII format to an interface for the duration of this connection:

```
cisco-avpair= "ip:inacl#1=deny ip 10.10.10.10 0.0.255.255 20.20.20.20 255.255.0.0"
cisco-avpair= "ip:inacl#2=deny ip 10.10.10.10 0.0.255.255 any"
cisco-avpair= "mac:inacl#3=deny any any decnet-iv"
```

This example shows how to apply an output ACL in ASCII format to an interface for the duration of this connection:

```
cisco-avpair= "ip:outacl#2=deny ip 10.10.10.10 0.0.255.255 any"
```

Other vendors have their own unique vendor-IDs, options, and associated VSAs. For more information about vendor-IDs and VSAs, see RFC 2138, “Remote Authentication Dial-In User Service (RADIUS).”

To configure the switch to recognize and use VSAs, perform these steps:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# radius-server vsa send [accounting authentication]	Enables the switch to recognize and use VSAs as defined by RADIUS IETF attribute 26. <ul style="list-style-type: none"> • (Optional) Use the accounting keyword to limit the set of recognized vendor-specific attributes to only accounting attributes. • (Optional) Use the authentication keyword to limit the set of recognized vendor-specific attributes to only authentication attributes. If you enter this command without keywords, both accounting and authentication vendor-specific attributes are used.
Step 3	Switch(config)# end	Returns to privileged EXEC mode.
Step 4	Switch# show running-config	Verifies your settings.
Step 5	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.



Note

For a complete list of RADIUS attributes or more information about vendor-specific attribute 26, see the “RADIUS Attributes” appendix in the *Cisco IOS Security Configuration Guide, Release 12.2* from the Cisco.com page under **Documentation > Cisco IOS Software > 12.2 Mainline > Command References**.

Configuring the Switch for Vendor-Proprietary RADIUS Server Communication

Although an IETF draft standard for RADIUS specifies a method for communicating vendor-proprietary information between the switch and the RADIUS server, some vendors have extended the RADIUS attribute set in a unique way. Cisco IOS software supports a subset of vendor-proprietary RADIUS attributes.

As mentioned earlier, to configure RADIUS (whether vendor-proprietary or IETF draft-compliant), you must specify the host running the RADIUS server daemon and the secret text string it shares with the switch. You specify the RADIUS host and secret text string by using the **radius-server** global configuration commands.

To specify a vendor-proprietary RADIUS server host and a shared secret text string, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# radius server host {hostname ip-address} non-standard	Specifies the IP address or hostname of the remote RADIUS server host and identify that it is using a vendor-proprietary implementation of RADIUS.
Step 3	Switch(config)# radius-server key string	Specifies the shared secret text string used between the switch and the vendor-proprietary RADIUS server. The switch and the RADIUS server use this text string to encrypt passwords and exchange responses. Note The key is a text string that must match the encryption key used on the RADIUS server. Leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in your key, do not enclose the key in quotation marks unless the quotation marks are part of the key.
Step 4	Switch(config)# end	Returns to privileged EXEC mode.
Step 5	Switch# show running-config	Verifies your settings.
Step 6	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To delete the vendor-proprietary RADIUS host, use the **no radius server host {hostname | ip-address} non-standard** global configuration command. To disable the key, use the **no radius-server key** global configuration command.

This example shows how to specify a vendor-proprietary RADIUS host and to use a secret key of *rad124* between the switch and the server:

```
Switch(config)# radius server host 172.20.30.15 nonstandard
Switch(config)# radius-server key rad124
```

Configuring CoA on the Switch

To configure CoA on a switch, perform these steps. This procedure is required.

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# aaa new-model	Enables AAA.
Step 3	Switch(config)# aaa server radius dynamic-author	Configures the switch as an authentication, authorization, and accounting (AAA) server to facilitate interaction with an external policy server.
Step 4	Switch(config-locsvr-da-radius)# client { <i>ip-address</i> <i>name</i> } [vrf <i>vrfname</i>] [server-key <i>string</i>]	Enters dynamic authorization local server configuration mode and specify a RADIUS client from which a device will accept CoA and disconnect requests.
Step 5	Switch(config-locsvr-da-radius)# server-key [0 7] <i>string</i>	Configures the RADIUS key to be shared between a device and RADIUS clients.
Step 6	Switch(config-locsvr-da-radius)# port <i>port-number</i>	Specifies the port on which a device listens for RADIUS requests from configured RADIUS clients.
Step 7	Switch(config-locsvr-da-radius)# auth-type { <i>any</i> <i>all</i> <i>session-key</i> }	Specifies the type of authorization the switch uses for RADIUS clients. The client must match all the configured attributes for authorization.
Step 8	Switch(config-locsvr-da-radius)# ignore session-key	(Optional) Configures the switch to ignore the session-key. For more information about the ignore command, see the Cisco IOS Intelligent Services Gateway Command Reference on Cisco.com.
Step 9	Switch(config-locsvr-da-radius)# ignore server-key	(Optional) Configures the switch to ignore the server-key. For more information about the ignore command, see the Cisco IOS Intelligent Services Gateway Command Reference on Cisco.com.
Step 10	Switch(config-locsvr-da-radius)# exit	Switches to global configuration mode.
Step 11	Switch(config)# authentication command bounce-port ignore	(Optional) Configures the switch to ignore a CoA request to temporarily disable the port hosting a session. The purpose of temporarily disabling the port is to trigger a DHCP renegotiation from the host when a VLAN change occurs and there is no supplicant on the endpoint to detect the change.
Step 12	Switch(config)# authentication command disable-port ignore	(Optional) Configures the switch to ignore a nonstandard command requesting that the port hosting a session be administratively shut down. Shutting down the port results in termination of the session. Use standard CLI or SNMP commands to re-enable the port.
Step 13	Switch# end	Returns to privileged EXEC mode.
Step 14	Switch# show running-config	Verifies your entries.
Step 15	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To disable AAA, use the **no aaa new-model** global configuration command. To disable the AAA server functionality on the switch, use the **no aaa server radius dynamic authorization** global configuration command:

```
Switch(config)# aaa server radius dynamic-author
Switch(config-locsvr-da-radius)# client ip addr vrf vrfname
Switch(config-locsvr-da-radius)# server-key cisco123
Switch(config-locsvr-da-radius)# port 3799
```

**Note**

Default port for packet of disconnect is 1700. Port 3799 is required to interoperate with ACS 5.1.

```
Switch(config)# authentication command bounce-port ignore
```

Monitoring and Troubleshooting CoA Functionality

The following Cisco IOS commands can be used to monitor and troubleshoot CoA functionality on the switch:

- **debug radius**
- **debug aaa coa**
- **debug aaa pod**
- **debug aaa subsys**
- **debug cmdhd [detail | error | events]**
- **show aaa attributes protocol radius**

Configuring RADIUS Server Load Balancing

This feature allows access and authentication requests to be evenly across all RADIUS servers in a server group. For more information, see the *RADIUS Server Load Balancing* chapter of the *Cisco IOS Security Configuration Guide*, Release 12.2:

http://www.ciscosystems.com/en/US/docs/ios/12_2sb/feature/guide/sbrldbl.html

Displaying the RADIUS Configuration

To display the RADIUS configuration, use the **show running-config** privileged EXEC command.

Configuring Device Sensor

This section includes the following:

- [About Device Sensor, page 49-118](#)
- [MSP-IOS Sensor Device Classifier Interaction, page 49-119](#)
- [Configuring Device Sensor, page 49-119](#)
- [Configuration Examples for the Device Sensor Feature, page 49-125](#)

About Device Sensor

Device Sensor uses protocols such as Cisco Discovery Protocol (CDP), Link Layer Discovery Protocol (LLDP), and DHCP to obtain endpoint information from network devices and make this information available to its clients. Device Sensor has internal clients, such as the embedded Device Classifier (local analyzer), Auto Smartports (ASP), MediaNet Service Interface Media Services Proxy, and EnergyWise. Device Sensor also has an external client, Identity Services Engine (ISE), which uses RADIUS accounting to receive and analyze endpoint data. When integrated with ISE, Device Sensor provides central policy management and device-profiling capabilities.


Note

Cisco Identity Services Engine (ISE) based profiling is not supported on the LAN Base image.

Device profiling capability consists of two parts:

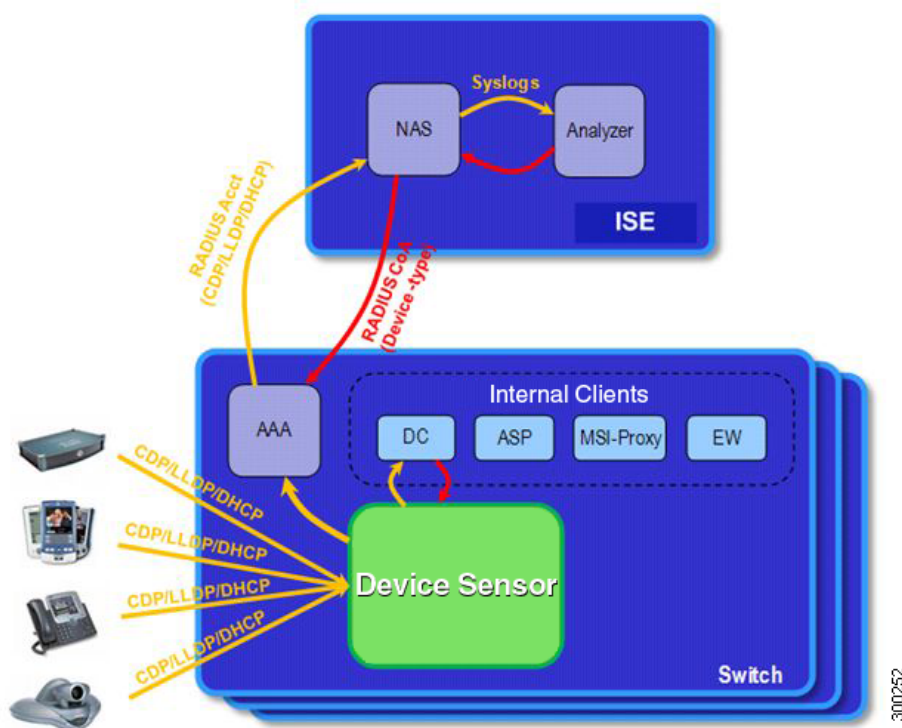
- Collector--Gathers endpoint data from network devices.
- Analyzer--Processes the data and determines the type of device.

For more information on device profiling, see the “Configuring Endpoint Profiling Policies” chapter in the *Cisco Identity Services Engine User Guide* at this URL:

http://www.cisco.com/en/US/docs/security/ise/1.1/user_guide/ise_prof_pol.html

Device Sensor represents the embedded collector functionality. Figure 22 shows a Device Sensor in the context of its internal clients and the ISE.

Figure 22 **Device Sensor and Clients**



Client notifications and accounting messages that contain profiling data and other session-related data are generated and sent to the internal clients and the ISE. By default, client notifications and accounting events are generated only when an incoming packet includes a Type-Length-Value (TLV) that has not previously been received within a given access session. You can enable client notifications and accounting events for TLV changes; that is, when a previously received TLV is received with a different value.

Device Sensor port security protects a switch from consuming memory and crashing during deliberate or unintentional denial-of-service (DoS)-type attacks. Device Sensor limits the maximum number of device monitoring sessions to 32 per port. While hosts are inactive, the age session limit is 12 hours.

MSP-IOS Sensor Device Classifier Interaction



Note

To enable MSP, you must configure the **profile flow** command. Once done, when SIP, H323, or mDNS traffic are present, appropriate (SIP, H323, or mDNS) TLV notifications are sent to the IOS sensor.

MSP (Media Service Proxy) offers bandwidth reservation for audio or video flows and Metadata services to 3rd-party endpoints. To offer and install Media services, MSP must identify flow attributes and device details. MSP device identification requires automatic identification of various media end points in the network, thereby avoiding any change to the installed end point base. To offer MSP device discovery services, MSP leverages current IOS sensor capability for device classification. (Starting with Release IOS XE 3.3.0SG and IOS 15.1(1)SG, IOS sensor can be used to perform device identification. MSP uses the same functionality with the addition of SIP, H323, and Multicast DNS (mDNS) protocols.) Starting with Release IOS XE 3.4.0SG and IOS 15.1(2)SG, MSP offers Media services to two kinds of media endpoints: IP Surveillance Cameras and Video-Conferencing Endpoints. Surveillance cameras are identified using mDNS protocol whereas Video-conference-Endpoints are identified using SIP and H.323 protocols.

mDNS compatible devices (Axis, Pelco cameras etc) send mDNS messages for DNS service discovery to a multicast IP address (224.0.0.251) on a standard mDNS port 5353. The mDNS client module listens to this UDP port, receives the mDNS message, and sends it in TLV format to the mDNS IOS sensor shim for further device classification. The module parses the mDNS query and Answer messages fields to create these TLVs.

A Session Initiation Protocol (SIP) registration message is used for SIP based device-discovery and is sent to Cisco Call manager by the SIP Client. A H.225 RAS client registration message is used for H323-based device discovery.

If no Cisco Unified Communicator Manager or GateKeeper exists in the topology, the Endpoint will not generate device Register messages. To handle device discovery in these scenarios, MSP expects the endpoint to make a SIP or H323 call so that MSP snoops the SIP invite or the H323 setup message to identify endpoint details and notify the IOS sensor.

After the IOS sensor receives these protocol details from MSP, the IOS sensor prepares Normalized TLVs, with the new protocols. These protocol details are sent to session manager for further classification.

Configuring Device Sensor

Device Sensor is enabled by default. Complete the following tasks when you want Device Sensor to include or exclude a list of TLVs (termed filter lists) for a particular protocol.



Note

If you do not perform any Device Sensor configuration tasks, the following TLVs are included by default:

- CDP filter--secondport-status-type and powernet-event-type (types 28 and 29)
- LLDP filter--organizationally-specific (type 127)
- DHCP filter--message-type (type 53)

- [Enabling MSP, page 49-120](#)
- [Enabling Accounting Augmentation, page 49-120](#)
- [Creating a Cisco Discovery Protocol Filter, page 49-121](#)
- [Creating an LLDP Filter, page 49-121](#)
- [Creating a DHCP Filter, page 49-122](#)
- [Applying a Protocol Filter to the Device Sensor Output, page 49-122](#)
- [Tracking TLV Changes, page 49-123](#)
- [Verifying the Device Sensor Configuration, page 49-124](#)
- [Troubleshooting Commands, page 49-125](#)
- [Restrictions for Device Sensor, page 49-125](#)

Enabling MSP

You must configure the MSP **profile flow** command to activate the MSP platform Packet parser. This is because the MSP device handler is tightly coupled with MSP flow parser. Not enabling this command means that MSP will not send SIP, H323 notifications to the IOS sensor.

To enable MSP, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	configure terminal Switch# configure terminal	Enters global configuration mode.
Step 2	profile flow Switch(config)# profile flow	Enables MSP.
Step 3	end Switch(config)# end	Returns to privileged EXEC mode.

Use the **no** form of the profile flow command to disable MSP.

Enabling Accounting Augmentation

For the Device Sensor protocol data to be added to accounting messages, you must first enable session accounting by using the following standard Authentication, Authorization, and Accounting (AAA) and RADIUS configuration commands:

```
Switch(config)# aaa new-model
Switch(config)# aaa accounting dot1x default start-stop group radius
```

```
Switch(config)# radius server host{hostname|ip-address}[auth-port
port-number][acct-port port-number] [timeout seconds][retransmit retries][key string]
Switch(config)# radius-server vsa send accounting
```

To add Device Sensor protocol data to accounting records, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	configure terminal Switch# configure terminal	Enters global configuration mode.
Step 2	device-sensor accounting Switch(config)# device-sensor accounting	Enables the addition of sensor protocol data to accounting records and also enables the generation of additional accounting events when new sensor data is detected.
Step 3	end Switch(config)# end	Returns to privileged EXEC mode.

Creating a Cisco Discovery Protocol Filter

To create a CDP filter containing a list of TLVs that can be included or excluded in the Device Sensor output, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	configure terminal Switch# configure terminal	Enters global configuration mode.
Step 2	device-sensor filter-list cdp list tlv-list-name Switch(config)# device-sensor filter-list cdp list cdp-list	Creates a TLV list and enters CDP sensor configuration mode, where you can configure individual TLVs.
Step 3	tlv {name tlv-name number tlv-number} Switch(config-sensor-cdplist)# tlv number 10	Adds individual CDP TLVs to the TLV list. You can delete the TLV list without individually removing TLVs from the list by using the no device-sensor filter-list cdp list tlv-list-name command.
Step 4	end Switch(config-sensor-cdplist)# end	Returns to privileged EXEC mode.

Creating an LLDP Filter

To create an LLDP filter containing a list of TLVs that can be included or excluded in the Device Sensor output, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	configure terminal Switch# configure terminal	Enters global configuration mode.
Step 2	device-sensor filter-list lldp list tlv-list-name Switch(config)# device-sensor filter-list lldp list lldp-list	Creates a TLV list and enters LLDP sensor configuration mode, where you can configure individual TLVs.
Step 3	tlv {name tlv-name number tlv-number} Switch(config-sensor-cdplist)# tlv number 10	Adds individual LLDP TLVs to the TLV list. You can delete the TLV list without individually removing TLVs from the list by using the no device-sensor filter-list lldp list tlv-list-name command.
Step 4	end Switch(config-sensor-llldplist)# end	Returns to privileged EXEC mode.

Creating a DHCP Filter

To create a DHCP filter containing a list of DHCP options that can be included or excluded in the Device Sensor output, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	configure terminal Switch# configure terminal	Enters global configuration mode.
Step 2	device-sensor filter-list dhcp list option-list-name Switch(config)# device-sensor filter-list dhcp list dhcp-list	Creates an options list and enters DHCP sensor configuration mode, where you can specify individual DHCP options.
Step 3	option {name option-name number option-number} Switch(config-sensor-dhcplist)# option number 50	Adds individual DHCP options to the option list. You can delete the entire option list without removing options individually from the list by using the no device-sensor filter-list dhcp list option-list-name command.
Step 4	end Switch(config)# end	Returns to privileged EXEC mode.

Applying a Protocol Filter to the Device Sensor Output

Beginning in privileged EXEC mode, follow these steps to apply a CDP, LLDP, or DHCP filter to the sensor output. The output is session notifications to internal sensor clients and accounting requests to the RADIUS server.



Note

Only one filter list can be included or excluded at a time.

	Command	Purpose
Step 1	configure terminal Switch# configure terminal	Enters global configuration mode.
Step 2	device-sensor filter-spec {cdp dhcp lldp} {exclude {all list list-name} include list list-name} Switch(config)# device-sensor filter-spec cdp include list list1	Applies a specific protocol filter containing a list of protocol TLV fields or DHCP options to the Device Sensor output. <ul style="list-style-type: none"> • cdp--Applies a CDP TLV filter list to the device sensor output. • lldp--Applies an LLDP TLV filter list to the device sensor output. • dhcp--Applies a DHCP option filter list to the device sensor output. • exclude--Specifies the TLVs that must be excluded from the device sensor output. • include--Specifies the TLVs that must be included from the device sensor output. • all--Disables all notifications for the associated protocol. • list list-name--Specifies the protocol TLV filter list name.
Step 3	end Switch(config)# end	Returns to privileged EXEC mode.

Tracking TLV Changes

By default, client notifications and accounting events are generated only when an incoming packet includes a TLV that has not previously been received within a given session.

To enable client notifications and accounting events for TLV changes, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	configure terminal Switch# configure terminal	Enters global configuration mode.
Step 2	device-sensor notify all-changes Switch(config)# device-sensor notify all-changes	Enables client notifications and accounting events for all TLV changes, that is, where either a new TLV is received or a previously received TLV is received with a new value in the context of a given session. Note Use the default device-sensor notify or the device-sensor notify new-tlvs command to return to the default TLV.
Step 3	end Switch(config)# end	Returns to privileged EXEC mode.

Verifying the Device Sensor Configuration

To verify the sensor cache entries for all devices, follow these steps, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	show device-sensor cache mac <i>mac-address</i>	Displays sensor cache entries (the list of protocol TLVs or options received from a device) for a specific device. <ul style="list-style-type: none"> mac-address is the MAC address of the endpoint
Step 2	show device-sensor cache all Switch(config)# device-sensor notify all-changes	Displays sensor cache entries for all devices.

This is an example of the **show device-sensor cache mac mac-address** privileged EXEC command output:

```
Switch# show device-sensor cache mac 0024.14dc.df4d

Device: 0024.14dc.df4d on port GigabitEthernet1/0/24
-----
Proto Type:Name                               Len Value
cdp    26:power-available-type                 16 00 1A 00 10 00 00 00 01 00 00 00 00 FF FF FF FF
cdp    22:mgmt-address-type                     17 00 16 00 11 00 00 00 01 01 01 CC 00 04 09 1B 65
                                0E
cdp    11:duplex-type                           5 00 0B 00 05 01
cdp    9:vtp-mgmt-domain-type                   4 00 09 00 04
cdp    4:capabilities-type                     8 00 04 00 08 00 00 00 28
cdp    1:device-name                           14 00 01 00 0E 73 75 70 70 6C 69 63 61 6E 74
lldp   0:end-of-lldpdu                         2 00 00
lldp   8:management-address                   14 10 0C 05 01 09 1B 65 0E 03 00 00 00 01 00
lldp   7:system-capabilities                   6 0E 04 00 14 00 04
lldp   4:port-description                     23 08 15 47 69 67 61 62 69 74 45 74 68 65 72 6E 65
                                74 31 2F 30 2F 32 34
lldp   5:system-name                           12 0A 0A 73 75 70 70 6C 69 63 61 6E 74
dhcp   82:relay-agent-info                     20 52 12 01 06 00 04 00 18 01 18 02 08 00 06 00 24
                                14 DC DF 80
dhcp   12:host-name                           12 0C 0A 73 75 70 70 6C 69 63 61 6E 74
dhcp   61:client-identifier                    32 3D 1E 00 63 69 73 63 6F 2D 30 30 32 34 2E 31 34
                                64 63 2E 64 66 34 64 2D 47 69 31 2F 30 2F 32 34
dhcp   57:max-message-size                     4 39 02 04 80
```

This is an example of the **show device-sensor cache all** privileged EXEC command output:

```
Switch# show device-sensor cache all

Device: 001c.0f74.8480 on port GigabitEthernet2/1
-----
Proto Type:Name                               Len Value
dhcp   52:option-overload                      3 34 01 03
dhcp   60:class-identifier                     11 3C 09 64 6F 63 73 69 73 31 2E 30
dhcp   55:parameter-request-list               8 37 06 01 42 06 03 43 96
dhcp   61:client-identifier                    27 3D 19 00 63 69 73 63 6F 2D 30 30 31 63 2E 30 66
                                37 34 2E 38 34 38 30 2D 56 6C 31
dhcp   57:max-message-size                      4 39 02 04 80
Device: 000f.f7a7.234f on port GigabitEthernet2/1
-----
Proto Type:Name                               Len Value
cdp    22:mgmt-address-type                     8 00 16 00 08 00 00 00 00
```

```
cdp 19:cos-type 5 00 13 00 05 00
cdp 18:trust-type 5 00 12 00 05 00
cdp 11:duplex-type 5 00 0B 00 05 01
cdp 10:native-vlan-type 6 00 0A 00 06 00 01
cdp 9:vtp-mgmt-domain-type 9 00 09 00 09 63 69 73 63 6F
```

Troubleshooting Commands

The following commands can help troubleshoot Device Sensor.

- **debug device-sensor {errors | events}**
- **debug authentication all**

Restrictions for Device Sensor

- Only CDP, LLDP, and DHCP protocols are supported.
- The session limit for profiling ports is 32.
- The length of one TLV must not be more than 1024 and the total length of TLVs (combined length of TLVs) of all protocols must not be more than 4096.
- Device Sensor profiles devices that are only one hop away.

Configuration Examples for the Device Sensor Feature

The following example shows how to create a CDP filter containing a list of TLVs:

```
Switch> enable
Switch# configure terminal
Switch(config)# device-sensor filter-list cdp list cdp-list
Switch(config-sensor-cdplist)# tlv name address-type
Switch(config-sensor-cdplist)# tlv name device-name
Switch(config-sensor-cdplist)# tlv number 34
Switch(config-sensor-cdplist)# end
```

The following example shows how to create an LLDP filter containing a list of TLVs:

```
Switch> enable
Switch# configure terminal
Switch(config)# device-sensor filter-list lldp list lldp-list
Switch(config-sensor-llldplist)# tlv name chassis-id
Switch(config-sensor-llldplist)# tlv name management-address
Switch(config-sensor-llldplist)# tlv number 28
Switch(config-sensor-llldplist)# end
```

The following example shows how to create a DHCP filter containing a list of options:

```
Switch> enable
Switch# configure terminal
Switch(config)# device-sensor filter-list dhcp list dhcp-list
Switch(config)# device-sensor filter-list dhcp list dhcp-list
Switch(config-sensor-dhcplist)# option name domain-name
Switch(config-sensor-dhcplist)# option name host-name
Switch(config-sensor-dhcplist)# option number 50
Switch(config-sensor-dhcplist)# end
```

The following example shows how to apply a CDP TLV filter list to the Device Sensor output:

```
Switch> enable
Switch# configure terminal
Switch(config)# device-sensor filter-spec cdp include cdp-list1
```

The following example shows how to enable client notifications and accounting events for all TLV changes:

```
Switch> enable
Switch# configure terminal
Switch(config)# device-sensor notify all-changes
```

Displaying 802.1X Statistics and Status

To display 802.1X statistics for all interfaces, use the **show dot1x all statistics** privileged EXEC command.

To display the 802.1X administrative and operational status for the switch, use the **show dot1x all details** privileged EXEC command. To display the 802.1X administrative and operational status for a specific interface, use the **show dot1x interface details** privileged EXEC command.

Displaying Authentication Details

This section includes these topics:

- [Determining the Authentication Methods Registered with the Auth Manager, page 49-126](#)
- [Displaying the Auth Manager Summary for an Interface, page 49-127](#)
- [Displaying the Summary of All Auth Manager Sessions on the Switch, page 49-127](#)
- [Displaying a Summary of All Auth Manager Sessions on the Switch Authorized for a Specified Authentication Method, page 49-127](#)
- [Verifying the Auth Manager Session for an Interface, page 49-127](#)
- [Displaying MAB Details, page 49-129](#)
- [EPM Logging, page 49-130](#)

Determining the Authentication Methods Registered with the Auth Manager

This example show how to display the registered authentication methods:

Enter the following:

```
Switch# show authentication registrations
Handle Priority Name
      3         0 dot1x
      2         1 mab
      1         2 webauth
```

Displaying the Auth Manager Summary for an Interface

In the following example, MAB was configured for a higher priority (lower value) than 802.1X:

```
Switch# show authentication int gi1/5
Client list:
Interface  MAC Address      Method  Domain  Status      Session ID
Gi1/5      000f.23c4.a401    mab     DATA   Authz Success 0A3462B10000000D24F80B58
Gi1/5      0014.bf5d.d26d    dot1x   DATA   Authz Success 0A3462B10000000E29811B94

Available methods list:
Handle  Priority  Name
3       0        dot1x
2       1        mab

Runnable methods list:
Handle  Priority  Name
2       0        mab
3       1        dot1x
```

Displaying the Summary of All Auth Manager Sessions on the Switch

This example shows how to display the summary of all sessions:

```
Switch# show authentication sessions
Interface  MAC Address      Method  Domain  Status      Session ID
Gi1/48     0015.63b0.f676   dot1x   DATA   Authz Success 0A3462B1000000102983C05C
Gi1/5      000f.23c4.a401   mab     DATA   Authz Success 0A3462B10000000D24F80B58
Gi1/5      0014.bf5d.d26d   dot1x   DATA   Authz Success 0A3462B10000000E29811B94
```

Displaying a Summary of All Auth Manager Sessions on the Switch Authorized for a Specified Authentication Method

This example shows how to display a summary of all sessions for a specific authentication method:

```
Switch# show authentication method dot1x
Interface  MAC Address      Method  Domain  Status      Session ID
Gi1/48     0015.63b0.f676   dot1x   DATA   Authz Success 0A3462B1000000102983C05C
Gi1/5      0014.bf5d.d26d   dot1x   DATA   Authz Success 0A3462B10000000E29811B94
```

Verifying the Auth Manager Session for an Interface

The Auth manage session can be verified by using the **show authentication sessions** command:

```
Switch# show authentication sessions int gi1/5
Interface: GigabitEthernet1/5
MAC Address: 000f.23c4.a401
IP Address: Unknown
User-Name: 000f23c4a401
Status: Authz Success
Domain: DATA
Oper host mode: multi-auth
Oper control dir: both
Authorized By: Authentication Server
```

```

Vlan Policy: N/A
Session timeout: N/A
Idle timeout: N/A
Common Session ID: 0A3462B10000000D24F80B58
Acct Session ID: 0x0000000F
Handle: 0x2400000D
Runnable methods list:
Method State
dot1x Failed over
mab Authc Success
-----
Interface: GigabitEthernet1/5
MAC Address: 0014.bf5d.d26d
IP Address: 20.0.0.7
User-Name: johndoe
Status: Authz Success
Domain: DATA
Oper host mode: multi-auth
Oper control dir: both
Authorized By: Authentication Server
Vlan Policy: N/A
Session timeout: N/A
Idle timeout: N/A
Common Session ID: 0A3462B10000000E29811B94
Acct Session ID: 0x00000010
Handle: 0x1100000E
Runnable methods list:
Method State
dot1x Authc Success
mab Not run

```

The individual output can be further refined by using the **handle**, **interface**, **MAC**, **session-id**, or **method** keywords:

```

Switch# show authentication sessions mac 000f.23c4.a401
Interface: GigabitEthernet1/5
MAC Address: 000f.23c4.a401
IP Address: Unknown
User-Name: 000f23c4a401
Status: Authz Success
Domain: DATA
Oper host mode: multi-auth
Oper control dir: both
Authorized By: Authentication Server
Vlan Policy: N/A
Session timeout: N/A
Idle timeout: N/A
Common Session ID: 0A3462B10000000D24F80B58
Acct Session ID: 0x0000000F
Handle: 0x2400000D
Runnable methods list:
Method State
dot1x Failed over
mab Authc Success

Switch# show authentication sessions session-id 0A3462B10000000D24F80B58
Interface: GigabitEthernet1/5
MAC Address: 000f.23c4.a401
IP Address: Unknown
User-Name: 000f23c4a401

```

```

Status: Authz Success
Domain: DATA
Oper host mode: multi-auth
Oper control dir: both
Authorized By: Authentication Server
Vlan Policy: N/A
Session timeout: N/A
Idle timeout: N/A
Common Session ID: 0A3462B10000000D24F80B58
Acct Session ID: 0x0000000F
Handle: 0x2400000D
Runnable methods list:
Method State
dot1x Failed over
mab uthc Success

Switch# show authentication session method dot1x int gi1/5
Interface: GigabitEthernet1/5
MAC Address: 0014.bf5d.d26d
IP Address: 20.0.0.7
User-Name: johndoe
Status: Authz Success
Domain: DATA
Oper host mode: multi-auth
Oper control dir: both
Authorized By: Authentication Server
Vlan Policy: N/A
Session timeout: N/A
Idle timeout: N/A
Common Session ID: 0A3462B10000000E29811B94
Acct Session ID: 0x00000010
Handle: 0x1100000E
Runnable methods list:
Method State
dot1x Authc Success
mab Not run

```

Displaying MAB Details

The following commands display these details:

```

Switch# show mab all
MAB details for FastEthernet5/9
-----
Mac-Auth-Bypass           = Enabled
Inactivity Timeout        = None

Switch# show mab all detail
MAB details for FastEthernet5/9
-----
Mac-Auth-Bypass           = Enabled
Inactivity Timeout        = None
MAB Client List
-----
Client MAC                 = 000f.23c4.a401
MAB SM state               = TERMINATE
Auth Status                = AUTHORIZED

```

```

Switch# show mab int fa5/9
MAB details for FastEthernet5/9
-----
Mac-Auth-Bypass           = Enabled
Inactivity Timeout        = None

Switch# show mab int fa5/9 detail
MAB details for FastEthernet5/9
-----
Mac-Auth-Bypass           = Enabled
Inactivity Timeout        = None
MAB Client List
-----
Client MAC                 = 000f.23c4.a401
MAB SM state               = TERMINATE
Auth Status                = AUTHORIZED

```

EPM Logging

EPM logging enables you to display EPM logging messages by using the **epm logging** command in global configuration mode. To disable EPM logging, enter **no epm logging**.

Logging messages are displayed during the following events:

- **POLICY_APP_SUCCESS**—Policy application success events on Named ACLs, Proxy ACLs, and service policies, URL redirect policies.
- **POLICY_APP_FAILURE**—Policy application failure conditions similar to unconfigured policies, wrong policies, download request failures and download failures from AAA.
- **IPEVENT**—IP assignment, IP release and IP wait events for clients.
- **AAA**—AAA events (similar to download requests, or download successes from AAA)

Example 1

```

Switch# conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# epm logging
Switch# clear dot1x all
Switch#
*May 15 08:31:26.561: %EPM-6-POLICY_REQ: IP=100.0.0.222| MAC=0000.0000.0001|
AUDITSEID=0A050B2C000000030004956C| AUTHTYPE=DOT1X|
EVENT=REMOVE
*May 15 08:31:26.581: %AUTHMGR-5-START: Starting 'dot1x' for client (0000.0000.0001) on
Interface Fa9/25
*May 15 08:31:26.681: %DOT1X-5-SUCCESS: Authentication successful for client
(0000.0000.0001) on Interface Fa9/25
*May 15 08:31:26.681: %AUTHMGR-7-RESULT: Authentication result 'success' from 'dot1x' for
client (0000.0000.0001) on Interface Fa9/25

```


Example 2

```
Switch# conf terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# epm logging
Switch(config)# int f9/25
Switch(config-if)# shut
Switch(config-if)# no shut
*May 15 08:41:56.329: %EPM-6-IPEVENT: IP=100.0.0.222| MAC=0000.0000.0001|
AUDITSESID=0A050B2C0000026108FB7924| AUTHTYPE=DOT1X|
EVENT=IP-RELEASE
*May 15 08:41:56.333: %EPM-6-IPEVENT: IP=100.0.0.222| MAC=0000.0000.0001|
AUDITSESID=0A050B2C0000026108FB7924| AUTHTYPE=DOT1X|
EVENT=IP-WAIT
```

Cisco IOS Security Features

This document provides a list of security software features that are supported in Cisco IOS XE 3.1.0SG. Links to the feature documentation are included.

Feature guides may contain information about more than one feature. To find information about a specific feature within a feature guide, see the Feature Information table at the end of the guide.

Feature guides document features that are supported on many different software releases and platforms. Your Cisco software release or platform may not support all the features documented in a feature guide. See the Feature Information table at the end of the feature guide for information about which features in that guide are supported in your software release. Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Role-Based Access Control CLI Commands

http://www.cisco.com/en/US/docs/ios/sec_user_services/configuration/guide/sec_role_base_cli.html

Authentication Proxy Accounting for HTTP

http://www.cisco.com/en/US/docs/ios/sec_user_services/configuration/guide/sec_cfg_authen_prxy.html

Enhanced Password Security

http://www.cisco.com/en/US/docs/ios/sec_user_services/configuration/guide/sec_cfg_sec_4cli.html

IEEE 802.1X - Flexible Authentication

http://www.cisco.com/en/US/docs/ios/sec_user_services/configuration/guide/sec_cfg_authen_prxy.html

Image Verification

http://www.cisco.com/en/US/docs/ios/sec_user_services/configuration/guide/sec_image_verifctn.html

Manual Certificate Enrollment via TFTP

http://www.cisco.com/en/US/docs/ios/sec_secure_connectivity/configuration/guide/sec_cert_enroll_pk_i.html

Pre-fragmentation For Ipsec VPNs

http://www.cisco.com/en/US/docs/ios/sec_secure_connectivity/configuration/guide/sec_pre_frag_vpns.html

Router Security Audit Manageability

http://www.cisco.com/en/US/prod/collateral/routers/ps10537/product_bulletin_ISR2_Manageability.pdf

Trusted Root Certification Authority

http://www.cisco.com/en/US/docs/security/cta/admin_guide/ctaCerts.html



X.509v3 Certificates for SSH Authentication

The X.509v3 Certificates for SSH Authentication feature uses public key algorithm (PKI) for server and user authentication, and allows the Secure Shell (SSH) protocol to verify the identity of the owner of a key pair via digital certificates, signed and issued by a Certificate Authority (CA).

This module describes how to configure server and user certificate profiles for a digital certificate.

This module describes the feature and consists of these sections:

- [Prerequisites for X.509v3 Certificates for SSH Authentication, page 50-1](#)
- [Restrictions for X.509v3 Certificates for SSH Authentication, page 50-2](#)
- [Information About X.509v3 Certificates for SSH Authentication, page 50-2](#)
- [How to Configure X.509v3 Certificates for SSH Authentication, page 50-3](#)
- [Configuration Examples for X.509v3 Certificates for SSH Authentication, page 50-5](#)
- [Verifying Server and User Authentication Using Digital Certificates, page 50-6](#)
- [Additional References for X.509v3 Certificates for SSH Authentication, page 50-9](#)
- [Feature Information for X.509v3 Certificates for SSH Authentication, page 50-11](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

Prerequisites for X.509v3 Certificates for SSH Authentication

The X.509v3 Certificates for SSH Authentication feature replaces the **ip ssh server authenticate user** command with the **ip ssh server algorithm authentication** command. Configure the **default ip ssh server authenticate user** command to remove the **ip ssh server authenticate user** command from the configuration. The IOS secure shell (SSH) server will start using the **ip ssh server algorithm authentication** command.

When you configure the **ip ssh server authenticate user** command, the following message is displayed: “SSH command accepted; but this CLI will be deprecated soon. Please move to new CLI **ip ssh server algorithm authentication**. Please configure the “**default ip ssh server authenticate user**” to make the CLI ineffective.”

Restrictions for X.509v3 Certificates for SSH Authentication

- The X.509v3 Certificates for SSH Authentication feature implementation is applicable only on the Cisco IOS Secure Shell (SSH) server side.
- The Cisco IOS SSH server supports only the x509v3-ssh-rsa algorithm-based certificate for server and user authentication.
- The Rivest, Shamir, and Adelman (RSA) 2-factor authentication on Catalyst 4506 SUP7L-E switches and Cisco Identity Services Engine (ISE) does not work correctly, when a user enters the incorrect password. Normal authentication and interworking with Cisco Adaptive Security Appliance (ASA) works fine. Configure the **ip ssh server algorithm authentication keyboard** command for the authentication to work.

Information About X.509v3 Certificates for SSH Authentication

- [X.509v3 Certificates for SSH Authentication Overview, page 50-2](#)
- [Server and User Authentication Using X.509v3, page 50-2](#)
- [OCSP Response Stapling, page 50-3](#)

X.509v3 Certificates for SSH Authentication Overview

The Secure Shell (SSH) protocol provides a secure remote access connection to network devices. The communication between the client and server is encrypted.

There are two SSH protocols that use public key cryptography for authentication. The Transport Layer Protocol, uses a digital signature algorithm (called the public key algorithm) to authenticate the server to the client. And the User Authentication Protocol uses a digital signature to authenticate (public key authentication) the client to the server.

The validity of the authentication depends upon the strength of the linkage between the public signing key and the identity of the signer. Digital certificates, such as those in X.509 Version 3 (X.509v3), are used to provide identity management. X.509v3 uses a chain of signatures by a trusted root certification authority and intermediate certificate authorities to bind a public signing key to a specific digital identity. This implementation allows the use of a public key algorithm for server and user authentication, and allows SSH to verify the identity of the owner of a key pair via digital certificates, signed and issued by a Certificate Authority (CA).

Server and User Authentication Using X.509v3

For server authentication, the Secure shell (SSH) server sends its own certificate to the SSH client for verification. This server certificate is associated with the trustpoint configured in the server certificate profile (ssh-server-cert-profile-server configuration mode).

For user authentication, the SSH client sends the user's certificate to the IOS SSH server for verification. The SSH server validates the incoming user certificate using public key infrastructure (PKI) trustpoints configured in the server certificate profile (ssh-server-cert-profile-user configuration mode).

By default, certificate-based authentication is enabled for server and user at the IOS SSH server end.

OCSP Response Stapling

The Online Certificate Status Protocol (OCSP) enables applications to determine the (revocation) state of an identified certificate. This protocol specifies the data that needs to be exchanged between an application checking the status of a certificate and the server providing that status. An OCSP client issues a status request to an OCSP responder and suspends acceptance of the certificate until a response is received. An OCSP response at a minimum consists of a responseStatus field that indicates the processing status of the a request.


For the public key algorithms, the key format consists of a sequence of one or more X.509v3 certificates followed by a sequence of zero or more OCSP responses.


The X.509v3 Certificate for SSH Authentication feature uses OCSP Response Stapling. By using OCSP response stapling, a device obtains the revocation information of its own certificate by contacting the OCSP server and then stapling the result along with its certificates and sending the information to the peer rather than having the peer contact the OCSP responder.

How to Configure X.509v3 Certificates for SSH Authentication



- [Configuring Digital Certificates for Server Authentication, page 50-3](#)
- [Configuring Digital Certificates for User Authentication, page 50-4](#)



Configuring Digital Certificates for Server Authentication

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	Switch# configure terminal	Enters global configuration mode.
Step 3	Switch(config)# ip ssh server algorithm hostkey {x509v3-ssh-rsa [ssh-rsa] ssh-rsa [x509v3-ssh-rsa]}	Defines the order of host key algorithms. Only the configured algorithm is negotiated with the Secure Shell (SSH) client. <div>  <p>Note The IOS SSH server must have at least one configured host key algorithm.</p> </div> <ul style="list-style-type: none"> • x509v3-ssh-rsa—certificate-based authentication • ssh-rsa—public key-based authentication
Step 4	Switch(config)# ip ssh server certificate profile	Configures server and user certificate profiles and enters SSH certificate profile configuration mode.
Step 5	Switch(ssh-server-cert-profile)# server	Configures server certificate profile and enters SSH server certificate profile server configuration mode. <ul style="list-style-type: none"> • The server profile is used to send out the certificate of the server to the SSH client during server authentication.

	Command or Action	Purpose
Step 6	Switch(ssh-server-cert-profile-server)# trustpoint sign <i>PKI-trustpoint-name</i>	Attaches the public key infrastructure (PKI) trustpoint to the server certificate profile. <ul style="list-style-type: none"> The SSH server uses the certificate associated with this PKI trustpoint for server authentication.
Step 7	Switch(ssh-server-cert-profile-server)# ocsp-response include	(Optional) Sends the Online Certificate Status Protocol (OCSP) response or OCSP stapling along with the server certificate. <div>  <p>Note By default, no OCSP response is sent along with the server certificate.</p> </div>
Step 8	Switch(ssh-server-cert-profile-server)# end	Exits SSH server certificate profile server configuration mode and returns to privileged EXEC mode.
Step 9	Switch(config)# line vty <i>line_number</i> <i>[ending_line_number]</i>	Enters line configuration mode to configure the virtual terminal line settings. For <i>line_number</i> and <i>ending_line_number</i> , specify a pair of lines. The range is 0 to 15.
Step 10	Switch(config-line)# transport input ssh	Specifies that the Switch prevent non-SSH Telnet connections. This limits the router to only SSH connections.

Configuring Digital Certificates for User Authentication

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	Switch# configure terminal	Enters global configuration mode.
Step 3	Switch(config)# ip ssh server algorithm authentication { publickey keyboard password }	Defines the order of host key algorithms. Only the configured algorithm is negotiated with the Secure Shell (SSH) client. <div>  <p>Note The IOS SSH server must have at least one configured host key algorithm.</p> </div> <ul style="list-style-type: none"> To use the certificate method for user authentication, the publickey keyword must be configured.
Step 4	Switch(config)# ip ssh server algorithm publickey { x509v3-ssh-rsa [ssh-rsa] ssh-rsa [x509v3-ssh-rsa]}	Defines the order of public key algorithms. Only the configured algorithm is accepted by the SSH client for user authentication. <div>  <p>Note The IOS SSH client must have at least one configured public key algorithm.</p> </div> <ul style="list-style-type: none"> x509v3-ssh-rsa—Certificate-based authentication ssh-rsa—Public-key-based authentication

	Command or Action	Purpose
Step 5	Switch(config)# ip ssh server certificate profile	Configures server certificate profile and user certificate profile and enters SSH certificate profile configuration mode.
Step 6	Switch(ssh-server-cert-profile)# user	Configures user certificate profile and enters SSH server certificate profile user configuration mode.
Step 7	Switch(ssh-server-cert-profile-user)# trustpoint sign <i>PKI-trustpoint-name</i>	Configures the public key infrastructure (PKI) trustpoint that is used to verify the incoming user certificate. <div>  Note </div> Configure multiple trustpoints by executing the same command multiple times. A maximum of 10 trustpoints can be configured.
Step 8	Switch(ssh-server-cert-profile-user)# ocsp-response include	(Optional) Sends the Online Certificate Status Protocol (OCSP) response or OCSP stapling along with the server certificate. <div>  Note </div> By default, no OCSP response is sent along with the server certificate.
Step 9	Switch(ssh-server-cert-profile-user)# end	Exits SSH server certificate profile user configuration mode and returns to privileged EXEC mode.
Step 10	Switch(config)# line vty <i>line_number</i> [<i>ending_line_number</i>]	Enters line configuration mode to configure the virtual terminal line settings. For line_number and ending_line_number, specify a pair of lines. The range is 0 to 15.
Step 11	Switch(config-line)# transport input ssh	Specifies that the Switch prevent non-SSH Telnet connections. This limits the router to only SSH connections.

Configuration Examples for X.509v3 Certificates for SSH Authentication

- [Example: Configuring Digital Certificates for Server Authentication, page 50-5](#)
- [Example: Configuring Digital Certificate for User Authentication, page 50-5](#)

Example: Configuring Digital Certificates for Server Authentication

```
Switch> enable
Switch# configure terminal
Switch(config)# ip ssh server algorithm hostkey x509v3-ssh-rsa
Switch(config)# ip ssh server certificate profile
Switch(ssh-server-cert-profile)# server
Switch(ssh-server-cert-profile-server)# trustpoint sign trust1
Switch(ssh-server-cert-profile-server)# exit
```

Example: Configuring Digital Certificate for User Authentication

```
Switch> enable
```

```
Switch# configure terminal
Switch(config)# ip ssh server algorithm authentication publickey
Switch(config)# ip ssh server algorithm publickey x509v3-ssh-rsa
Switch(config)# ip ssh server certificate profile
Switch(ssh-server-cert-profile)# user
Switch(ssh-server-cert-profile-user)# trustpoint verify trust2
Switch(ssh-server-cert-profile-user)# end
```

Verifying Server and User Authentication Using Digital Certificates

Displays the currently configured authentication methods. To confirm the use of certificate-based authentication, ensure that the x509v3-ssh-rsa algorithm is the configured host key algorithm.

```
Switch# show ip ssh
```

```
SSH Enabled - version 1.99
Authentication methods:publickey,keyboard-interactive,password
Authentication Publickey Algorithms:x509v3-ssh-rsa,ssh-rsa
Hostkey Algorithms:x509v3-ssh-rsa,ssh-rsa
Authentication timeout: 120 secs; Authentication retries: 3
Minimum expected Diffie Hellman key size : 1024 bits
```

Use the **debug ip ssh detail** and **debug ip packet** debug commands to debug the SSH authentication using x.509v3 Certificates.

The following example shows the sample output for the **debug ip ssh detail** command:

```
Switch#debug ip ssh detail
ssh detail messages debugging is on
Switch#sh log
Syslog logging: enabled (0 messages dropped, 9 messages rate-limited, 0 flushes, 0
overruns, xml disabled, filtering disabled)
```

```
No Active Message Discriminator.
```

```
No Inactive Message Discriminator.
```

```
Console logging: disabled
Monitor logging: level debugging, 0 messages logged, xml disabled,
                  filtering disabled
Buffer logging:  level debugging, 233 messages logged, xml disabled,
                  filtering disabled
Exception Logging: size (4096 bytes)
Count and timestamp logging messages: disabled
File logging: disabled
Persistent logging: disabled
```

```
No active filter modules.
```

```
Trap logging: level informational, 174 message lines logged
Logging Source-Interface:      VRF Name:
```



```

Log Buffer (4096 bytes):
5 IST: SSH2 CLIENT 0: SSH2_MSG_KEXINIT sent
*Sep 6 14:44:08.496 IST: SSH0: protocol version id is - SSH-1.99-Cisco-1.25
*Sep 6 14:44:08.496 IST: SSH2 0: kexinit sent: kex algo =
diffie-hellman-group-exchange-sha1,diffie-hellman-group14-sha1
*Sep 6 14:44:08.496 IST: SSH2 0: Server certificate trustpoint not found. Skipping
hostkey algo = x509v3-ssh-rsa
*Sep 6 14:44:08.496 IST: SSH2 0: kexinit sent: hostkey algo = ssh-rsa
*Sep 6 14:44:08.496 IST: SSH2 0: kexinit sent: encryption algo =
aes128-ctr,aes192-ctr,aes256-ctr
*Sep 6 14:44:08.496 IST: SSH2 0: kexinit sent: mac algo =
hmac-sha2-256,hmac-sha2-512,hmac-sha1,hmac-sha1-96
*Sep 6 14:44:08.496 IST: SSH2 0: SSH2_MSG_KEXINIT sent
*Sep 6 14:44:08.496 IST: SSH2 0: SSH2_MSG_KEXINIT received
*Sep 6 14:44:08.496 IST: SSH2 0: kex: client->server enc:aes128-ctr mac:hmac-sha2-256
*Sep 6 14:44:08.496 IST: SSH2 0: kex: server->client enc:aes128-ctr mac:hmac-sha2-256
*Sep 6 14:44:08.496 IST: SSH2 0: Using hostkey algo = ssh-rsa
*Sep 6 14:44:08.496 IST: SSH2 0: Using kex_algo = diffie-hellman-group-exchange-sha1
*Sep 6 14:44:08.497 IST: SSH2 CLIENT 0: SSH2_MSG_KEXINIT received
*Sep 6 14:44:08.497 IST: SSH2 CLIENT 0: kex: server->client enc:aes128-ctr
mac:hmac-sha2-256
*Sep 6 14:44:08.497 IST: SSH2 CLIENT 0: kex: client->server enc:aes128-ctr
mac:hmac-sha2-256
*Sep 6 14:44:08.497 IST: SSH2 CLIENT 0: Using hostkey algo = ssh-rsa
*Sep 6 14:44:08.497 IST: SSH2 CLIENT 0: Using kex_algo =
diffie-hellman-group-exchange-sha1
*Sep 6 14:44:08.497 IST: SSH2 CLIENT 0: SSH2_MSG_KEX_DH_GEX_REQUEST sent
*Sep 6 14:44:08.497 IST: SSH2 CLIENT 0: Range sent- 2048 < 2048 < 4096
*Sep 6 14:44:08.497 IST: SSH2 0: SSH2_MSG_KEX_DH_GEX_REQUEST received
*Sep 6 14:44:08.497 IST: SSH2 0: Range sent by client is - 2048 < 2048 < 4096
*Sep 6 14:44:08.497 IST: SSH2 0: Modulus size established : 2048 bits
*Sep 6 14:44:08.510 IST: SSH2 0: expecting SSH2_MSG_KEX_DH_GEX_INIT
*Sep 6 14:44:08.510 IST: SSH2 CLIENT 0: SSH2_MSG_KEX_DH_GEX_GROUP received
*Sep 6 14:44:08.510 IST: SSH2 CLIENT 0: Server has chosen 2048 -bit dh keys
*Sep 6 14:44:08.523 IST: SSH2 CLIENT 0: expecting SSH2_MSG_KEX_DH_GEX_REPLY
*Sep 6 14:44:08.524 IST: SSH2 0: SSH2_MSG_KEXDH_INIT received
*Sep 6 14:44:08.555 IST: SSH2: kex_derive_keys complete
*Sep 6 14:44:08.555 IST: SSH2 0: SSH2_MSG_NEWKEYS sent
*Sep 6 14:44:08.555 IST: SSH2 0: waiting for SSH2_MSG_NEWKEYS
*Sep 6 14:44:08.555 IST: SSH2 CLIENT 0: SSH2_MSG_KEX_DH_GEX_REPLY received
*Sep 6 14:44:08.555 IST: SSH2 CLIENT 0: Skipping ServerHostKey Validation
*Sep 6 14:44:08.571 IST: SSH2 CLIENT 0: signature length 271
*Sep 6 14:44:08.571 IST: SSH2: kex_derive_keys complete
*Sep 6 14:44:08.571 IST: SSH2 CLIENT 0: SSH2_MSG_NEWKEYS sent
*Sep 6 14:44:08.571 IST: SSH2 CLIENT 0: waiting for SSH2_MSG_NEWKEYS
*Sep 6 14:44:08.571 IST: SSH2 CLIENT 0: SSH2_MSG_NEWKEYS received
*Sep 6 14:44:08.571 IST: SSH2 0: SSH2_MSG_NEWKEYS received
*Sep 6 14:44:08.571 IST: SSH2 0: Authentications that can continue =
publickey,keyboard-interactive,password
*Sep 6 14:44:08.572 IST: SSH2 0: Using method = none
*Sep 6 14:44:08.572 IST: SSH2 0: Authentications that can continue =
publickey,keyboard-interactive,password
*Sep 6 14:44:08.572 IST: SSH2 0: Using method = keyboard-interactive
*Sep 6 14:44:11.983 IST: SSH2 0: authentication successful for cisco
*Sep 6 14:44:11.984 IST: %SEC_LOGIN-5-LOGIN_SUCCESS: Login Success [user: cisco] [Source:
192.168.121.40] [localport: 22] at 14:44:11 IST Thu Sep 6 2018
*Sep 6 14:44:11.984 IST: SSH2 0: channel open request
*Sep 6 14:44:11.985 IST: SSH2 0: pty-req request
*Sep 6 14:44:11.985 IST: SSH2 0: setting TTY - requested: height 24, width 80; set:
height 24, width 80
*Sep 6 14:44:11.985 IST: SSH2 0: shell request
*Sep 6 14:44:11.985 IST: SSH2 0: shell message received
*Sep 6 14:44:11.985 IST: SSH2 0: starting shell for vty

```

```
*Sep  6 14:44:22.066 IST: %SYS-6-LOGOUT: User cisco has exited tty session
1(192.168.121.40)
*Sep  6 14:44:22.166 IST: SSH0: Session terminated normally
*Sep  6 14:44:22.167 IST: SSH CLIENT0: Session terminated normally
```

The following example shows the sample output for the **debug ip packet** command:

```
Switch#debug ip packet
IP packet debugging is on
Switch#sh log
Syslog logging: enabled (0 messages dropped, 9 messages rate-limited, 0 flushes, 0
overruns, xml disabled, filtering disabled)

No Active Message Discriminator.

No Inactive Message Discriminator.

Console logging: disabled
Monitor logging: level debugging, 0 messages logged, xml disabled,
                  filtering disabled
Buffer logging:  level debugging, 1363 messages logged, xml disabled,
                  filtering disabled
Exception Logging: size (4096 bytes)
Count and timestamp logging messages: disabled
File logging: disabled
Persistent logging: disabled

No active filter modules.

Trap logging: level informational, 176 message lines logged
Logging Source-Interface:      VRF Name:

Log Buffer (4096 bytes):
bleid=0, s=192.168.121.40 (local), d=192.168.121.40 (FortyGigabitEthernet1/0/1), routed
via RIB
*Sep  6 14:45:45.177 IST: IP: s=192.168.121.40 (local), d=192.168.121.40
(FortyGigabitEthernet1/0/1), len 40, sending
*Sep  6 14:45:45.177 IST: IP: s=192.168.121.40 (local), d=192.168.121.40
(FortyGigabitEthernet1/0/1), len 40, output feature, NAT Inside(8), rtype 1, forus FALSE,
sendself FALSE, mtu 0, fwdchk FALSE
*Sep  6 14:45:45.177 IST: IP: tableid=0, s=192.168.121.40 (FortyGigabitEthernet1/0/1),
d=192.168.121.40 (FortyGigabitEthernet1/0/1), routed via RIB
*Sep  6 14:45:45.177 IST: IP: tableid=0, s=192.168.121.40 (FortyGigabitEthernet1/0/1),
d=192.168.121.40 (FortyGigabitEthernet1/0/1), routed via RIB
*Sep  6 14:45:45.177 IST: IP: s=192.168.121.40 (local), d=192.168.121.40, len 40, local
feature, feature skipped, NAT(2), rtype 0, forus FALSE, sendself FALSE, mtu 0, fwdchk
FALSE
*Sep  6 14:45:45.178 IST: IP: tableid=0, s=192.168.121.40 (local), d=192.168.121.40
(FortyGigabitEthernet1/0/1), routed via RIB
*Sep  6 14:45:45.178 IST: IP: s=192.168.121.40 (local), d=192.168.121.40
(FortyGigabitEthernet1/0/1), len 40, sending
*Sep  6 14:45:45.178 IST: IP: s=192.168.121.40 (local), d=192.168.121.40
(FortyGigabitEthernet1/0/1), len 40, output feature, NAT Inside(8), rtype 1, forus FALSE,
sendself FALSE, mtu 0, fwdchk FALSE
*Sep  6 14:45:45.178 IST: IP: tableid=0, s=192.168.121.40 (FortyGigabitEthernet1/0/1),
d=192.168.121.40 (FortyGigabitEthernet1/0/1), routed via RIB
*Sep  6 14:45:45.178 IST: IP: s=192.168.121.40 (local), d=192.168.121.40, len 40, local
feature, feature skipped, NAT(2), rtype 0, forus FALSE, sendself FALSE, mtu 0, fwdchk
FALSE
*Sep  6 14:45:45.178 IST: IP: tableid=0, s=192.168.121.40 (local), d=192.168.121.40
(FortyGigabitEthernet1/0/1), routed via RIB
```

```

*Sep  6 14:45:45.178 IST: IP: s=192.168.121.40 (local), d=192.168.121.40
(FortyGigabitEthernet1/0/1), len 40, sending
*Sep  6 14:45:45.178 IST: IP: s=192.168.121.40 (local), d=192.168.121.40
(FortyGigabitEthernet1/0/1), len 40, output feature, NAT Inside(8), rtype 1, forus FALSE,
sendself FALSE, mtu 0, fwdchk FALSE
*Sep  6 14:45:45.178 IST: IP: tableid=0, s=192.168.121.40 (FortyGigabitEthernet1/0/1),
d=192.168.121.40 (FortyGigabitEthernet1/0/1), routed via RIB
*Sep  6 14:45:45.178 IST: IP: tableid=0, s=192.168.121.40 (FortyGigabitEthernet1/0/1),
d=192.168.121.40 (FortyGigabitEthernet1/0/1), routed via RIB
*Sep  6 14:45:45.178 IST: IP: tableid=0, s=192.168.121.40 (FortyGigabitEthernet1/0/1),
d=192.168.121.40 (FortyGigabitEthernet1/0/1), routed via RIB
*Sep  6 14:45:45.178 IST: IP: s=192.168.121.40 (local), d=192.168.121.40, len 40, local
feature, feature skipped, NAT(2), rtype 0, forus FALSE, sendself FALSE, mtu 0, fwdchk
FALSE
*Sep  6 14:45:45.178 IST: IP: tableid=0, s=192.168.121.40 (local), d=192.168.121.40
(FortyGigabitEthernet1/0/1), routed via RIB
*Sep  6 14:45:45.178 IST: IP: s=192.168.121.40 (local), d=192.168.121.40
(FortyGigabitEthernet1/0/1), len 40, sending
*Sep  6 14:45:45.178 IST: IP: s=192.168.121.40 (local), d=192.168.121.40
(FortyGigabitEthernet1/0/1), len 40, output feature, NAT Inside(8), rtype 1, forus FALSE,
sendself FALSE, mtu 0, fwdchk FALSE
*Sep  6 14:45:45.179 IST: IP: tableid=0, s=192.168.121.40 (FortyGigabitEthernet1/0/1),
d=192.168.121.40 (FortyGigabitEthernet1/0/1), routed via RIB
*Sep  6 14:45:45.179 IST: IP: s=192.168.121.40 (local), d=192.168.121.40, len 40, local
feature, feature skipped, NAT(2), rtype 0, forus FALSE, sendself FALSE, mtu 0, fwdchk
FALSE
*Sep  6 14:45:45.179 IST: IP: tableid=0, s=192.168.121.40 (local), d=192.168.121.40
(FortyGigabitEthernet1/0/1), routed via RIB
*Sep  6 14:45:45.179 IST: IP: s=192.168.121.40 (local), d=192.168.121.40
(FortyGigabitEthernet1/0/1), len 40, sending
*Sep  6 14:45:45.179 IST: IP: s=192.168.121.40 (local), d=192.168.121.40
(FortyGigabitEthernet1/0/1), len 40, output feature, NAT Inside(8), rtype 1, forus FALSE,
sendself FALSE, mtu 0, fwdchk FALSE
*Sep  6 14:45:45.179 IST: IP: tableid=0, s=192.168.121.40 (FortyGigabitEthernet1/0/1),
d=192.168.121.40 (FortyGigabitEthernet1/0/1), routed via RIB

```

Additional References for X.509v3 Certificates for SSH Authentication

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Catalyst 4500 switch commands	Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch
PKI configuration	Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment

Standards & MIBs

MIB	MIBs Link
•	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2784	Generic Routing Encapsulation (GRE)

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for X.509v3 Certificates for SSH Authentication

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

**Note**

Table 1 lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 1 Feature Information for X509v3 Certificates for SSH Authentication

Feature Name	Releases	Feature Information
X509v3 Certificates for SSH Authentication	Cisco IOS Release 15.2(4)E1 Cisco IOS XE Release 3.8.1E	The X.509v3 Certificates for SSH Authentication feature uses the X5.09v3 digital certificates in server and user authentication at the SSH server side. The following commands were introduced or modified: ip ssh server algorithm hostkey , ip ssh server algorithm authentication , and ip ssh server certificate profile .



Configuring SSH File Transfer Protocol

Secure Shell (SSH) includes support for SSH File Transfer Protocol (SFTP), which is a new standard file transfer protocol introduced in SSHv2. This feature provides a secure and authenticated method for copying device configuration or device image files.

This module describes the feature and consists of these sections:

- [Prerequisites for SSH File Transfer Protocol, page 51-1](#)
- [Restrictions for SSH File Transfer Protocol, page 51-1](#)
- [Information About SSH File Transfer Protocol, page 51-2](#)
- [How to Configure SSH File Transfer Protocol, page 51-2](#)
- [Example: Configuring SSH File Transfer Protocol, page 51-3](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see publications at this location:

[Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#)

If a command is not in the *Catalyst 4500 Series Switch Command Reference*, you can locate it in the Cisco IOS library, at this location:

[Cisco IOS Master Command List, All Releases](#)

Prerequisites for SSH File Transfer Protocol

- SSH must be enabled.
- The **ip ssh source-interface** *interface-type interface-number* command must be configured.

Restrictions for SSH File Transfer Protocol

- The SFTP server is not supported.
- SFTP boot is not supported.
- The **sftp** option in the **install add** command is not supported.

Information About SSH File Transfer Protocol

The SFTP client functionality is provided as part of the SSH component and is always enabled on the corresponding device. Therefore, any SFTP server user with the appropriate permission can copy files to and from the device.

An SFTP client is VRF-aware; you can configure the secure FTP client to use the virtual routing and forwarding (VRF) associated with a particular source interface during connection attempts.

How to Configure SSH File Transfer Protocol

The following sections provide information about the various tasks that comprise an SFTP configuration.

- [Configuring SFTP, page 51-2](#)
- [Perform an SFTP Copy Operation, page 51-2](#)

Configuring SFTP

Before you begin

To configure a Cisco device for SFTP client-side functionality, the **ip ssh source-interface interface-type interface-number** command must be configured first.

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	Switch# configure terminal	Enters global configuration mode.
Step 3	Switch(config)# ip ssh source-interface interface-type interface-number	Defines the source IP for the SSH session.
Step 4	Switch(config)# exit	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	Switch# show running-config	(Optional) Displays the SFTP client-side functionality.
Step 6	Switch# debug ip sftp	(Optional) Enables SFTP debugging.

Perform an SFTP Copy Operation

SFTP copy takes the IP or hostname of the corresponding server if Domain Name System (DNS) is configured. To perform SFTP copy operations, use the following commands in privileged EXEC mode.

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	Switch# copy ios-file-system:file sftp://user:pwd@server-ip//filepath Or Switch# copy ios-file-system: sftp:	Copies a file from the local Cisco IOS file system to the server. Specify the username, password, IP address, and filepath of the server.
Step 3	Switch# copy sftp://user:pwd@server-ip //filepath ios-file-system:file Or Switch# copy sftp: ios-file-system:	Copies the file from the server to the local Cisco IOS file system. Specify the username, password, IP address, and filepath of the server.

Example: Configuring SSH File Transfer Protocol

The following example shows how to configure the client-side functionality of SFTP:

```
Switch> enable
Switch# configure terminal
Switch(config)# ip ssh source-interface gigabitethernet 1/0/1
Switch(config)# end
```

■ Example: Configuring SSH File Transfer Protocol



Configuring the PPPoE Intermediate Agent

DSL Forum TR-101 [1] offers a means by which the PPPoE Discovery packets are tagged at the service provider's access switch with subscriber line specific information. The mechanism specifies using VSA of the PPPoE Discovery packets to add the line specific information at the switch. Even though you can perform Subscriber Line Identification (SLI) in another way (recreating virtual paths and circuits using stacked VLAN tags), DSL Forum 2004-071 [4] recommends the PPPoE Intermediate Agent mechanism. It cites lower provisioning costs and simpler co-ordination between OSS systems in charge of access switch and BRAS. PPPoE Intermediate Agent helps the service provider, BRAS, distinguish between end hosts connected over Ethernet to an access switch.

This chapter describes PPPoE Intermediate Agent on Catalyst 4500 series switches. It includes the following sections:

- [About PPPoE Intermediate Agent, page 52-2](#)
- [Enabling PPPoE IA on a Switch, page 52-2](#)
- [Configuring the Access Node Identifier for PPPoE IA on a Switch, page 52-2](#)
- [Configuring the Identifier String, Option, and Delimiter for PPPoE IA on a Switch, page 52-3](#)
- [Configuring the Generic Error Message for PPPoE IA on a Switch, page 52-3](#)
- [Enabling PPPoE IA on an Interface, page 52-4](#)
- [Configuring the PPPoE IA Trust Setting on an Interface, page 52-4](#)
- [Configuring PPPoE IA Rate Limiting Setting on an Interface, page 52-4](#)
- [Configuring PPPoE IA Vendor-tag Stripping on an Interface, page 52-5](#)
- [Configuring PPPoE IA Circuit-ID and Remote-ID on an Interface, page 52-5](#)
- [Enabling PPPoE IA for a Specific VLAN on an Interface, page 52-5](#)
- [Configuring PPPoE IA Circuit-ID and Remote-ID for a VLAN on an Interface, page 52-6](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

Related Documents

Related Topic	Document Title
PPPoE Circuit-Id Tag Processing	Refer to the PPOE Circuit-Id Tag Processing chapter
RADIUS attributes	Refer to the Cisco IOS Security Configuration Guide, Release 12.4
DSL Forum Line-ID tag solution	DSL Forum 2004-71: Solution for Remote-ID in PPPoE Discovery Phase
Migration to Ethernet-based DSL aggregation	DSL Forum Technical Report 101

RFCs

RFC	Title
RFC 2516	A Method for Transmitting PPP over Ethernet (PPPoE)

About PPPoE Intermediate Agent

PPPoE Intermediate Agent (PPPoE IA) is placed between a subscriber and BRAS to help the service provider BRAS distinguish between end hosts connected over Ethernet to an access switch. On the access switch, PPPoE IA enables Subscriber Line Identification by appropriately tagging Ethernet frames of different users. (The tag contains specific information such as which subscriber is connected to the switch and VLAN.) PPPoE IA acts as mini security firewall between host and BRAS by intercepting all PPPoE Active Discovery (PAD) messages on a per-port per-VLAN basis. It provides specific security feature such as verifying the intercepted PAD message from untrusted port, performing per-port PAD message rate limiting, inserting and removing VSA Tags into and from PAD messages, respectively.

Enabling PPPoE IA on a Switch

This functionality allows you to enable or disable PPPoE IA globally on the switch:

```
Switch> enable
Switch# configure terminal
Switch(config)# pppoe intermediate-agent
```

By default, PPPoE IA is disabled globally.

Configuring the Access Node Identifier for PPPoE IA on a Switch

This functionality allows you to set the Access Node Identifier of the switch. If unspecified, this parameter is derived automatically with the IP address of the management interface.

The following example shows how to set an access node identifier of **abcd**:

```
Switch> enable
Switch# configure terminal
Switch(config)# pppoe intermediate-agent format-type access-node-id string abcd
```

By default, *access-node-id* is not set.

Configuring the Identifier String, Option, and Delimiter for PPPoE IA on an Switch

This functionality overrides the default automatic generation of circuit-id by the system.

The options available are sp, sv, pv and spv denoting slot:port, slot-vlan, port-vlan, and slot-port-vlan combinations, respectively. Valid delimiters are # . , ; / space.

The **no** form of this command without *WORD*, *options*, and *delimiters*, reverts to the default automatic generation of circuit-id.

The following example shows how to set an identifier string **word** with option **spv** delimited by “:”:

```
Switch> enable
Switch# configure terminal
Switch(config)# pppoe intermediate-agent format-type
                identifier-string string word
                option spv delimiter :
```

This command does not affect the circuit ID configured explicitly per-interface or per-interface per-VLAN with the **pppoe intermediate-agent format-type circuit-id** or **pppoe intermediate-agent vlan num format-type circuit-id** commands.

Configuring the Generic Error Message for PPPoE IA on an Switch

This functionality sets the Generic-Error message of the switch. PPPoE IA sends this message only on a specific error condition. If you do not specify **string {WORD}**, the error message is not added.

The following example shows how to configure a generic message of **packet_length>1484**:

```
Switch> enable
Switch# configure terminal
Switch(config)# pppoe intermediate-agent format-type
                generic-error-message string packet_length>1484
PPPoE Discover packet too large to process. Try reducing the number of tags added.
```

By default the **generic-error-message** is not set. The **string** value is converted to UTF-8 before it is added to the response. The message similar to the following will appear:

PPPoE Discover packet too large to process. Try reducing the number of tags added.



Note

This TAG (0x0203 Generic-Error) indicates an error. It can be added to PADO or PADS packets generated by PPPoE IA and then sent back to user in reply of PADI or PADR, when a PPPoE discovery packet received by PPPoE IA with PPPoE payload greater than 1484 bytes. Error data must be a UTF-8 string.

Enabling PPPoE IA on an Interface

This functionality enables the PPPoE IA feature on an interface. The **pppoe intermediate-agent** command has an effect only if the PPPoE IA feature was enabled globally with this command. (You need to enable *globally* to activate PPPoE IA static ACL and on an interface for PPPoE IA processing of PPPoE discovery packets received on that interface.)

This setting applies to all frames passing through this interface, regardless of the VLAN they belong to. By default the PPPoE IA feature is disabled on all interfaces. You need to run this command on every interface that requires this feature.

The following example shows how to enable PPPoE IA on FastEthernet 3/1:

```
Switch> enable
Switch# configure terminal
Switch(config) interface FastEthernet 3/1
Switch(config-if) pppoe intermediate-agent
```

**Note**

Enabling PPPoE IA on an interface does not ensure that incoming packets are tagged. For this to happen PPPoE IA must be enabled globally, and at least one interface that connects the switch to PPPoE server has a trusted PPPoE IA setting. Refer to the following section for details.

Configuring the PPPoE IA Trust Setting on an Interface

This functionality sets a physical interface as trusted. The following example shows how to set FastEthernet interface 3/2 as trusted:

```
Switch> enable
Switch# configure terminal
Switch(config) interface FastEthernet 3/2
Switch(config-if) pppoe intermediate-agent trust
```

This setting is disabled by default.

**Note**

Interfaces that connect the switch to PPPoE server are configured as trusted. Interfaces that connect the switch to users (PPPoE clients) are untrusted.

Configuring PPPoE IA Rate Limiting Setting on an Interface

This functionality limits the rate (per second) at which PPPoE Discovery packets (PADI, PADO, PADR, PADS, or PADT) are received on an interface. When the incoming packet rate achieves or exceeds the configured limit, a port enters an err-disabled state.

The following example shows how to set a rate limit of 30 at FastEthernet 3/1:

```
Switch> enable
Switch# configure terminal
Switch(config) interface FastEthernet 3/1
Switch(config-if) pppoe intermediate-agent limit rate 30
```

**Note**

The parameter for rate limiting is the number of packets per second. If the incoming packet rate exceeds this value, the port shuts down.

Configuring PPPoE IA Vendor-tag Stripping on an Interface

This functionality enables an administrator to strip the vendor-specific tag (VSA) from PADO, PADS, and PADT packets received on an interface before forwarding them to the user.

The following example shows how to enable stripping on FastEthernet 3/2:

```
Switch> enable
Switch# configure terminal
Switch(config) interface FastEthernet 3/2
Switch(config-if) pppoe intermediate-agent vendor-tag strip
```

This setting is disabled by default.



Note

Generally, you would configure vendor-tag stripping on an interfaces connected to the PPPoE server. If you configure stripping, incoming packets are stripped of their VSAs (which carry subscriber and line identification information). For this to happen, the PPPoE Intermediate agent must be enabled to make the **pppoe intermediate-agent vendor-tag strip** command effective, and the interface must be set to trust. In isolation, the command has no effect.

Configuring PPPoE IA Circuit-ID and Remote-ID on an Interface

The **[no] pppoe intermediate-agent format-type circuit-id** command sets the circuit ID on an interface and overrides the automatic generation of circuit ID by the switch. Without this command, one default tag (for example, Ethernet x/y:z on the PPPoE to which the user is connected) inserted by an intermediate-agent.

The **[no] pppoe intermediate-agent format-type remote-id** command sets the remote ID on an interface.

This functionality causes tagging of PADI, PADR, and PADT packets (belonging to PPPoE Discovery stage) received on this physical interface with circuit ID or remote ID. This happens regardless s of their VLAN if PPPoE IA is not enabled for that VLAN.

You should use remote ID instead of circuit ID for subscriber line identification. You should configure this setting on every interface where you enabled PPPoE IA because it is not set by default. The default value for remote-id is the switch MAC address (for all physical interfaces).

The following example shows how to configure the circuit ID as root and the remote ID as granite:

```
Switch> enable
Switch# configure terminal
Switch(config) interface FastEthernet 3/1
Switch(config-if) pppoe intermediate-agent format-type circuit-id string root
Switch(config-if) pppoe intermediate-agent format-type remote-id string granite
```

Enabling PPPoE IA for a Specific VLAN on an Interface

This functionality allows you to enable PPPoE IA on either a specific VLAN, a comma-separated list such as “x,y,” or a range such as “x-y.”

Specific VLAN:

```
Switch# configure terminal
Switch(config)# interface FastEthernet 3/1
Switch(config-if)# vlan-range 5
Switch(config-if-vlan-range)# pppoe intermediate-agent
```

Comma-separated VLAN list:

```
Switch# configure terminal
Switch(config)# interface FastEthernet 3/1
Switch(config-if)# vlan-range 5,6
Switch(config-if-vlan-range)# pppoe intermediate-agent
```

VLAN range:

```
Switch# configure terminal
Switch(config)# interface FastEthernet 3/1
Switch(config-if)# vlan-range 5-9
Switch(config-if-vlan-range)# pppoe intermediate-agent
```



Note

The **pppoe intermediate-agent** command in the vlan-range mode is not dependent on the same command in interface mode. The **pppoe intermediate-agent** command will take effect independently of the command in the interface mode. To make this happen, PPPoE IA must be enabled globally and at least one interface is connected to the PPPoE server.

Configuring PPPoE IA Circuit-ID and Remote-ID for a VLAN on an Interface

In this section you set the circuit ID and remote ID for a specific VLAN on an interface. The command overrides the circuit ID and remote ID specified for this physical interface and the switch uses the *WORD* value to tag packets received on this VLAN. This parameter is unset by default.

The default value of **remote-id** is the switch MAC address (for all VLANs). You would set this parameter to encode subscriber-specific information.



Note

The **circuit-id** and **remote-id** configurations in vlan-range mode are affected only if PPPoE IA is enabled globally and in vlan-range mode.

This example shows how to set the circuit-id to aaa and the remote-id as ccc on interface g3/7:

```
Switch(config)# int g3/7
Switch(config-if)# vlan-range 5
Switch(config-if)# pppoe intermediate-agent
Switch(config-if-vlan-range)# pppoe intermediate-agent format-type circuit-id string aaa
Switch(config-if-vlan-range)# pppoe intermediate-agent format-type remote-id string ccc
```



Note

The **vlan-range** mode commands configure PPPoE IA for either a specific VLAN, multiple VLANs, or VLAN range, depending on what you specify in the syntax.

Displaying Configuration Parameters

The **show pppoe intermediate-agent [info|statistics] [interface {interface}]** command displays the various configuration parameters, statistics, and counters stored for PPPoE.

The **info** keyword appears if the PPPoE Intermediate Agent is enabled globally on an interface or on a VLAN (in an interface). It also informs you about the access node ID and generic error message of the switch, as well as the identifier string options and delimiter values configured globally by the following command:

```
Switch(config)# pppoe intermediate-agent format-type ?
  access-node-id      Access Node Identifier
  generic-error-message  Generic Error Message
  identifier-string    Identifier String
```

The **info** keyword also displays the circuit ID, remote ID, trust and rate limit configurations, and vendor tag strip setting for all interfaces and for all VLANs pertaining to those interfaces. If any of these parameters are not set, they are not displayed.

The **statistics** option displays the number of PADI/PADR/PADT packets received, and the time the last packet was received on all interfaces and on all VLANs pertaining to those interfaces.

If **interface** is specified, information or statistics applicable only to that physical interface and pertaining VLANs is displayed.

Although PPoE IA is supported on PVLANS, be aware that no PVLAN association (primary and secondary VLAN mapping) information is displayed.

The PPPoE IA show commands such as **show pppoe intermediate-agent info**, **show pppoe intermediate-agent info interface g3/7**, or **show pppoe intermediate-agent statistics** do not provide information about private VLAN association (primary and secondary VLAN mapping). However, they do provide information about VLANs regardless of private or normal VLANs, as the following example illustrate:

```
Switch# show pppoe intermediate-agent info
Switch PPPOE Intermediate-Agent is enabled
```

PPPOE Intermediate-Agent trust/rate is configured on the following Interfaces:

Interface	IA	Trusted	Vsa Strip	Rate limit (pps)
GigabitEthernet3/4	no	yes	yes	unlimited

PPPOE Intermediate-Agent is configured on following VLANs:
2-3

Interface	IA	Trusted	Vsa Strip	Rate limit (pps)
GigabitEthernet3/7	no	no	no	unlimited

PPPOE Intermediate-Agent is configured on following VLANs:
2-3

```
Switch# show pppoe intermediate-agent info interface g3/7
Interface      IA      Trusted  Vsa Strip  Rate limit (pps)
-----
GigabitEthernet3/7  yes    no       no         unlimited
PPPoE Intermediate-Agent is configured on following VLANs:
2-3
```

```
Switch# show pppoe intermediate-agent statistics
```

```
PPPOE IA Per-Port Statistics
-----
```

```
Interface : GigabitEthernet3/7
Packets received
All = 0
PADI = 0 PADO = 0
PADR = 0 PADS = 0
PADT = 0
Packets dropped:
Rate-limit exceeded = 0
```

```

Server responses from untrusted ports = 0
Client requests towards untrusted ports = 0
Malformed PPPoE Discovery packets = 0
Vlan 2: Packets received PADI = 0 PADO = 0 PADR = 0 PADS = 0 PADT = 0
Vlan 3: Packets received PADI = 0 PADO = 0 PADR = 0 PADS = 0 PADT = 0

Switch# show pppoe intermediate-agent statistics interface g3/7
Interface : GigabitEthernet3/7
Packets received
  All = 3
  PADI = 0 PADO = 0
  PADR = 0 PADS = 0
  PADT = 3
Packets dropped:
  Rate-limit exceeded = 0
  Server responses from untrusted ports = 0
  Client requests towards untrusted ports = 0
  Malformed PPPoE Discovery packets = 0
Vlan 2: Packets received PADI = 6 PADO = 0 PADR = 6 PADS = 0 PADT = 6
Vlan 3: Packets received PADI = 4 PADO = 0 PADR = 4 PADS = 0 PADT = 4

```

Clearing Packet Counters

This section illustrates how to clear packet counters on all interfaces (per-port and per-port-per-VLAN).

The following example illustrates how to do this:

```
Switch# clear pppoe intermediate-agent statistics
```

Issuing of the above command clears the counters for all PPPoE discovery packets (PADI,PADO,PADR,PADS,PADT) received on DUT.

```

Switch# show pppoe intermediate-agent statistics interface g3/7
Interface : GigabitEthernet3/7
Packets received
  All = 0
  PADI = 0 PADO = 0
  PADR = 0 PADS = 0
  PADT = 0
Packets dropped:
  Rate-limit exceeded = 0
  Server responses from untrusted ports = 0
  Client requests towards untrusted ports = 0
  Malformed PPPoE Discovery packets = 0
Vlan 2: Packets received PADI = 0 PADO = 0 PADR = 0 PADS = 0 PADT = 0
Vlan 3: Packets received PADI = 0 PADO = 0 PADR = 0 PADS = 0 PADT = 0

```

Debugging PPPoE Intermediate Agent

The **debug pppoe intermediate-agent [packet | event | all]** command enables you to display useful PPPoE information that assists in debugging. This command is disabled by default.

The **packet** option of the command displays the contents of a packet received in the software: source and destination MAC address of Ethernet frame, code, version and type of PPPoE Discovery packet and a list of TAGs present.

The **event** option of the command echoes important messages (interface state change to errdisabled due to PPPoE discovery packets entering at a rate exceeding the configured limit). It is the only event shown by the **debug pppoe intermediate-agent event** command.

The **all** option enables both package and event options.

The following example illustrates how to enter the debug command with the **packet** option:

```
Switch# debug pppoe intermediate-agent packet
PPPOE IA Packet debugging is on

*Sep  2 06:12:56.133: PPPOE_IA: Process new PPPoE packet, Message type: PADI, input
interface: Gi3/7, vlan : 2 MAC da: ffff.ffff.ffff, MAC sa: aabb.cc00.0000
*Sep  2 06:12:56.137: PPPOE_IA: received new PPPOE packet from inputinterface
(GigabitEthernet3/4)
*Sep  2 06:12:56.137: PPPOE_IA: received new PPPOE packet from inputinterface
(GigabitEthernet3/8)
*Sep  2 06:12:56.137: PPPOE_IA: Process new PPPoE packet, Message type: PADO, input
interface: Gi3/4, vlan : 2 MAC da: aabb.cc00.0000, MAC sa: 001d.e64c.6512
*Sep  2 06:12:56.137: PPPOE_IA: Process new PPPoE packet, Message type: PADO, input
interface: Gi3/8, vlan : 2 MAC da: aabb.cc00.0000, MAC sa: aabb.cc80.0000
*Sep  2 06:12:56.137: PPPOE_IA: received new PPPOE packet from inputinterface
(GigabitEthernet3/7)
*Sep  2 06:12:56.137: PPPOE_IA: Process new PPPoE packet, Message type: PADR, input
interface: Gi3/7, vlan : 2 MAC da: 001d.e64c.6512, MAC sa: aabb.cc00.0000
*Sep  2 06:12:56.145: PPPOE_IA: received new PPPOE packet from inputinterface
(GigabitEthernet3/4)
*Sep  2 06:12:56.145: PPPOE_IA: Process new PPPoE packet, Message type: PADS, input
interface: Gi3/4, vlan : 2 MAC da: aabb.cc00.0000, MAC sa: 001d.e64c.6512
```

The following example illustrates how to enter the debug command with the **event** option:

```
Switch# debug pppoe intermediate-agent event
PPPOE IA Event debugging is on

*Jul 30 19:00:10.254: %PPPOE_IA-4-PPPOE_IA_ERRDISABLE_WARNING: PPPOE IA received 5 PPPOE
packets on interface Gi3/7
*Jul 30 19:00:10.254: %PPPOE_IA-4-PPPOE_IA_RATE_LIMIT_EXCEEDED: The interface Gi3/7 is
receiving more than the threshold set
*Jul 30 19:00:10.394: %PM-4-ERR_DISABLE: STANDBY:pppoe-ia-rate-limit error detected on
Gi3/7, putting Gi3/7 in err-disable stat
```

Troubleshooting Tips

When the **radius-server attribute 31 remote-id** global configuration command is entered in the PPPoE Agent Remote-ID Tag and DSL Line Characteristics feature configuration on the BRAS, the **debug radius** privileged EXEC command can be used to generate a report that includes information about the incoming access interface, where discovery frames are received, and about the session being established in PPPoE extended NAS-Port format (format d).



Configuring Web-Based Authentication

This chapter describes how to configure web-based authentication. It consists of these sections:

- [About Web-Based Authentication, page 53-1](#)
- [Configuring Web-Based Authentication, page 53-6](#)
- [Displaying Web-Based Authentication Status, page 53-14](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About Web-Based Authentication

The web-based authentication feature, known as Web Authentication Proxy, enables you to authenticate end users on host systems that do not run the IEEE 802.1X supplicant.



Note

You can configure web-based authentication on Layer 2 and Layer 3 interfaces.

When you initiate an HTTP session, web-based authentication intercepts ingress HTTP packets from the host and sends an HTML login page to the user. The user keys in their credentials, which the web-based authentication feature sends to the AAA server for authentication:

- If authentication succeeds, web-based authentication sends a Login-Successful HTML page to the host and applies the access policies returned by the AAA server.
- If authentication fails, web-based authentication forwards a Login-Fail HTML page to the user, prompting the user to retry the login. If the user exceeds the maximum number of attempts, web-based authentication forwards a Login-Expired HTML page to the host and the user is placed on a watch-list for a waiting period.

These sections describe the role of web-based authentication as part of the authentication, authorization, and accounting (AAA) system:

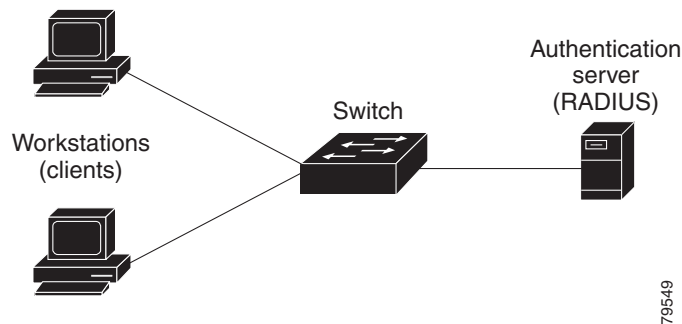
- [Device Roles, page 53-2](#)
- [Host Detection, page 53-2](#)
- [Session Creation, page 53-3](#)

- [Authentication Process, page 53-3](#)
- [Customization of the Authentication Proxy Web Pages, page 53-4](#)
- [Web-Based Authentication Interactions with Other Features, page 53-4](#)

Device Roles

With web-based authentication, the devices in the network have specific roles ([Figure 53-1](#)).

Figure 53-1 Web-Based Authentication Device Roles



The roles are as follows:

- *Client*—The device (workstation) that requests access to the LAN and switch services and responds to requests from the switch. The workstation must be running an HTML browser with Java Script enabled.
- *Authentication server*—Performs the actual authentication of the client. The authentication server validates the identity of the client and notifies the switch that the client is authorized to access the LAN and switch services or that the client is denied.
- *Switch*—Controls the physical access to the network based on the authentication status of the client. The switch acts as an intermediary (proxy) between the client and the authentication server, requesting identity information from the client, verifying that information with the authentication server, and relaying a response to the client.

Host Detection

The switch maintains an IP device tracking table to store information about detected hosts.



Note

By default, the IP device tracking feature is disabled on a switch. You must enable the IP device tracking feature to use web-based authentication.

For Layer 3 interfaces, web-based authentication sets an HTTP intercept ACL when the feature is configured on the interface (or when the interface is put in service).

For Layer 2 interfaces, web-based authentication detects IP hosts using the following mechanisms:

- ARP-based trigger—ARP redirect ACL allows web-based authentication to detect hosts with static IP address or dynamically acquired IP address.
- Dynamic ARP inspection (DAI)

- DHCP snooping—Web-based authentication is notified when the switch creates a DHCP binding entry for the host.

Session Creation

When web-based authentication detects a new host, it creates a session as follows:

- Checks for Auth bypass

If the host IP is not on the exception list, web-based authentication sends a nonresponsive host (NRH) request to the server.

If the server response is Access Accepted, authorization is bypassed for this host. The session is established.
- Sets up the HTTP Intercept ACL

If the server response to the NRH request is Access Rejected, the HTTP intercept ACL is activated and the session waits for HTTP traffic from the host.

Authentication Process

When you enable web-based authentication, the following events occur:

- The user initiates an HTTP session.
- The HTTP traffic is intercepted, and authorization is initiated. The switch sends the login page to the user. The user enters a username and password on the login page, and the switch sends the entries to the authentication server.
- If the client identity is valid and the authentication succeeds, the switch downloads and activates the user's access policy from the authentication server. The login success page is sent to the user.
- If the authentication fails, the switch sends the login fail page. The user retries the login. If the maximum number of attempts fails, the switch sends the login expired page and the host is placed in a watch list, using the following commands:

ip admission watch-list enable

ip admission watch-list expiry-time <milliseconds>

- After the watch list times out, the user can retry the authentication process.
- If the authentication server does not respond to the switch, and if an AAA fail policy is configured, the switch applies the failure access policy to the host. The login success page is sent to the user. See the [“Customization of the Authentication Proxy Web Pages”](#) section on page 53-4.
- The switch reauthenticates a client when the host does not respond to an ARP probe on a Layer 2 interface, or the host does not send any traffic within the idle timeout on a Layer 3 interface.
- The feature applies the downloaded timeout or the locally configured session timeout.
- If the terminate action is RADIUS, the feature sends a nonresponsive host (NRH) request to the server. The terminate action is included in the response from the server.
- If the terminate action is default, the session is dismantled and the applied policy is removed.

Customization of the Authentication Proxy Web Pages

During the web-based authentication process, the internal HTTP server of the switch hosts four HTML pages for delivery to an authenticating client. The four pages allow the server to notify you of the following four states of the authentication process:

- Login—Your credentials are requested.
- Success—The login was successful.
- Fail—The login failed.
- Expire—The login session has expired because of excessive login failures.

**Note**

When your customized web-based authentication page is replaced with a new page (file) of the same name in the switch system directory (i.e. flash), the new page will not be seen; you will see the older page. Beginning with Release 15.0(2)SG, the new page will not display until you enter the **ip admission proxy http refresh-all** command.

In Cisco IOS Release 12.2(50)SG, you can substitute your custom HTML pages for the four default internal HTML pages, or you can specify a URL to which you are redirected upon successful authentication, effectively replacing the internal Success page.

Web-Based Authentication Interactions with Other Features

These sections describe web-based authentication interactions with these features:

- [Port Security, page 53-4](#)
- [LAN Port IP, page 53-5](#)
- [Gateway IP, page 53-5](#)
- [ACLs, page 53-5](#)
- [Context-Based Access Control, page 53-5](#)
- [802.1X Authentication, page 53-5](#)
- [EtherChannel, page 53-5](#)
- [Switchover, page 53-5](#)

Port Security

You can configure web-based authentication and port security on the same port. (You configure port security on the port with the **switchport port-security** interface configuration command.) When you enable port security and web-based authentication on a port, web-based authentication authenticates the port, and port security manages network access for all MAC addresses, including that of the client. You can then limit the number or group of clients that can access the network using the port.

For more information about enabling port security, see [Chapter 55, “Configuring Port Security.”](#)

LAN Port IP

You can configure LAN port IP (LPIP) and Layer 2 web-based authentication on the same port. The host is authenticated using web-based authentication first, followed by LPIP posture validation. The LPIP host policy overrides the web-based authentication host policy.

If the web-based authentication idle timer expires, the NAC policy is removed. The host is authenticated and posture is validated again.

Gateway IP

You cannot configure Gateway IP on a Layer 3 VLAN interface if web-based authentication is configured on any of the switch ports in the VLAN.

You can configure web-based authentication on the same Layer 3 interface as Gateway IP. The host policies for both features are applied in software. The GWIP policy overrides the web-based authentication host policy.

ACLs

If you configure a VLAN ACL or Cisco IOS ACL on an interface, the ACL is applied to the host traffic only after the web-based authentication host policy is applied.

For Layer 2 web-based authentication, you must configure a port ACL (PACL) as the default access policy for ingress traffic from hosts connected to the port. After authentication, the web-based authentication host policy overrides the PACL.

You cannot configure a MAC ACL and web-based authentication on the same interface.

You cannot configure web-based authentication on a port whose access VLAN is configured for VACL capture.

Context-Based Access Control

Web-based authentication cannot be configured on a Layer 2 port if context-based access control (CBAC) is configured on the Layer 3 VLAN interface of the port's VLAN.

802.1X Authentication

You cannot configure web-based authentication on the same port as 802.1X authentication except as a fallback authentication method.

EtherChannel

You can configure web-based authentication on a Layer 2 EtherChannel interface. The web-based authentication configuration applies to all member channels.

Switchover

On Catalyst 4500 series switches with redundant supervisor engines in RPR mode, information about currently authenticated hosts is maintained during a switchover. You do not need to reauthenticate.

Configuring Web-Based Authentication

These sections describe how to configure web-based authentication:

- [Default Web-Based Authentication Configuration, page 53-6](#)
- [Web-Based Authentication Configuration Guidelines and Restrictions, page 53-6](#)
- [Web-Based Authentication Configuration Task List, page 53-7](#)
- [Configuring the Authentication Rule and Interfaces, page 53-7](#)
- [Configuring AAA Authentication, page 53-9](#)
- [Configuring Switch-to-RADIUS-Server Communication, page 53-9](#)
- [Configuring the HTTP Server, page 53-11](#)
- [Configuring the Web-Based Authentication Parameters, page 53-13](#)
- [Removing Web-Based Authentication Cache Entries, page 53-14](#)

Default Web-Based Authentication Configuration

Table 53-1 shows the default web-based authentication configuration.

Table 53-1 **Default Web-based Authentication Configuration**

Feature	Default Setting
AAA	Disabled
RADIUS server	
• IP address	• None specified
• UDP authentication port	• 1812
• Key	• None specified
Default value of inactivity timeout	3600 seconds
Inactivity timeout	Enabled

Web-Based Authentication Configuration Guidelines and Restrictions

When configuring web-based authentication, consider these guidelines and restrictions:

- Web authentication requires two Cisco Attribute-Value (AV) pair attributes:

The first attribute, `priv-lvl=15`, must always be set to 15. This sets the privilege level of the user who is logging into the switch.

The second attribute is an access list to be applied for web-authenticated hosts. The syntax is similar to 802.1x per-user access control lists (ACLs). However, instead of `ip:inacl`, this attribute must begin with `proxyacl`, and the source field in each entry must be any. (After authentication, the client IP address replaces the any field when the ACL is applied.)

For example:

```
proxyacl# 10=permit ip any 10.0.0.0 255.0.0.0
proxyacl# 20=permit ip any 11.1.0.0 255.255.0.0
proxyacl# 30=permit udp any any eq syslog
```

```
proxyacl# 40=permit udp any any eq tftp
```



Note The proxyacl entry determines the type of allowed network access.

- Web-based authentication is an ingress-only feature.
- You can configure web-based authentication only on access ports. Web-based authentication is not supported on trunk ports, EtherChannel member ports, or dynamic trunk ports.
- You must configure the default ACL on the interface before configuring web-based authentication. Configure a port ACL for a Layer 2 interface, or a Cisco IOS ACL for a Layer 3 interface.
- On Layer 2 interfaces, you cannot authenticate hosts with static ARP cache assignment. These hosts are not detected by the web-based authentication feature, because they do not send ARP messages.
- By default, the IP device tracking feature is disabled on a switch. You must enable the IP device tracking feature to use web-based authentication.
- You must configure at least one IP address to run the HTTP server on the switch. You must also configure routes to reach each host IP address. The HTTP server sends the HTTP login page to the host.
- Hosts that are more than one hop away may experience traffic disruption if an STP topology change results in the host traffic arriving on a different port. It is because ARP and DHCP updates may not be sent after a Layer 2 (STP) topology change.
- Web-based authentication does not support VLAN assignment as a downloadable host policy.
- Cisco IOS Release 12.2(50)SG supports downloadable ACLs (DACLS) from the RADIUS server.
- Web-based authentication is not supported for IPv6 traffic.

Web-Based Authentication Configuration Task List

To configure the web-based authentication feature, perform the following tasks:

- [Configuring the Authentication Rule and Interfaces, page 53-7](#)
- [Configuring AAA Authentication, page 53-9](#)
- [Configuring Switch-to-RADIUS-Server Communication, page 53-9](#)
- [Configuring the HTTP Server, page 53-11](#)
- [Configuring the Web-Based Authentication Parameters, page 53-13](#)
- [Removing Web-Based Authentication Cache Entries, page 53-14](#)

Configuring the Authentication Rule and Interfaces

To configure web-based authentication, perform this task:

	Command	Purpose
Step 1	Switch(config)# ip admission name name proxy http	Configures an authentication rule for web-based authorization.
	Switch(config)# no ip admission name name	Removes the authentication rule.

	Command	Purpose
Step 2	Switch(config)# interface <i>type slot/port</i>	Enters interface configuration mode and specifies the ingress Layer 2 or Layer 3 interface to be enabled for web-based authentication. <i>type</i> can be fastethernet, gigabit ethernet, or tengigabitethernet
Step 3	Switch(config-if)# ip access-group <i>name</i>	Applies the default ACL.
Step 4	Switch(config-if)# ip admission <i>name</i>	Configures web-based authentication on the specified interface.
Step 5	Switch(config-if)# exit	Returns to configuration mode.
Step 6	Switch(config)# ip device tracking	Enables the IP device tracking table. Note Starting from Cisco IOS XE Release 3.10.1E, the following IPDT commands are deprecated; there are no replacement commands: [no] ip device tracking probe count [no] ip device tracking probe delay . For more related information, see the <i>Configuring SISF-Based Device Tracking</i> chapter in this guide.
Step 7	Switch(config)# end	Returns to privileged EXEC mode.
Step 8	Switch# show ip admission configuration	Displays the configuration.

This example shows how to enable web-based authentication on Fast Ethernet port 5/1:

```
Switch(config)# ip admission name webauth1 proxy http
Switch(config)# interface fastethernet 5/1
Switch(config-if)# ip admission webauth1
Switch(config-if)# exit
Switch(config)# ip device tracking
```

This example shows how to verify the configuration:

```
Switch# show ip admission configuration
Authentication Proxy Banner not configured
Authentication global cache time is 60 minutes
Authentication global absolute time is 0 minutes
Authentication global init state time is 2 minutes
Authentication Proxy Watch-list is disabled

Authentication Proxy Rule Configuration
Auth-proxy name webauth1
http list not specified inactivity-time 60 minutes

Authentication Proxy Auditing is disabled
Max Login attempts per user is 5
```

Configuring AAA Authentication

To enable web-based authentication, perform this task:



Note

Beginning with Cisco IOS XE Release 3.11.3aE, the legacy command **tacacs-server** is deprecated. Use the **tacacs server** command if the software running on your device is Cisco IOS XE Release 3.11.3aE or later releases.

	Command	Purpose
Step 1	Switch(config)# aaa new-model	Enables AAA functionality.
	Switch(config)# no aaa new-model	Disables AAA functionality.
Step 2	Switch(config)# aaa authentication login default group {tacacs+ radius}	Defines the list of authentication methods at login.
Step 3	Switch(config)# aaa authorization auth-proxy default group {tacacs+ radius}	Creates an authorization method list for web-based authorization.
	Switch(config)# no aaa authorization auth-proxy default group {tacacs+ radius}	Clears the configured method list.
Step 4	Switch(config)# tacacs server servername	Specifies an AAA server. For RADIUS servers, see the section “ Configuring Switch-to-RADIUS-Server Communication ” section on page 53-9.
Step 5	Switch(config-server-tacacs)# address {ipv4 ipv6} ip-address	Configures the IP address for the TACACS server.
Step 6	Switch(config-server-tacacs)# key key-data	Configures the authorization and encryption key used between the switch and the TACACS server.

This example shows how to enable AAA:

```
Switch(config)# aaa new-model
Switch(config)# aaa authentication login default group tacacs+
Switch(config)# aaa authorization auth-proxy default group tacacs+
```

Configuring Switch-to-RADIUS-Server Communication

RADIUS security servers are identified by one of the following:

- Host name
- Host IP address
- Host name and specific UDP port numbers
- IP address and specific UDP port numbers

The combination of the IP address and UDP port number creates a unique identifier, which enables RADIUS requests to be sent to multiple UDP ports on a server at the same IP address. If two different host entries on the same RADIUS server are configured for the same service (for example, authentication) the second host entry that is configured functions as the failover backup to the first one. The RADIUS host entries are chosen in the order that they were configured.

To configure the RADIUS server parameters, perform this task:

	Command	Purpose
Step 1	Switch(config)# ip radius source-interface <i>interface_name</i>	Specifies that the RADIUS packets have the IP address of the indicated interface.
	Switch(config)# no ip radius source-interface	Prevents the RADIUS packets from having the IP address of the previously indicated interface.
Step 2	Switch(config)# radius server host { <i>hostname</i> <i>ip-address</i> } test username <i>username</i>	Specifies the host name or IP address of the remote RADIUS server. The test username <i>username</i> option enables automated testing of the RADIUS server connection. The specified <i>username</i> does not need to be a valid user name. The key option specifies an authentication and encryption key to be used between the switch and the RADIUS server. To use multiple RADIUS servers, reenter this command.
	Switch(config)# no radius server host { <i>hostname</i> <i>ip-address</i> }	Deletes the specified RADIUS server.
	Switch(config)# radius-server key <i>string</i>	Configures the authorization and encryption key used between the switch and the RADIUS daemon running on the RADIUS server.
Step 4	Switch(config)# radius-server vsa send authentication	Enables downloading of an ACL from the RADIUS server. This feature is supported in Cisco IOS Release 12.2(50)SG.
Step 5	Switch(config)# radius-server dead-criteria tries <i>num-tries</i>	Specifies the number of unanswered transmits to a RADIUS server before considering the server to be inactive. The range of <i>num-tries</i> is 1 to 100.

When you configure the RADIUS server parameters, follow these steps:

- Specify the **key string** on a separate command line.
- For **key string**, specify the authentication and encryption key used between the switch and the RADIUS daemon running on the RADIUS server. The key is a text string that must match the encryption key used on the RADIUS server.
- When you specify the **key string**, use spaces within and at the end of the key. If you use spaces in the key, do not enclose the key in quotation marks unless the quotation marks are part of the key. This key must match the encryption used on the RADIUS daemon.
- You can globally configure the timeout, retransmission, and encryption key values for all RADIUS servers with the **radius server host** global configuration command. If you want to configure these options on a per-server basis, use the **radius-server timeout**, **radius-server retransmit**, and the **radius-server key** global configuration commands. For more information, see the URL:

http://www.cisco.com/en/US/products/ps6586/products_ios_technology_home.html

**Note**

You need to configure some settings on the RADIUS server, including: the IP address of the switch, the key string to be shared by both the server and the switch, and the downloadable ACL (DACL). (Cisco IOS Release 12.2(50)SG supports DACLs.) For more information, see the RADIUS server documentation.

This example shows how to configure the RADIUS server parameters on a switch:

```
Switch(config)# ip radius source-interface Vlan80
Switch(config)# radius server host 172.120.39.46 test username user1
Switch(config)# radius-server key rad123
Switch(config)# radius-server dead-criteria tries 2
```

Configuring the HTTP Server

To use web-based authentication, you must enable the HTTP server within the switch. You can enable the server for either HTTP or HTTPS.

To enable the server, perform one of these tasks:

Command	Purpose
Switch(config)# ip http server	Enables the HTTP server. The web-based authentication feature uses the HTTP server to communicate with the hosts for user authentication.
Switch(config)# ip http secure-server	Enables HTTPS.

Starting with Cisco IOS Release 12.2(50)SG, you can optionally configure custom authentication proxy web pages or specify a redirection URL for successful login, as described in the following sections:

- [Customizing the Authentication Proxy Web Pages, page 53-11](#)
- [Specifying a Redirection URL for Successful Login, page 53-13](#)

Customizing the Authentication Proxy Web Pages

With Cisco IOS Release 12.2(50)SG, you have the option to display four substitute HTML pages to the user in place of the switch's internal default HTML pages during web-based authentication.

To specify the use of your custom authentication proxy web pages, first store your custom HTML files on the switch's internal disk or flash memory, then perform this task in global configuration mode:

	Command	Purpose
Step 1	Switch(config)# ip admission proxy http login page file <i>device:login-filename</i>	Specifies the location in the switch memory file system of the custom HTML file to use in place of the default login page. The <i>device:</i> is either disk or flash memory, such as <i>disk0:</i> .
Step 2	Switch(config)# ip admission proxy http success page file <i>device:success-filename</i>	Specifies the location of the custom HTML file to use in place of the default login success page.

	Command	Purpose
Step 3	Switch(config)# ip admission proxy http failure page file <i>device:fail-filename</i>	Specifies the location of the custom HTML file to use in place of the default login failure page.
Step 4	Switch(config)# ip admission proxy http login expired page file <i>device:expired-filename</i>	Specifies the location of the custom HTML file to use in place of the default login expired page.

When configuring customized authentication proxy web pages, observe the following guidelines:

- To enable the custom web pages feature, specify all four custom HTML files. If you specify fewer than four files, the internal default HTML pages are used.
- The four custom HTML files must be present on the disk or flash of the switch. The maximum size of each HTML file is 8 KB.
- Any images on the custom pages must be located on an accessible HTTP server. An intercept ACL must be configured within the admission rule to allow access to the HTTP server.
- Any external link from a custom page requires configuration of an intercept ACL within the admission rule.
- Any name resolution required for external links or images requires configuration of an intercept ACL within the admission rule to access a valid DNS server.
- If the custom web pages feature is enabled, a configured auth-proxy-banner is not used.
- If the custom web pages feature is enabled, the redirection URL for successful login feature is not available.
- To remove the specification of a custom file, use the **no** form of the command.

Because the custom login page is a public web form, consider the following guidelines for this page:

- The login form must accept user input for the username and password and must POST the data as **uname** and **pwd**.
- The custom login page should follow best practices for a web form, such as page timeout, hidden password, and prevention of redundant submissions.

The following example shows how to configure custom authentication proxy web pages:

```
Switch(config)# ip admission proxy http login page file disk1:login.htm
Switch(config)# ip admission proxy http success page file disk1:success.htm
Switch(config)# ip admission proxy http fail page file disk1:fail.htm
Switch(config)# ip admission proxy http login expired page file disk1:expired.htm
```

The following example shows how to verify the configuration of custom authentication proxy web pages:

```
Switch# show ip admission configuration

Authentication proxy webpage
Login page           : disk1:login.htm
Success page         : disk1:success.htm
Fail Page            : disk1:fail.htm
Login expired Page   : disk1:expired.htm

Authentication global cache time is 60 minutes
Authentication global absolute time is 0 minutes
Authentication global init state time is 2 minutes
Authentication Proxy Session ratelimit is 100
Authentication Proxy Watch-list is disabled
Authentication Proxy Auditing is disabled
Max Login attempts per user is 5
```


Specifying a Redirection URL for Successful Login

With Cisco IOS Release 12.2(50)SG, you have the option to specify a URL to which the user is redirected upon successful authentication, effectively replacing the internal Success HTML page.

To specify a redirection URL for successful login, perform this task:

Command	Purpose
Switch(config)# ip admission proxy http success redirect <i>url-string</i>	Specifies a URL for redirection of the user in place of the default login success page.

When configuring a redirection URL for successful login, consider the following guidelines:

- If the custom authentication proxy web pages feature is enabled, the redirection URL feature is disabled and is not available in the CLI. You can perform redirection in the custom login success page.
- If the redirection URL feature is enabled, a configured auth-proxy-banner is not used.
- To remove the specification of a redirection URL, use the **no** form of the command.

The following example shows how to configure a redirection URL for successful login:

```
Switch(config)# ip admission proxy http success redirect www.cisco.com
```

The following example shows how to verify the redirection URL for successful login:

```
Switch# show ip admission configuration
```

```
Authentication Proxy Banner not configured
Customizable Authentication Proxy webpage not configured
HTTP Authentication success redirect to URL: http://www.cisco.com
Authentication global cache time is 60 minutes
Authentication global absolute time is 0 minutes
Authentication global init state time is 2 minutes
Authentication Proxy Watch-list is disabled
Authentication Proxy Max HTTP process is 7
Authentication Proxy Auditing is disabled
Max Login attempts per user is 5
```

Configuring the Web-Based Authentication Parameters

You can configure the maximum number of failed login attempts allowed before the client is placed in a watch-list for a waiting period.

To configure the web-based authentication parameters, perform this task:

	Command	Purpose
Step 1	Switch(config)# ip admission max-login-attempts <i>number</i>	Sets the maximum number of failed login attempts. The default is 5. Note A typical custom setting for this value should not exceed 50.
Step 2	Switch(config)# end	Returns to privileged EXEC mode.

	Command	Purpose
Step 3	Switch# show ip admission configuration	Displays the authentication proxy configuration.
Step 4	Switch# show ip admission cache	Displays the list of authentication entries.

This example shows how to set the maximum number of failed login attempts to 10:

```
Switch(config)# ip admission max-login-attempts 10
```

Removing Web-Based Authentication Cache Entries

To delete existing session entries, perform either of these tasks:

Command	Purpose
Switch# clear ip auth-proxy cache [* <i>host ip address</i>]	Deletes authentication proxy entries. Use an asterisk to delete all cache entries. Enter a specific IP address to delete the entry for a single host.
Switch# clear ip admission cache [* <i>host ip address</i>]	Deletes authentication proxy entries. Use an asterisk to delete all cache entries. Enter a specific IP address to delete the entry for a single host.

This example shows how to remove the web-based authentication session for the client at IP address 209.165.201.1:

```
Switch# clear ip auth-proxy cache 209.165.201.1
```

Displaying Web-Based Authentication Status

To display the web-based authentication settings for all interfaces or for specific ports, perform this task:

Command	Purpose
Switch# show authentication sessions [interface <i>type slot/port</i>]	Displays the web-based authentication settings. type = fastethernet, gigabitethernet, or tengigabitethernet (Optional) Use the interface keyword to display the web-based authentication settings for a specific interface.

This example shows how to view only the global web-based authentication status:

```
Switch# show authentication sessions
```

This example shows how to view the web-based authentication settings for interface Gi 3/27:

```
Switch# show authentication sessions interface gigabitethernet 3/27
```



Auto Identity

The Auto Identity feature provides a set of built-in policies at global configuration and interface configuration modes. This feature is available only in Class-Based Policy Language (CPL) control policy-equivalent new-style mode. To convert all the relevant authentication commands to their CPL control policy-equivalents, use the authentication convert-to new-style command.

This module describes the feature and consists of these sections:

- [Information About Auto Identity, page 54-1](#)
- [How to Configure Auto Identity, page 54-5](#)
- [Configuration Examples for Auto Identity, page 54-6](#)
- [Verifying Auto Identity, page 54-7](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

Information About Auto Identity

- [Auto Identity Overview, page 54-2](#)
- [Auto Identity Global Template, page 54-2](#)
- [Auto Identity Interface Templates, page 54-3](#)
- [Auto Identity Built-in Policies, page 54-4](#)
- [Auto Identity Class Map Templates, page 54-4](#)
- [Auto Identity Parameter Maps, page 54-5](#)
- [Auto Identity Service Templates, page 54-5](#)

Auto Identity Overview

The Cisco Identity-Based Networking Services (IBNS) solution provides a policy and identity-based framework in which edge devices can deliver flexible and scalable services to subscribers. IBNS allows the concurrent operation of IEEE 802.1x (dot1x), MAC authentication bypass (MAB), and web authentication methods, making it possible to invoke multiple authentication methods in parallel, on a single subscriber session. These authentication methods, dot1x, authentication, authorization, and accounting (AAA), and RADIUS are available in global configuration and interface configuration modes.

The Auto Identity feature uses the Cisco Common Classification Policy Language-based configuration that significantly reduces the number of commands used to configure both authentication methods and interface-level commands. The Auto Identity feature provides a set of built-in policies that are based on policy maps, class maps, parameter maps, and interface templates.

In global configuration mode, the **source template AI_GLOBAL_CONFIG_TEMPLATE** command enables the Auto Identity feature. In interface configuration mode, configure the **AI_MONITOR_MODE**, **AI_LOW_IMPACT_MODE**, or **AI_CLOSED_MODE** interface templates to enable the feature on interfaces.

You can configure multiple templates; however, you must bind multiple templates together using the **merge** command. If you do not bind the templates, the last configured template is used. While binding templates, if the same command is repeated in two templates with different arguments, the last configured command is used.

**Note**

You can also enable user-defined templates that are configured using the **template name** command in global configuration mode.

Use the **show template interface** or **show template global** commands to display information about built-in templates.

Built-in templates can be edited. Built-in template information is displayed in the output of the **show running-config** command, if the template is edited. If you delete an edited built-in template, the built-in template reverts to the default and is not deleted from the configuration. However, if you delete a user-defined template, it is deleted from the configuration.

**Note**

Before you delete a template, ensure that it is not attached to a device.

Auto Identity Global Template

To enable the global template, configure the **source template template-name** command in global configuration mode.

**Note**

You must configure the RADIUS server commands, because these are not automatically configured when the global template is enabled.

The following example shows how to enable the global template:

```
Switch(config)# source template AI_GLOBAL_CONFIG_TEMPLATE
Switch(config)# radius server ISE
Switch(config-radius-server)# address ipv4 172.20.254.4 auth-port 1645 acct-port 1646
Switch(config-radius-server)# key cisco
```

```
Switch(config-radius-server)# end
```

The AI_GLOBAL_CONFIG_TEMPLATE automatically configures the following commands:

```
dot1x system-auth-control
aaa new-model
aaa authentication dot1x default group radius
aaa authorization network default group radius
aaa authorization auth-proxy default group radius
aaa accounting identity default start-stop group radius
aaa accounting system default start-stop group radius
radius-server attribute 6 on-for-login-auth
radius-server attribute 6 support-multiple
radius-server attribute 6 voice 1
radius-server attribute 8 include-in-access-req
radius-server attribute 25 access-request include
```

Auto Identity Interface Templates

The following interface templates are available in the Auto Identity feature:

- AI_MONITOR_MODE—Passively monitors sessions that have authentication in open mode.
- AI_LOW_IMPACT_MODE—Similar to monitor mode, but with a configured static policy such as a port access control list (PACL).
- AI_CLOSED_MODE—Secure mode in which data traffic is not allowed into the network, until authentication is complete. This mode is the default.

The following commands are inbuilt in the AI_MONITOR_MODE:

```
switchport mode access
access-session port-control auto
access-session host-mode multi-auth
dot1x pae authenticator
mab
service-policy type control subscriber AI_DOT1X_MAB_POLICIES
```

The following commands are inbuilt in the AI_LOW_IMPACT_MODE:

```
switchport mode access
access-session port-control auto
access-session host-mode multi-auth
dot1x pae authenticator
mab
ip access-group AI_PORT_ACL in
service-policy type control subscriber AI_DOT1X_MAB_POLICIES
```

The following commands are inbuilt in the AI_CLOSED_MODE:

```
switchport mode access
access-session closed
access-session port-control auto
access-session host-mode multi-auth
dot1x pae authenticator
mab
service-policy type control subscriber AI_DOT1X_MAB_POLICIES
```

Auto Identity Built-in Policies

The following five built-in policies are available in the Auto Identity feature:

- **AI_DOT1X_MAB_AUTH**—Enables flexible authentication with dot1x, and then MAC Address Bypass (MAB).
- **AI_DOT1X_MAB_POLICIES**—Enables flexible authentication with dot1x, and then MAB. Applies critical VLAN in case the Authentication, Authorization, and Accounting (AAA) server is not reachable.
- **AI_DOT1X_MAB_WEBAUTH**—Enables flexible authentication with dot1x, MAB, and then web authentication.
- **AI_NEXTGEN_AUTHBYBASS**—Skips authentication if an IP phone device is detected. Enables the **device classifier** command in global configuration mode and the **voice-vlan** command in interface configuration mode to detect the device. This is a reference policy map, and users can copy the contents of this policy map to other policy maps.
- **AI_STANDALONE_WEBAUTH**—Defines standalone web authentication.

Auto Identity Class Map Templates

The following built-in class maps are supported by the Auto Identity feature:

- **AI_NRH**—Specifies that the nonresponsive host (NRH) authentication method is enabled.
- **AI_WEBAUTH_METHOD**—Specifies that the web authentication method is enabled.
- **AI_WEBAUTH_FAILED**—Specifies that the web authentication method failed to authenticate.
- **AI_WEBAUTH_NO_RESP**—Specifies that the web authentication client failed to respond.
- **AI_DOT1X_METHOD**—Specifies that the dot1x method is enabled.
- **AI_DOT1X_FAILED**—Specifies that the dot1x method failed to authenticate.
- **AI_DOT1X_NO_RESP**—Specifies that the dot1x client failed to respond.
- **AI_DOT1X_TIMEOUT**—Specifies that the dot1x client stopped responding after the initial acknowledge (ACK) request.
- **AI_MAB_METHOD**—Specifies that the MAC Authentication Bypass (MAB) method is enabled.
- **AI_MAB_FAILED**—Specifies that the MAB method failed to authenticate.
- **AI_AAA_SVR_DOWN_AUTHD_HOST**—Specifies that the Authentication, Authorization, and Accounting (AAA) server is down, and the client is in authorized state.
- **AI_AAA_SVR_DOWN_UNAUTHD_HOST**—Specifies that the AAA server is down, and the client is in unauthorized state.
- **AI_IN_CRITICAL_AUTH**—Specifies that the critical authentication service template is applied.
- **AI_NOT_IN_CRITICAL_AUTH**—Specifies that the critical authentication service template is not applied.
- **AI_METHOD_DOT1X_DEVICE_PHONE**—Specifies that the method is dot1x and the device type is IP phone.
- **AI_DEVICE_PHONE**—Specifies that the device type is IP phone.

Auto Identity Parameter Maps

The following built-in parameter map templates are supported by the Auto Identity feature:

- **AI_NRH_PMAP**—Starts nonresponsive host (NRH) authentication.
- **AI_WEBAUTH_PMAP**—Starts web authentication.

Auto Identity Service Templates

Service templates are available inside built-in policy maps. The following built-in service templates are supported by the Auto Identity feature:


- **AI_INACTIVE_TIMER**—Template to start the inactivity timer.
- **AI_CRITICAL_ACL**—Dummy template; users can configure this template as per their requirements.


How to Configure Auto Identity

- [Configuring Auto Identity Globally, page 54-5](#)
- [Configuring Auto Identity at an Interface Level, page 54-6](#)

Configuring Auto Identity Globally

To configure Auto Identity globally, perform this task:

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	Switch# configure terminal	Enters global configuration mode.
Step 3	Switch(config)# source template { AI_GLOBAL_CONFIG_TEMPLATE <i>template-name</i> }	Configures an auto identity template. <ul style="list-style-type: none"> • AI_GLOBAL_CONFIG_TEMPLATE is a built-in template. • The <i>template-name</i> argument is a user-defined template.
Step 4	Switch(config)# aaa new-model	Enables the authentication, authorization, and accounting (AAA) access control mode.
Step 5	Switch(config)# radius server name	Specifies the name for the RADIUS server configuration for Protected Access Credential (PAC) provisioning and enters RADIUS server configuration mode.
Step 6	Switch(config-radius-server)# address ipv4 { <i>hostname</i> <i>ipv4-address</i> }	Configures the IPv4 address for the RADIUS server accounting and authentication parameters. <div>  <p>Note This command is not a part of the global template, and you must configure it.</p> </div>

	Command or Action	Purpose
Step 7	Switch(config-radius-server)# key ipv4 {0 string 7 string} string	Specifies the authentication and encryption key for all RADIUS communications between the device and the RADIUS server.
		 Note This command is not a part of the global template, and you must configure it.
Step 8	Switch(config-radius-server)# end	Exits RADIUS server configuration mode and returns to privileged EXEC mode.

Configuring Auto Identity at an Interface Level

When you configure two interface templates, you must configure the **merge** keyword. If you do not, the last configured template is used.

	Command or Action	Purpose
Step 1	Switch# enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	Switch# configure terminal	Enters global configuration mode.
Step 3	Switch(config)# interface type number	Configures an interface and enters interface configuration mode.
Step 4	Switch(config-if)# source template {AI_CLOSED_MODE AI_LOW_IMPACT_MODE AI_MONITOR_MODE template-name} [merge]	Configures a source template for the interface.
Step 5	Switch(config-if)# source template {AI_CLOSED_MODE AI_LOW_IMPACT_MODE AI_MONITOR_MODE template-name} [merge]	(Optional) Configures a source template for the interface and merges this template with the previously configured template. <ul style="list-style-type: none"> When you configure two templates, if you do not configure the merge keyword, the last configured template is used.
Step 6	Switch(config-if)# switchport access vlan vlan-id	Sets the VLAN when the interface is in access mode.
Step 7	Switch(config-if)# switchport voice vlan vlan-id	Configures a voice VLAN on a multiple VLAN access port.
Step 8	Repeat Steps 4, 6, and 7 on all interfaces that must have the Auto Identity feature configured.	—
Step 9	Switch(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for Auto Identity

- [Example: Configuring Auto Identity Globally, page 54-7](#)
- [Example: Configuring Auto Identity at an Interface Level, page 54-7](#)

Example: Configuring Auto Identity Globally

```
Switch> enable
Switch# configure terminal
Switch(config)# source template AI_GLOBAL_CONFIG_TEMPLATE
Switch(config)# aaa new-model
Switch(config)# radius server ISE
Switch(config-radius-server)# address ipv4 10.1.1.1
Switch(config-radius-server)# key ipv4 cisco
Switch(config-radius-server)# end
```

Example: Configuring Auto Identity at an Interface Level

```
Switch> enable
Switch# configure terminal
Switch(config)# interface gigabitethernet 1/0/1
Switch(config-if)# source template AI_CLOSED_MODE
Switch(config-if)# source template AI_MONITOR_MODE merge
Switch(config-if)# switchport access vlan 100
Switch(config-if)# switchport voice vlan 101
Switch(config-if)# end
```

Verifying Auto Identity

To verify the Auto Identity configuration, use the following commands:

The following output from the **show template interface source built-in all** command displays all the configured built-in interface templates:

```
Switch# show template interface source built-in all
```

```
Template Name      : AI_CLOSED_MODE
Modified           : No
Template Definition :
  dot1x pae authenticator
  switchport mode access
  mab
  access-session closed
  access-session port-control auto
  service-policy type control subscriber AI_DOT1X_MAB_POLICIES
!
```

```
Template Name      : AI_LOW_IMPACT_MODE
Modified           : No
Template Definition :
  dot1x pae authenticator
  switchport mode access
  mab
  access-session port-control auto
  service-policy type control subscriber AI_DOT1X_MAB_POLICIES
  ip access-group AI_PORT_ACL in
!
```

```
Template Name      : AI_MONITOR_MODE
Modified           : No
Template Definition :
  dot1x pae authenticator
  switchport mode access
  mab
```

```

access-session port-control auto
service-policy type control subscriber AI_DOT1X_MAB_POLICIES
!

```

The following output from the **show template global source built-in all** command displays all the configured global built-in templates:

```

Switch# show template global source built-in all

Global Template Name      : AI_GLOBAL_CONFIG_TEMPLATE
Modified                  : No
Global Template Definition : global
dot1x system-auth-control
aaa new-model
aaa authentication dot1x default group radius
aaa authorization network default group radius
aaa authorization auth-proxy default group radius
aaa accounting identity default start-stop group radius
aaa accounting system default start-stop group radius
radius-server attribute 6 on-for-login-auth
radius-server attribute 6 support-multiple
radius-server attribute 6 voice 1
radius-server attribute 8 include-in-access-req
radius-server attribute 25 access-request include
!

```

The following output from the **show derived-config | include aaa | radius-server** command displays the composite results of all the configuration commands that apply to an interface, including commands that come from sources such as static templates, dynamic templates, dialer interfaces, and authentication, authorization, and accounting (AAA) per-user attributes:

```

Switch# show derived-config | include aaa | radius-server

aaa new-model
aaa authentication dot1x default group radius
aaa authorization network default group radius
aaa authorization auth-proxy default group radius
aaa accounting identity default start-stop group radius
aaa accounting system default start-stop group radius
aaa session-id common
radius-server attribute 6 on-for-login-auth
radius-server attribute 6 support-multiple
radius-server attribute 6 voice 1
radius-server attribute 8 include-in-access-req
radius-server attribute 25 access-request include
radius server host 10.25.18.42 key cisco

```

The following output from the **show derived-config | interface type-number** command displays the composite results of all configuration for an interface:

```

Switch# show derived-config | interface gigabitethernet2/0/6

Building configuration...
Derived configuration : 267 bytes
!
interface GigabitEthernet2/0/6
 switchport mode access
 switchport voice vlan 100
 access-session closed
 access-session port-control auto
 mab

```

```
dot1x pae authenticator
spanning-tree portfast edge
service-policy type control subscriber AI_DOT1X_MAB_POLICIES
end
```

The following output from the **show access-session | interface *interface-type-number*** details command displays the policies applied to an interface:

```
Switch# show access-session interface gigabitethernet2/0/6 details
```

```
Interface          : GigabitEthernet2/0/6
                   MAC Address: c025.5c43.be00
                   IPv6 Address: Unknown
                   IPv4 Address: Unknown
                   User-Name: CP-9971-SEPC0255C43BE00
                   Device-type: Cisco-IP-Phone-9971
                   Status: Authorized
                   Domain: VOICE
                   Oper host mode: multi-auth
                   Oper control dir: both
                   Session timeout: N/A
                   Common Session ID: 091A1C5B00000017002003EE
                   Acct Session ID: 0x00000005
                   Handle: 0xBB00000B
                   Current Policy: AI_DOT1X_MAB_POLICIES
```

Local Policies:

Server Policies:

```
Vlan Group: Vlan: 100
Security Policy: Must Not Secure
Security Status: Link Unsecure
```

Method status list:

```
Method  State
dot1x   Authc Success
```

The following output from the **show running-config interface *type-number*** command displays the contents of the current running configuration file or the configuration for an interface:

```
Switch# show running-config interface gigabitethernet2/0/6
```

```
Building configuration...
Current configuration : 214 bytes
!
interface GigabitEthernet2/0/6
 switchport mode access
 switchport voice vlan 100
 access-session port-control auto
 spanning-tree portfast edge
 service-policy type control subscriber AI_NEXTGEN_AUTHBYPASS
end
```

The following output from the **show lldp neighbor** command displays information about one or all neighboring devices discovered using the Link Layer Discovery Protocol (LLDP):

```
Switch# show lldp neighbor
```

Capability codes:

```
(R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
(W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other
```

Device ID	Local Intf	Hold-time	Capability	Port ID
SEPC0255C43BE00	Gi2/0/6	180	B,T	C0255C43BE00:P1

Total entries displayed: 1



Configuring Port Security

This chapter describes how to configure port security on the Catalyst 4500 series switch. It provides an overview of port security on the Catalyst 4500 series switch and details the configuration on various types of ports such as access, voice, trunk, and private VLAN (PVLAN).

This chapter consists of these sections:

- [Port Security Commands, page 55-1](#)
- [About Port Security, page 55-3](#)
- [Configuring Port Security on Access Ports, page 55-7](#)
- [Configuring Port Security on PVLAN Ports, page 55-14](#)
- [Configuring Port Security on Trunk Ports, page 55-17](#)
- [Configuring Port Security on Voice Ports, page 55-22](#)
- [Displaying Port Security Settings, page 55-27](#)
- [Configuring Port Security with Other Features/Environments, page 55-31](#)
- [Port Security Configuration Guidelines and Restrictions, page 55-33](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

Port Security Commands

This table lists the commands most commonly used with port security.

Command	Purpose	Navigation
errdisable recovery cause psecure-violation	Brings a secure port out of error-disabled state.	Violation Actions, page 55-6
errdisable recovery interval	Customizes the time to recover from a specified error disable cause.	Violation Actions, page 55-6

Command	Purpose	Navigation
port-security mac-address	Configures all secure MAC addresses on each VLAN.	Secure MAC Addresses, page 55-3
port-security maximum	Configures a maximum number of MAC addresses on an interface.	Configuring Port Security on Access Ports, page 55-7
private-vlan association add	Creates an association between a secondary VLAN and a primary VLAN.	Example of Port Security on an Isolated Private VLAN Host Port, page 55-16
private-vlan isolated	Designates the VLAN as a private VLAN.	Configuring Port Security on an Isolated Private VLAN Host Port, page 55-14
private-vlan primary	Specifies the VLAN as the primary private VLAN.	Configuring Port Security on an Isolated Private VLAN Host Port, page 55-14
switchport mode private-vlan host	Specifies that ports with valid private VLAN trunk association become active host private VLAN trunk ports.	Configuring Port Security on an Isolated Private VLAN Host Port, page 55-14
switchport private-vlan host-association	Defines a host association on an isolated host port.	Configuring Port Security on an Isolated Private VLAN Host Port, page 55-14
switchport private-vlan mapping	Defines a private VLAN for the promiscuous ports.	Configuring Port Security on an Isolated Private VLAN Host Port, page 55-14
switchport port-security	Enables port security.	Configuring Port Security on Access Ports, page 55-7
switchport port-security aging static	Configures static aging of MAC address.	Aging Secure MAC Addresses, page 55-5
switchport port-security aging time	Specifies an aging time for a port.	Example 3: Setting the Aging Timer, page 55-11
switchport port-security limit rate invalid-source-mac	Sets the rate limit for bad packets.	Example 7: Setting a Rate Limit for Bad Packets, page 55-13
switchport port-security mac-address	Configures a secure MAC address for an interface.	Example 5: Configuring a Secure MAC Address, page 55-12
switchport port-security mac-address <i>mac_address</i> sticky	Specifies the sticky MAC address for an interface.	Configuring Port Security on Access Ports, page 55-7
switchport port-security mac-address sticky	Enables sticky Port Security.	Sticky Addresses on a Port, page 55-5
no switchport port-security mac-address sticky	Converts a sticky secure MAC address to a dynamic MAC secure address.	Configuring Port Security on Access Ports, page 55-7
switchport port-security maximum	Sets the maximum number of secure MAC addresses for an interface.	Example 1: Setting Maximum Number of Secure Addresses, page 55-11
switchport port-security violation	Sets the violation mode.	Example 2: Setting a Violation Mode, page 55-11

Command	Purpose	Navigation
no switchport port-security violation	Sets the violation mode.	Configuring Port Security on Access Ports, page 55-7
switchport trunk encapsulation dot1q	Sets the encapsulation mode to dot1q.	Example 1: Configuring a Maximum Limit of Secure MAC Addresses for All VLANs, page 55-19

About Port Security

Port security enables you to restrict the number of MAC addresses (termed *secure MAC addresses*) on a port, allowing you to prevent access by unauthorized MAC addresses. It also allows you to configure a maximum number of secure MAC addresses on a given port (and optionally for a VLAN for trunk ports). When a secure port exceeds the maximum, a security violation is triggered, and a violation action is performed based on the violation action mode configured on the port.

If you configure the maximum number of secure MAC addresses as 1 on the port, the device attached to the secure port is assured sole access to the port.

If a secure MAC address is secured on a port, that MAC address is not allowed to enter on any other port off that VLAN. If it does, the packet is dropped unnoticed in the hardware. Other than using the interface or port counters, you do not receive a log message reflecting this fact. Be aware that this condition does not trigger a violation. Dropping these packets in the hardware is more efficient and can be done without putting additional load on the CPU.

Port security has the following characteristics:

- It allows you to age out secure MAC addresses. Two types of aging are supported: inactivity and absolute.
- It supports a sticky feature whereby the secure MAC addresses on a port are retained through switch reboots and link flaps.
- It can be configured on various types of ports such as access, voice, trunk, EtherChannel, and private VLAN ports.

This overview contains the following topics:

- [Secure MAC Addresses, page 55-3](#)
- [Maximum Number of Secure MAC Addresses, page 55-4](#)
- [Aging Secure MAC Addresses, page 55-5](#)
- [Sticky Addresses on a Port, page 55-5](#)
- [Violation Actions, page 55-6](#)

Secure MAC Addresses

Port security supports the following types of secure MAC addresses:

- **Dynamic or Learned**—Dynamic secure MAC addresses are learned when packets are received from the host on the secure port. You might want to use this type if the user's MAC address is not fixed (laptop).
- **Static or configured**—Static secure MAC addresses are configured by the user through CLI or SNMP. You might want to use this type if your MAC address remains fixed (PC).

- **Sticky**—Sticky addresses are learned such as dynamic secure MAC addresses, but persist through switch reboots and link flaps such as static secure MAC addresses. You might want to use this type if a large number of fixed MAC addresses exist and you do not want to configure MAC addresses manually (100 PCs secured on their own ports).

If a port has reached its maximum number of secure MAC addresses and you try to configure a static secure MAC address, your configuration is rejected and an error message displays. If a port has reached its maximum number of secure MAC addresses and a new dynamic secure MAC address is added, a violation action is triggered.

You can clear dynamic secure MAC addresses with the **clear port-security** command. You can clear sticky and static secure MAC addresses one at a time with the **no** form of the **switchport port-security mac-address** command.

Maximum Number of Secure MAC Addresses

A secure port has a default of one MAC address. You can change the default to any value between 1 and 3,000. The upper limit of 3,000 guarantees one MAC address per-port and an additional 3,000 across all ports in the system.

After you have set the maximum number of secure MAC addresses on a port, you can include the secure addresses in an address table in one of the following ways:

- You can configure the secure MAC addresses with the **switchport port-security mac-address mac_address** interface configuration command.
- You can configure all secure MAC addresses on a range of VLANs with the **port-security mac-address** VLAN range configuration command for trunk ports.
- You can allow the port to dynamically configure secure MAC addresses with the MAC addresses of connected devices.
- You can configure some of the addresses and allow the rest to be dynamically configured.



Note

If a port's link goes down, all dynamically secured addresses on that port are no longer secure.

- You can configure MAC addresses to be sticky. These can be dynamically learned or manually configured, stored in the address table, and added to the running configuration. After these addresses are saved in the configuration file, the interface does not need to dynamically relearn them when the switch restarts. Although you can manually configure sticky secure addresses, this action is not recommended.



Note

On a trunk port, a maximum number of secure MAC addresses can be configured on both the port and port VLAN. The port's maximum value can be greater than or equal to the port VLAN maximum(s) but not less than the port VLAN maximum(s). If the port's maximum value is less than at least one of the port VLAN's maximum (for example, if we have max set to 3 on VLAN 10 while no "sw port max" is set (defaults to 1)), the port shuts down when dynamic adds reaches 2 on VLAN 10 (see "Port Security Configuration Guidelines and Restrictions" on page 33). The port VLAN maximum enforces the maximum allowed on a given port on a given VLAN. If the maximum is exceeded on a given VLAN but the port's maximum is not exceeded, the port still shuts down. The entire port is shut down even if one of the VLANs on the port has actually caused the violation.

Aging Secure MAC Addresses

You might want to age secure MAC addresses when the switch may be receiving more than 3,000 MAC addresses ingress.

**Note**

Aging of sticky addresses is not supported.

By default, port security does not age out the secure MAC addresses. After learned, the MAC addresses remain on the port until either the switch reboots or the link goes down (unless the sticky feature is enabled). However, port security does allow you to configure aging based on the absolute or inactivity mode and aging interval (in minutes, from 1 to n).

- Absolute mode—Ages between n and n+1
- Inactivity mode—Ages between n+1 and n+2

Use this feature to remove and add PCs on a secure port without manually deleting the existing secure MAC addresses, while still limiting the number of secure addresses on a port.

Unless static aging is explicitly configured with the **switchport port-security aging static** command, static addresses are not aged even if aging is configured on the port.

**Note**

The aging increment is one minute.

Sticky Addresses on a Port

By enabling sticky port security, you can configure an interface to convert the dynamic MAC addresses to sticky secure MAC addresses and to add them to the running configuration. You might want to do this if you do not expect the user to move to another port, and you want to avoid statically configuring a MAC address on every port.

**Note**

If you use a different chassis, you might need another MAC address.

To enable sticky port security, enter the **switchport port-security mac-address sticky** command. When you enter this command, the interface converts all the dynamic secure MAC addresses, including those that were dynamically learned before sticky learning was enabled, to sticky secure MAC addresses.

The sticky secure MAC addresses do not automatically become part of the configuration file, which is the startup configuration used each time the switch restarts. If you save the running config file to the configuration file, the interface does not need to relearn these addresses when the switch restarts. If you do not save the configuration, they are lost.

If sticky port security is disabled, the sticky secure MAC addresses are converted to dynamic secure addresses and are removed from the running configuration.

After the maximum number of secure MAC addresses is configured, they are stored in an address table. To ensure that an attached device has sole access of the port, configure the MAC address of the attached device and set the maximum number of addresses to one, which is the default.

A security violation occurs if the maximum number of secure MAC addresses to a port has been added to the address table and a workstation whose MAC address is not in the address table attempts to access the interface.

Forbidden MAC Addresses

You can prevent the switch from learning specific MAC addresses, by forbidding the MAC addresses on all interfaces, globally, or on a specific port-security enabled interface.

Violation Actions

A security violation is triggered in these situations:

- When the number of secure MAC addresses on the port exceeds the maximum number of secure MAC addresses allowed on the port.

**Note**

A secure violation is not triggered if the host secured on one port shows up on another port. The Catalyst 4500 series switch drops such packets on the new port silently in the hardware and does not overload the CPU.

- Running diagnostic tests with port security enabled.

You can configure the interface for one of following violation modes, which are based on the response to the violation:

- Restrict—A port security violation restricts data (that is, packets are dropped in software), causes the SecurityViolation counter to increment, and causes an SNMP Notification to be generated. You might want to configure this mode in order to provide uninterrupted service/access on a secure port.

The rate at which SNMP traps are generated can be controlled by the **snmp-server enable traps port-security trap-rate** command. The default value (“0”) causes an SNMP trap to be generated for every security violation.

- Shutdown—A port security violation causes the interface to shut down immediately. You might want to configure this mode in a highly secure environment, where you do not want unsecured MAC addresses to be denied in software and service interruption is not an issue.
- Shutdown VLAN—Use to set the security violation mode for each VLAN. In this mode, the offending VLAN is error disabled instead of the entire port when a violation occurs.

When a secure port is in the error-disabled state, you can bring it out of this state automatically by configuring the **errdisable recovery cause psecure-violation** global configuration command or you can manually reen able it by entering the **shutdown** and **no shut down** interface configuration commands. it is the default mode. If a port is in per-VLAN errdisable mode, you can also use **clear errdisable interface name vlan range** command to reen able the VLAN on the port.

You can also customize the time to recover from the specified error disable cause (default is 300 seconds) by entering the **errdisable recovery interval interval** command.

Invalid Packet Handling

You might want to rate limit invalid source MAC address packets on a secure port if you anticipate that a device will send invalid packets (such as traffic generator, sniffer, and bad NICs).

The port security feature considers the following as “invalid frames”:

- Packets with a source or destination MAC address that is all zero
- Packets with a multicast or broadcast source MAC address
- Packets from an address either learned or configured on a secure interface that are observed on another secure interface in the same VLAN

You can chose to rate limit these packets. If the rate is exceeded, you can trigger a violation action for the port.

Configuring Port Security on Access Ports

These sections describe how to configure port security:

- [Configuring Port Security on Access Ports, page 55-7](#)
- [Examples of Port Security on Access Ports, page 55-10](#)



Note

Port security can be enabled on a Layer 2 port channel interface configured in access mode. The port security configuration on an EtherChannel is independent of the configuration of any member ports.

Configuring Port Security on Access Ports

To restrict traffic through a port by limiting and identifying MAC addresses of the stations allowed to the port, perform this task:

	Command	Purpose
Step 1	Switch(config)# interface <i>interface_id</i> interface <i>port-channel port_channel_number</i>	Enters interface configuration mode and specifies the interface to configure. Note The interface can be a Layer 2 port channel logical interface.
Step 2	Switch(config-if)# switchport mode access	Sets the interface mode. Note An interface in the default mode (dynamic auto) cannot be configured as a secure port.
Step 3	Switch(config-if)# [no] switchport port-security	Enables port security on the interface. To return the interface to the default condition as a not secured, use the no switchport port-security command.
Step 4	Switch(config-if)# [no] switchport port-security maximum <i>value</i>	(Optional) Sets the maximum number of secure MAC addresses for the interface. The range is 1 to 3072; the default is 1. To return the interface to the default number of secure MAC addresses, use the no switchport port-security maximum <i>value</i> .

Command	Purpose
Step 5 Switch(config-if)# switchport port-security mac-address forbidden OR Switch(config)# port-security mac-address forbidden	(Optional) Sets the MAC address forbidden on the interface. OR Optional) Sets the MAC address forbidden on all interfaces, globally. To verify the MAC addresses forbidden on the interface, use the show port-security address forbidden command, in privileged EXEC mode.
Step 6 Switch(config-if)# switchport port-security [aging {static time aging_time type {absolute inactivity}}]	Sets the aging time and aging type for all secure addresses on a port. Use this feature to remove and add PCs on a secure port without manually deleting the existing secure MAC addresses while still limiting the number of secure addresses on a port. The static keyword enables aging for statically configured secure addresses on this port. The time aging_time value specifies the aging time for this port. Valid range for <i>aging_time</i> is from 0 to 1440 minutes. If the time is equal to 0, aging is disabled for this port. The type keyword sets the aging type as absolute or inactive . <ul style="list-style-type: none"> • absolute—All the secure addresses on this port age out exactly after the time (minutes) specified and are removed from the secure address list. • inactive—The secure addresses on this port age out only if there is no data traffic from the secure source address for the specified time period. To disable port security aging for all secure addresses on a port, use the no switchport port-security aging time interface configuration command.

	Command	Purpose
Step 7	Switch(config-if)# [no] switchport port-security violation {restrict shutdown shutdown vlan}	<p>(Optional) Sets the violation mode, the action to be taken when a security violation is detected, as one of these:</p> <ul style="list-style-type: none"> • restrict—A port security violation restricts data and causes the SecurityViolation counter to increment and send an SNMP trap notification. • shutdown—The interface is error-disabled when a security violation occurs. • shutdown vlan—Use to set the security violation mode for each VLAN. In this mode, the VLAN is error-disabled instead of the entire port when a violation occurs. <p>Note When a secure port is in the error-disabled state, you can bring it out of this state by entering the errdisable recovery cause psecure-violation global configuration command or you can manually reenable it by entering the shutdown and no shut down interface configuration commands.</p> <p>To return the violation mode to the default condition (shutdown mode), use the no switchport port-security violation shutdown command.</p>
Step 8	Switch(config-if)# switchport port-security limit rate invalid-source-mac <i>packets_per_sec</i>	<p>Sets the rate limit for bad packets.</p> <p>Default is 10 pps.</p>
Step 9	Switch(config-if)# [no] switchport port-security mac-address <i>mac_address</i>	<p>(Optional) Enters a secure MAC address for the interface. You can use this command to configure a secure MAC addresses. If you configure fewer secure MAC addresses than the maximum, the remaining MAC addresses are dynamically learned.</p> <p>To delete a MAC address from the address table, use the no switchport port-security mac-address <i>mac_address</i> command.</p> <p>Note This command only applies to access, PVLAN host, and PVLAN promiscuous mode. For more details on PVLAN, trunk, or regular trunk mode, refer to the “Configuring Port Security on Trunk Ports” section on page 55-17.</p>
Step 10	Switch(config-if)# [no] switchport port-security mac-address sticky	<p>(Optional) Enables sticky learning on the interface.</p> <p>To disable sticky learning on an interface, use the no switchport port-security mac-address sticky command. The interface converts the sticky secure MAC addresses to dynamic secure addresses.</p>

Command	Purpose
Step 11 Switch(config-if)# [no] switchport port-security mac-address <i>mac_address</i> sticky [vlan [voice access]]	<p>Specifies the sticky mac-address for the interface.</p> <p>When you specify the vlan keyword, the mac-address becomes sticky in the specified VLAN.</p> <p>To delete a sticky secure MAC addresses from the address table, use the no switchport port-security mac-address <i>mac_address</i> sticky command. To convert sticky to dynamic addresses, use the no switchport port-security mac-address sticky command.</p> <p>Note This command only applies to access, PVLAN host, and PVLAN promiscuous mode. For more details on PVLAN or trunk or regular trunk mode, refer to the “Configuring Port Security on Trunk Ports” section on page 55-17.</p>
Step 12 Switch(config-if)# end	Returns to privileged EXEC mode.
Step 13 Switch# show port-security address Switch# show port-security address	Verifies your entries.

**Note**

To clear dynamically learned port security MAC addresses in the CAM table, use the clear port-security dynamic command. The address keyword enables you to clear a secure MAC addresses. The interface keyword enables you to clear all secure addresses on any interface (including any port channel interface). The VLAN keyword allows you to clear port security MACs on a per-VLAN per-port basis.

Examples of Port Security on Access Ports

The following examples are provided:

- [Example 1: Setting Maximum Number of Secure Addresses, page 55-11](#)
- [Example 2: Setting a Violation Mode, page 55-11](#)
- [Example 3: Setting the Aging Timer, page 55-11](#)
- [Example 4: Setting the Aging Timer Type, page 55-12](#)
- [Example 5: Configuring a Secure MAC Address, page 55-12](#)
- [Example 6: Configuring Sticky Port Security, page 55-13](#)
- [Example 7: Setting a Rate Limit for Bad Packets, page 55-13](#)
- [Example 8: Clearing Dynamic Secure MAC Addresses, page 55-14](#)

Example 1: Setting Maximum Number of Secure Addresses

This example shows how to enable port security on the Fast Ethernet interface 3/12 and how to set the maximum number of secure addresses to 5. The violation mode is the default, and no secure MAC addresses are configured.

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface fastethernet 3/12
Switch(config-if)# switchport mode access
Switch(config-if)# switchport port-security
Switch(config-if)# switchport port-security maximum 5
Switch(config-if)# switchport port-security mac-address sticky
Switch(config-if)# end
Switch# show port-security interface fastethernet 3/12
Port Security                : Enabled
Port Status                  : Secure-up
Violation Mode                : Shutdown
Aging Time                   : 0 mins
Aging Type                    : Absolute
SecureStatic Address Aging   : Enabled
Maximum MAC Addresses        : 5
Total MAC Addresses          : 0
Configured MAC Addresses     : 0
Sticky MAC Addresses         : 0
Last Source Address:Vlan    : 0000.0000.0000:0
Security Violation Count     : 0
```

Example 2: Setting a Violation Mode

This example shows how to set the violation mode on the Fast Ethernet interface 3/12 to restrict.

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface fastethernet 3/12
Switch(config-if)# switchport port-security violation restrict
Switch(config-if)# end
Switch#
```

SNMP traps can be enabled with a rate-limit to detect port-security violations due to restrict mode. The following example shows how to enable traps for port-security with a rate of 5 traps per second:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# snmp-server enable traps port-security trap-rate 5
Switch(config)# end
Switch#
```

Example 3: Setting the Aging Timer

This example shows how to set the aging time to 2 hours (120 minutes) for the secure addresses on the Fast Ethernet interface 5/1:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface fastethernet 5/1
Switch(config-if)# switchport port-security aging time 120
Switch(config-if)# end
Switch#
```

This example shows how to set the aging time to 2 minutes:

```
Switch(config-if)# switchport port-security aging time 2
```

You can verify the previous commands with the **show port-security interface** command.

Example 4: Setting the Aging Timer Type

This example shows how to set the aging timer type to Inactivity for the secure addresses on the Fast Ethernet interface 3/5:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface fastethernet 3/5
Switch(config-if)# switchport port-security aging type inactivity
Switch(config-if)# end
Switch# show port-security interface fastethernet 3/5
Port Security           : Enabled
Port Status             : Secure-up
Violation Mode          : Shutdown
Aging Time              : 0 mins
Aging Type              : Inactivity
SecureStatic Address Aging : Disabled
Maximum MAC Addresses   : 1
Total MAC Addresses     : 0
Configured MAC Addresses : 0
Sticky MAC Addresses    : 0
Last Source Address:Vlan : 0000.0000.0000:0
Security Violation Count : 0
```

Example 5: Configuring a Secure MAC Address

This example shows how to configure a secure MAC address on Fast Ethernet interface 5/1 and to verify the configuration:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface fastethernet 5/1
Switch(config-if)# switchport mode access
Switch(config-if)# switchport port-security
Switch(config-if)# switchport port-security maximum 10
Switch(config-if)# switchport port-security mac-address 0000.0000.0003 (Static secure MAC)
Switch(config-if)# end
Switch#show port address
Secure Mac Address Table
-----
```

Vlan	Mac Address	Type	Ports	Remaining Age (mins)
1	0000.0000.0003	SecureConfigured	Fa5/1	-

```
-----
Total Addresses in System (excluding one mac per port) : 2
Max Addresses limit in System (excluding one mac per port) : 3072
```


Example 6: Configuring Sticky Port Security

This example shows how to configure a sticky MAC address on Fast Ethernet interface 5/1 and to verify the configuration:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface fa5/1
Switch(config-if)# switchport mode access
Switch(config-if)# switchport port-security
Switch(config-if)# switchport port-security mac-address sticky
Switch(config-if)# switchport port-security maximum 5
Switch(config-if)# end
```



Note

Sending traffic to the ports causes the system to configure the port with sticky secure addresses.

```
Switch# show port-security address
Secure Mac Address Table
-----
Vlan    Mac Address      Type                Ports    Remaining Age
(mins)
-----
1       0000.0000.0001   SecureSticky        Fa5/1    -
1       0000.0000.0002   SecureSticky        Fa5/1    -
1       0000.0000.0003   SecureSticky        Fa5/1    -
-----
Total Addresses in System (excluding one mac per port) : 2
Max Addresses limit in System (excluding one mac per port) : 3072
Switch# show running-config interface fastEthernet 5/1
Building configuration...

Current configuration : 344 bytes
!
interface FastEthernet5/1
 switchport mode access
 switchport port-security
 switchport port-security maximum 5
 switchport port-security mac-address sticky
 switchport port-security mac-address sticky 0000.0000.0001
 switchport port-security mac-address sticky 0000.0000.0002
 switchport port-security mac-address sticky 0000.0000.0003
end

Switch#
```

Example 7: Setting a Rate Limit for Bad Packets

The following example shows how to configure rate limit for invalid source packets on Fast Ethernet interface 5/1:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface fastethernet 5/1
Switch(config-if)# switchport port-security limit rate invalid-source-mac 100
Switch(config-if)# end
Switch#
```

The following example shows how to configure rate limit for invalid source packets on Fast Ethernet interface 5/1:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface fastethernet 5/1
Switch(config-if)# switchport port-security limit rate invalid-source-mac none
Switch(config-if)# end
Switch#
```

Example 8: Clearing Dynamic Secure MAC Addresses

The following example shows how to clear a dynamic secure MAC address:

```
Switch# clear port-security dynamic address 0000.0001.0001
```

The following example shows how to clear all dynamic secure MAC addresses on Fast Ethernet interface 2/1:

```
Switch# clear port-security dynamic interface fa2/1
```

The following example shows how to clear all dynamic secure MAC addresses in the system:

```
Switch# clear port-security dynamic
```

Configuring Port Security on PVLAN Ports

You can configure port security on a private VLAN port to take advantage of private VLAN functionality as well as to limit the number of MAC addresses.



Note

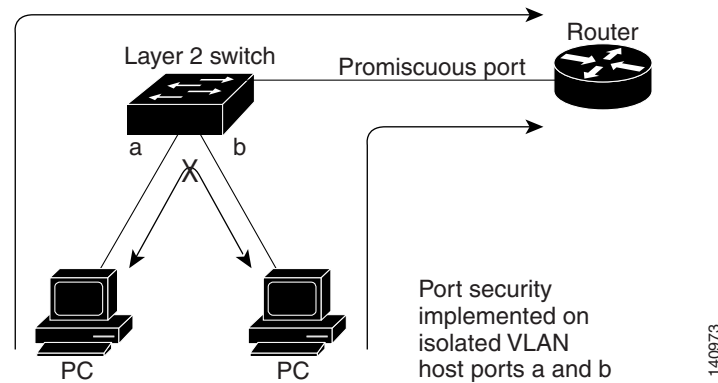
This section follows the same configuration model that was presented for access ports.

These sections describe how to configure trunk port security on host and promiscuous ports:

- [Configuring Port Security on an Isolated Private VLAN Host Port, page 55-14](#)
- [Example of Port Security on an Isolated Private VLAN Host Port, page 55-16](#)
- [Configuring Port Security on a Private VLAN Promiscuous Port, page 55-16](#)
- [Example of Port Security on a Private VLAN Promiscuous Port, page 55-17](#)

Configuring Port Security on an Isolated Private VLAN Host Port

[Figure 55-1](#) illustrates a typical topology for port security implemented on private VLAN host ports. In this topology, the PC connected through port a on the switch can communicate only with the router connected using the promiscuous port on the switch. The PC connected through port a cannot communicate with the PC connected through port b.

Figure 55-1 Port Security on Isolated Private VLAN Host Ports**Note**

Dynamic addresses secured on an isolated private VLAN host port on private VLANs are secured on the secondary VLANs, and not primary VLANs.

To configure port security on an isolated private VLAN host port, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enter global configuration mode.
Step 2	Switch(config)# vlan sec_vlan_id	Specifies a secondary VLAN.
Step 3	Switch(config-vlan)# private-vlan isolated	Sets the private VLAN mode to isolated.
Step 4	Switch(config-vlan)# exit	Returns to global configuration mode.
Step 5	Switch(config)# vlan pri_vlan_id	Specifies a primary VLAN.
Step 6	Switch(config-vlan)# private-vlan primary	Specifies the VLAN as the primary private VLAN.
Step 7	Switch(config-vlan)# private-vlan association add sec_vlan_id	Creates an association between a secondary VLAN and a primary VLAN.
Step 8	Switch(config-vlan)# exit	Returns to global configuration mode.
Step 9	Switch(config)# interface interface_id	Enters interface configuration mode and specifies the physical interface to configure.
Step 10	Switch(config-if)# switchport mode private-vlan host	Specifies that the ports with a valid private VLAN trunk association become active host private VLAN trunk ports.
Step 11	Switch(config-if)# switchport private-vlan host-association primary_vlan secondary_vlan	Establishes a host association on an isolated host port.
Step 12	Switch(config-if)# [no] switchport port-security	Enables port security on the interface.
Step 13	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 14	Switch# show port-security address interface interface_id Switch# show port-security address	Verifies your entries.

Example of Port Security on an Isolated Private VLAN Host Port

The following example shows how to configure port security on an isolated private VLAN host port, Fast Ethernet interface 3/12:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# vlan 6
Switch(config-vlan)# private-vlan isolated
Switch(config-vlan)# exit
Switch(config)# vlan 3
Switch(config-vlan)# private-vlan primary
Switch(config-vlan)# private-vlan association add 6
Switch(config-vlan)# exit
Switch(config)# interface fastethernet 3/12
Switch(config-if)# switchport mode private-vlan host
Switch(config-if)# switchport private-vlan association host 3 6
Switch(config-if)# switchport port-security
Switch(config-if)# end
```

Configuring Port Security on a Private VLAN Promiscuous Port

To configure port security on a private VLAN promiscuous port, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# vlan <i>sec_vlan_id</i>	Specifies the VLAN.
Step 3	Switch(config-vlan)# private-vlan isolated	Sets the private VLAN mode to isolated.
Step 4	Switch(config-vlan)# exit	Returns to global configuration mode.
Step 5	Switch(config)# vlan <i>pri_vlan_id</i>	Specifies the VLAN.
Step 6	Switch(config-vlan)# private-vlan primary	Designates the VLAN as the primary private VLAN.
Step 7	Switch(config-vlan)# private-vlan association add <i>sec_vlan_id</i>	Creates an association between a secondary VLAN and a primary VLAN.
Step 8	Switch(config-vlan)# exit	Returns to global configuration mode.
Step 9	Switch(config)# interface <i>interface_id</i>	Enters interface configuration mode and specifies the physical interface to configure.
Step 10	Switch(config-if)# switchport mode private-vlan promiscuous	Specifies that the ports with a valid PVLAN mapping become active promiscuous ports.
Step 11	Switch(config-if)# switchport private-vlan mapping <i>primary_vlan secondary_vlan</i>	Configures a private VLAN for the promiscuous ports.
Step 12	Switch(config-if)# switchport port-security	Enables port security on the interface.
Step 13	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 14	Switch# show port-security address interface <i>interface_id</i> Switch# show port-security address	Verifies your entries.

Example of Port Security on a Private VLAN Promiscuous Port

The following example shows how to configure port security on a private VLAN promiscuous port, Fast Ethernet interface 3/12:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# vlan 6
Switch(config-vlan)# private-vlan isolated
Switch(config-vlan)# exit
Switch(config)# vlan 3
Switch(config-vlan)# private-vlan primary
Switch(config-vlan)# private-vlan association add 6
Switch(config-vlan)# exit
Switch(config)# interface fastethernet 3/12
Switch(config-if)# switchport mode private-vlan promiscuous
Switch(config-if)# switchport mode private-vlan mapping 3 6
Switch(config-if)# switchport port-security
Switch(config-if)# end
```

Configuring Port Security on Trunk Ports

You might want to configure port security on trunk ports in metro aggregation to limit the number of MAC addresses per-VLAN. Trunk port security extends port security to trunk ports. It restricts the allowed MAC addresses or the maximum number of MAC addresses to individual VLANs on a trunk port. Trunk port security enables service providers to block the access from a station with a different MAC address than the ones specified for that VLAN on that trunk port. Trunk port security is also supported on private VLAN trunk ports.



Note

Port security can be enabled on a Layer 2 port channel interface configured in mode. The port security configuration on an EtherChannel is kept independent of the configuration of any physical member ports.

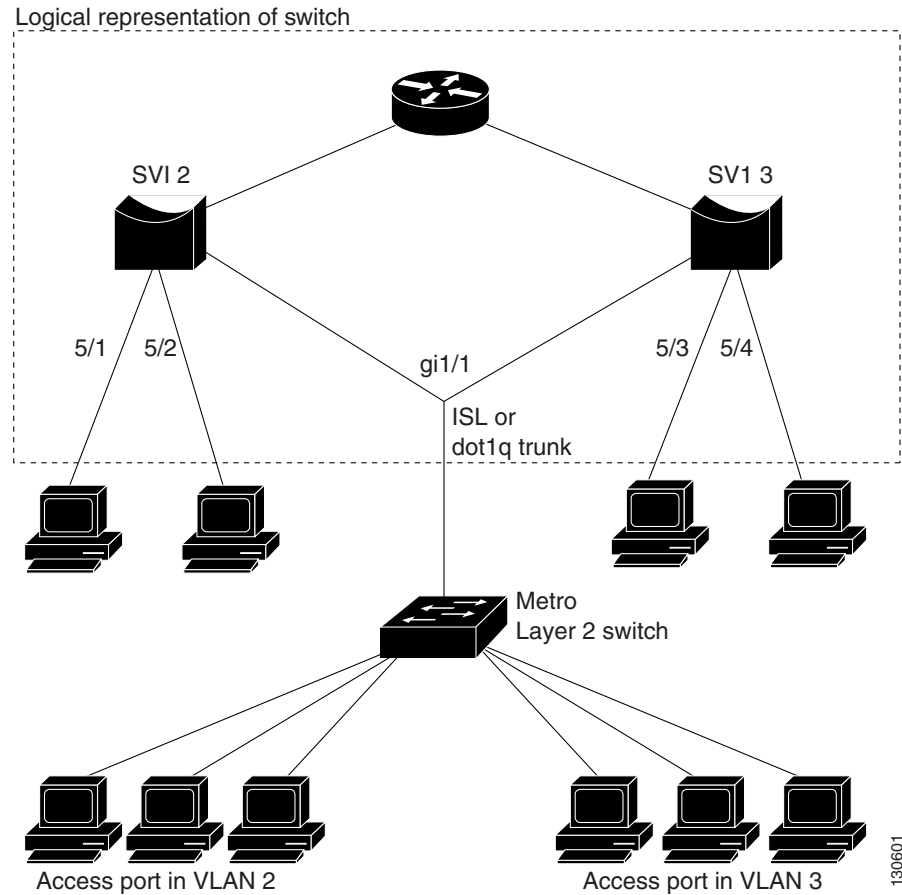
These sections describe how to configure trunk port security:

- [Configuring Trunk Port Security, page 55-17](#)
- [Examples of Trunk Port Security, page 55-19](#)
- [Trunk Port Security Configuration Guidelines and Restrictions, page 55-21](#)

Configuring Trunk Port Security

Trunk port security is used when a Catalyst 4500 series switch has a dot1q or isl trunk attached to a neighborhood Layer 2 switch. This may be used, for example, in metro aggregation networks ([Figure 55-2](#)).

Figure 55-2 Trunk Port Security



You can configure various port security related parameters on a per-port per-VLAN basis.



Note

The steps involved in configuring port security parameters is similar to those for access ports. In addition to those steps, the following per-port per-VLAN configuration steps are supported for trunk ports.

To configure port security related parameters on a per-VLAN per-port basis, perform this task:

Command	Purpose
Step 1 Switch(config)# interface <i>interface_id</i> interface <i>port-channel port_channel_number</i>	Enters interface configuration mode and specifies the interface to configure. Note The interface can be a Layer 2 port channel logical interface.
Step 2 Switch(config-if)# switchport trunk encapsulation dot1q	Sets the trunk encapsulation format to 802.1Q.
Step 3 Switch(config-if)# switchport mode trunk	Sets the interface mode. Note An interface in the default mode (dynamic auto) cannot be configured as a secure port.

	Command	Purpose
Step 4	Switch(config-if)# switchport port-security maximum value vlan	Configures a maximum number of secure mac-addresses for each VLAN on the interface that are not explicitly configured with a maximum mac-address limit. See the “Maximum Number of Secure MAC Addresses” section on page 55-4.
Step 5	Switch(config-if)# vlan-range range	Enters VLAN range sub-mode. Note You can specify single or multiple VLANs.
Step 6	Switch(config-if-vlan-range)# port-security maximum value	Configures a maximum number of secure MAC addresses for each VLAN.
Step 7	Switch(config-if-vlan-range)# no port-security maximum	Removes a maximum number of secure MAC addresses configuration for all the VLANs. Subsequently, the maximum value configured on the port will be used for all the VLANs.
Step 8	Switch(config-if-vlan-range)# [no] port-security mac-address mac_address	Configures a secure MAC-address on a range of VLANs.
Step 9	Switch(config-if-vlan-range)# [no] port-security mac-address sticky mac_address	Configures a sticky MAC-address on a range of VLANs.
Step 10	Switch(config-if-vlan-range)# end	Returns to interface configuration mode.
Step 11	Switch(config-if)# end	Returns to privileged EXEC mode.

Examples of Trunk Port Security

The following examples are provided:

- [Example 1: Configuring a Maximum Limit of Secure MAC Addresses for All VLANs, page 55-19](#)
- [Example 2: Configuring a Maximum Limit of Secure MAC Addresses for Specific VLANs, page 55-20](#)
- [Example 3: Configuring Secure MAC Addresses in a VLAN Range, page 55-20](#)

Example 1: Configuring a Maximum Limit of Secure MAC Addresses for All VLANs

This example shows how to configure a secure MAC-address and a maximum limit of secure MAC addresses on Gigabit Ethernet interface 1/1 for all VLANs:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface g1/1
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# sw mode trunk
Switch(config-if)# switchport port-security
Switch(config-if)# switchport port-security maximum 3

Switch# show port-security in g1/1 vlan
Default maximum: 3
VLAN  Maximum      Current
    1         3         0
    2         3         0
    3         3         0
    4         3         0
```

```

5          3          0
6          3          0
Switch#

Switch# show running interface g1/1
Building configuration...

Current configuration : 161 bytes
!
interface GigabitEthernet1/1
 switchport trunk encapsulation dot1q
 switchport mode trunk
 switchport port-security
 switchport port-security maximum 3 vlan
end

```

Example 2: Configuring a Maximum Limit of Secure MAC Addresses for Specific VLANs

This example shows how to configure a secure MAC-address on interface g1/1 in a specific VLAN or range of VLANs:

```

Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface g1/1
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# sw mode trunk
Switch(config-if)# switchport port-security
Switch(config-if)# vlan-range 2-6
Switch(config-if-vlan-range)# port-security maximum 3
Switch(config-if-vlan-range)# exit

Switch# show port-security interface g1/1 vlan
Default maximum: not set, using 3072
VLAN  Maximum    Current
2          3          0
3          3          0
4          3          0
5          3          0
6          3          0
Switch#

```

Example 3: Configuring Secure MAC Addresses in a VLAN Range

This example shows how to configure a secure MAC-address in a VLAN on interface g1/1:

```

Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface g1/1
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# sw mode trunk
Switch(config-if)# switchport port-security
Switch(config-if)# switchport port-security mac-address sticky
Switch(config-if)# vlan-range 2-6
Switch(config-if-vlan-range)# port-security mac-address 1.1.1
Switch(config-if-vlan-range)# port-security mac-address sticky 1.1.2
Switch(config-if-vlan-range)# port-security mac-address sticky 1.1.3
Switch(config-if-vlan-range)# exit

```



```
Switch# show port-security interface g1/1 address vlan 2-4
Secure Mac Address Table
```

Vlan	Mac Address	Type	Ports	Remaining Age (mins)
2	0001.0001.0001	SecureConfigured	Gi1/1	-
2	0001.0001.0002	SecureSticky	Gi1/1	-
2	0001.0001.0003	SecureSticky	Gi1/1	-
3	0001.0001.0001	SecureConfigured	Gi1/1	-
3	0001.0001.0002	SecureSticky	Gi1/1	-
3	0001.0001.0003	SecureSticky	Gi1/1	-
4	0001.0001.0001	SecureConfigured	Gi1/1	-
4	0001.0001.0002	SecureSticky	Gi1/1	-
4	0001.0001.0003	SecureSticky	Gi1/1	-

```
Total Addresses: 9
```

```
Switch#
```

Trunk Port Security Configuration Guidelines and Restrictions

When configuring port security related parameters on a per-port per-VLAN basis, consider these guidelines and restrictions:

- A secure MAC-address cannot be configured on a VLAN that is not allowed on a regular trunk port.
- The configuration on the primary VLAN on the private VLAN trunk is not allowed. The CLI is rejected and an error message is displayed.
- If a specific VLAN on a port is not configured with a maximum value (directly or indirectly), the maximum configured for the port is used for that VLAN. In this situation, the maximum number of addresses that can be secured on this VLAN is limited to the maximum value configured on the port.

Each VLAN can be configured with a maximum count that is greater than the value configured on the port. Also, the sum of the maximum configured values for all the VLANs can exceed the maximum configured for the port. In either of these situations, the number of MAC addresses secured on each VLAN is limited to the lesser of the VLAN configuration maximum and the port configuration maximum. Also, the number of addresses secured on the port across all VLANs cannot exceed a maximum that is configured on the port.

- For private VLAN trunk ports, the VLAN on which the configuration is being performed must be in either the allowed VLAN list of the private VLAN trunk or the secondary VLAN list in the association pairs. (The CLI is rejected if this condition is not met.) The allowed VLAN list on a private VLAN trunk is intended to hold the VLAN-IDs of all the regular VLANs that are allowed on the private VLAN trunk.
- Removal of an association pair from a PVLAN trunk causes all static and sticky addresses associated with the secondary VLAN of the pair to be removed from the running configuration. Dynamic addresses associated with the secondary VLAN are deleted from the system.

Similarly, when a VLAN is removed from the list of allowed PVLAN trunks, the addresses associated with that VLAN are removed.



Note

For a regular or private VLAN trunk port, if the VLAN is removed from the allowed VLAN list, all the addresses associated with that VLAN are removed.

Port Mode Changes

Generally, when a port mode changes, all dynamic addresses associated with that port are removed. All static or sticky addresses and other port security parameters configured on the native VLAN are moved to the native VLAN of the port in the new mode. All the addresses on the non-native VLANs are removed.

The native VLAN refers to the following VLAN on the specified port type:

Port Type	Native VLAN
access	access VLAN
trunk	native VLAN
isolated	secondary VLAN (from host association)
promiscuous	primary VLAN (from mapping)
private VLAN trunk	private VLAN trunk native VLAN
.1Q tunnel	access VLAN

For example, when the mode changes from access to private VLAN trunk, all the static or sticky addresses configured on the access VLAN of the access port are moved to the private VLAN native VLAN of the private VLAN trunk port. All other addresses are removed.

Similarly, when the mode changes from private VLAN trunk to access mode, all the static or sticky addresses configured on the private VLAN native VLAN are moved to the access VLAN of the access port. All other addresses are removed.

When a port is changed from trunk to private VLAN trunk, addresses associated with a VLAN on the trunk are retained if that VLAN is present in the allowed list of private VLAN trunk or the secondary VLAN of an association on the private VLAN trunk. If the VLAN is not present in either of them, the address is removed from the running configuration.

When a port is changed from private VLAN trunk to trunk, a static or sticky address is retained if the VLAN associated with the address is present in the allowed VLAN list of the trunk. If the VLAN is not present in the allowed list, the address is removed from running configuration.

Configuring Port Security on Voice Ports

You might want to configure port security in an IP phone environment when a port is configured with a data VLAN for a PC and a voice VLAN for a Cisco IP Phone.

These sections describe how to configure port security on voice ports:

- [Configuring Port Security on Voice Ports, page 55-23](#)
- [Examples of Voice Port Security, page 55-25](#)
- [Voice Port Security Configuration Guidelines and Restrictions, page 55-27](#)

Configuring Port Security on Voice Ports

To configure port security on a voice port, perform this task:

	Command	Purpose
Step 1	Switch(config)# interface <i>interface_id</i>	Enters interface configuration mode and specifies the physical interface to configure.
Step 2	Switch(config-if)# switchport mode access	Sets the interface mode. Note An interface in the default mode (dynamic auto) cannot be configured as a secure port.
Step 3	Switch(config-if)# [no] switchport port-security	Enables port security on the interface. To return the interface to the default condition as not secured, use the no switchport port-security command.
Step 4	Switch(config-if)# [no] switchport port-security violation { restrict shutdown }	(Optional) Sets the violation mode, the action to be taken when a security violation is detected, as one of these: <ul style="list-style-type: none"> • restrict—A port security violation restricts data and causes the SecurityViolation counter to increment and send an SNMP trap notification. • shutdown—The interface is error-disabled when a security violation occurs. Note When a secure port is in the error-disabled state, you can bring it out of this state by entering the errdisable recovery cause psecure-violation global configuration command or you can manually reen able it by entering the shutdown and no shut down interface configuration commands. To return the violation mode to the default condition (shutdown mode), use the no switchport port-security violation shutdown command.
Step 5	Switch(config-if)# switchport port-security limit rate invalid-source-mac <i>packets_per_sec</i>	Sets the rate limit for bad packets. Default is 10 pps.

Command	Purpose
Step 6 Switch(config-if)# [no] switchport port-security mac-address <i>mac_address</i> [vlan { voice access }]	<p>(Optional) Specifies a secure MAC address for the interface.</p> <p>When you specify the vlan keyword, addresses are configured in the specified VLAN.</p> <ul style="list-style-type: none"> • voice—MAC address is configured in the voice VLAN. • access—MAC address is configured in the access VLAN. <p>You can use this command to configure secure MAC addresses. If you configure fewer secure MAC addresses than the maximum, the remaining MAC addresses are dynamically learned.</p> <p>To delete a MAC address from the address table, use the no switchport port-security mac-address <i>mac_address</i> command.</p> <p>Note This command only applies to access, PVLAN host, and PVLAN promiscuous mode. For more details on PVLAN, trunk, or regular trunk mode, refer to the “Configuring Port Security on Trunk Ports” section on page 55-17.</p>
Step 7 Switch(config-if)# [no] switchport port-security mac-address sticky	<p>(Optional) Enables sticky learning on the interface.</p> <p>To disable sticky learning on an interface, use the no switchport port-security mac-address sticky command. The interface converts the sticky secure MAC addresses to dynamic secure addresses.</p>
Step 8 Switch(config-if)# [no] switchport port-security mac-address <i>mac_address</i> sticky [vlan { voice access }]	<p>Specifies the sticky mac-address for the interface.</p> <p>When you specify the vlan keyword, the mac-address becomes sticky in the specified VLAN.</p> <ul style="list-style-type: none"> • voice—MAC address becomes sticky in the voice VLAN. • access—MAC address becomes sticky in the access VLAN. <p>To delete a sticky secure MAC addresses from the address table, use the no switchport port-security mac-address <i>mac_address</i> sticky command. To convert sticky to dynamic addresses, use the no switchport port-security mac-address sticky command.</p> <p>Note This command only applies to access, PVLAN host, and PVLAN promiscuous mode. For more details on PVLAN or trunk or regular trunk mode, refer to the “Configuring Port Security on Trunk Ports” section on page 55-17.</p>

	Command	Purpose
Step 9	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 10	Switch# show port-security address Switch# show port-security address Switch# show port-security address	Verifies your entries.

**Note**

To clear dynamically learned port security MAC addresses in the CAM table, use the **clear port-security dynamic** command. The **address** keyword enables you to clear a secure MAC addresses. The **interface** keyword enables you to clear all secure addresses on an interface (including any port channel interface). The **VLAN** keyword allows you to clear port security MACs on a per-VLAN per-port basis.

**Note**

Each port security-configured interface accepts one MAC-address by default. With port security port level port-security configuration takes precedence over VLAN level port-security configuration. To allow one MAC-address each for voice and data VLAN, configure the port for a maximum of greater than or equal to two addresses.

Examples of Voice Port Security

The following examples are provided:

- [Example 1: Configuring Maximum MAC Addresses for Voice and Data VLANs, page 55-25](#)
- [Example 2: Configuring Sticky MAC Addresses for Voice and Data VLANs, page 55-26](#)

Example 1: Configuring Maximum MAC Addresses for Voice and Data VLANs

This example shows how to designate a maximum of one MAC address for a voice VLAN (for a Cisco IP Phone, let's say) and one MAC address for the data VLAN (for a PC, let's say) on Fast Ethernet interface 5/1 and to verify the configuration:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface fa5/1
Switch(config-if)# switchport mode access
Switch(config-if)# switchport port-security
Switch(config-if)# switchport port-security maximum 2
Switch(config-if)# switchport port-security mac-address sticky
Switch(config-if)# switchport port-security maximum 1 vlan voice
Switch(config-if)# switchport port-security maximum 1 vlan access
Switch(config-if)# end
```

**Note**

Sending traffic to the ports causes the system to configure the port with sticky secure addresses.

```
Switch# show port-security address
      Secure Mac Address Table
-----
Vlan    Mac Address      Type                Ports    Remaining Age
-----
      (mins)
-----
      1    0000.0000.0001    SecureSticky        Fa5/1    -
      3    0000.0000.0004    SecureSticky        Fa5/1    -
-----
Total Addresses in System (excluding one mac per port)    : 1
Max Addresses limit in System (excluding one mac per port) : 3072

Switch# show running-config interface fastEthernet 5/1
Building configuration...

Current configuration : 344 bytes
!
interface FastEthernet5/1
 switchport mode access
 switchport voice vlan 3
 switchport port-security
 switchport port-security maximum 1 vlan voice
 switchport port-security maximum 3072
 switchport port-security maximum 1 vlan access
 switchport port-security mac-address sticky
 switchport port-security mac-address sticky 0000.0000.0001
 switchport port-security mac-address sticky 0000.0000.0004 vlan voice
end

Switch#
```

Example 2: Configuring Sticky MAC Addresses for Voice and Data VLANs

This example shows how to configure sticky MAC addresses for voice and data VLANs on Fast Ethernet interface 5/1 and to verify the configuration:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface fa5/1
Switch(config-if)# switchport mode access
Switch(config-if)# switchport port-security
Switch(config-if)# switchport port-security maximum 3072
Switch(config-if)# switchport port-security mac-address sticky 0000.0000.obob vlan voice
Switch(config-if)# switchport port-security mac-address sticky 0000.0000.0005 vlan access
Switch(config-if)# end
```



Note

Sending traffic to the ports causes the system to configure the port with sticky secure addresses.

```
Switch# show port-security address
      Secure Mac Address Table
-----
Vlan    Mac Address      Type                Ports    Remaining Age
-----
      (mins)
-----
      1    0000.0000.0001    SecureSticky        Fa5/1    -
      1    0000.0000.0002    SecureSticky        Fa5/1    -
      1    0000.0000.0003    SecureSticky        Fa5/1    -
      3    0000.0000.0004    SecureSticky        Fa5/1    -
      1    0000.0000.0005    SecureSticky        Fa5/1    -
      3    0000.0000.0b0b    SecureSticky        Fa5/1    -
-----
```

```

Total Addresses in System (excluding one mac per port)      : 5
Max Addresses limit in System (excluding one mac per port) : 3072

Switch# show running-config interface fastEthernet 5/1
Building configuration...

Current configuration : 344 bytes
!
interface FastEthernet5/1
 switchport mode access
 switchport voice vlan 3
 switchport port-security
 switchport port-security maximum 3072
 switchport port-security maximum 5 vlan voice
 switchport port-security mac-address sticky
 switchport port-security mac-address sticky 0000.0000.0001
 switchport port-security mac-address sticky 0000.0000.0002
 switchport port-security mac-address sticky 0000.0000.0003
 switchport port-security mac-address sticky 0000.0000.0004 vlan voice
 switchport port-security mac-address sticky 0000.0000.0005
 switchport port-security mac-address sticky 0000.0000.0b0b vlan voice
end

Switch#

```

Voice Port Security Configuration Guidelines and Restrictions



Note

When a Catalyst 4500 series switch port is configured to support voice as well as port security, a security violation occurs on subsequent shut down when the number of MAC addresses on the port is equal to the maximum number of allowable secure MAC addresses. To avoid the security violation, configure the maximum number of secure MAC addresses to be more than the number of MAC addresses on the port.



Note

Port security as implemented on voice ports functions the same as port security on access ports.

When using (or configuring) voice port security, consider these guidelines and restrictions:

- You can configure sticky port security on voice ports. If sticky port security is enabled on a voice port, addresses secured on data and voice VLANs are secured as sticky addresses.
- You can configure maximum secure addresses per-VLAN. You can set a maximum for either the data VLAN or the voice VLAN. You can also set a maximum per-port, just as with access ports.
- You can configure port security MAC addresses on a per-VLAN basis on either the data or voice VLANs.
- Prior to Cisco IOS Release 12.2(31)SG, you required three MAC addresses as the maximum parameter to support an IP phone and a PC. With Cisco IOS Release 12.2(31)SG and later releases, the maximum parameter must be configured to two, one for the phone and one for the PC.

Displaying Port Security Settings

Use the **show port-security** command to display port security settings for an interface or for the switch.

To display traffic control information, perform one or more of these tasks:

Command	Purpose
Switch# show interface status err-disable	Displays interfaces that have been error-disabled along with the cause for which they were disabled.
Switch# show port-security [interface <i>interface_id</i> / interface <i>port_channel port_channel_number</i>]	Displays port security settings for the switch or for the specified interface, including the maximum allowed number of secure MAC addresses for each interface, the number of secure MAC addresses on the interface, the number of security violations that have occurred, and the violation mode. The interface can be a port channel logical interface.
Switch# show port-security [interface <i>interface_id</i> / interface <i>port_channel port_channel_number</i>] address	Displays all secure MAC addresses configured on all switch interfaces or on a specified interface with aging information for each address.
Switch# show port-security [interface <i>interface_id</i> / interface <i>port_channel port_channel_number</i>] vlan <i>vlan_list</i>	Displays the maximum allowed number of secure MAC addresses and the current number of secure MAC addresses on a specific VLAN-list and a specific interface.
Switch# show port-security [interface <i>interface_id</i> / interface <i>port_channel port_channel_number</i>] [address [vlan <i>vlan_list</i>]]	Displays all secure MAC addresses configured on a specific VLAN-list and a specific interface.

Examples of Security Settings

The following examples are provided:

- [Example 1: Displaying Security Settings for the Entire Switch, page 55-28](#)
- [Example 2: Displaying Security Settings for an Interface, page 55-29](#)
- [Example 3: Displaying All Secure Addresses for the Entire Switch, page 55-29](#)
- [Example 4: Displaying a Maximum Number of MAC Addresses on an Interface, page 55-30](#)
- [Example 5: Displaying Security Settings on an Interface for a VLAN Range, page 55-30](#)
- [Example 6: Displaying Secured MAC Addresses and Aging Information on an Interface, page 55-30](#)
- [Example 7: Displaying Secured MAC Addresses for a VLAN Range on an Interface, page 55-31](#)

Example 1: Displaying Security Settings for the Entire Switch

This example shows how to display port security settings for the entire switch:

```
Switch# show port-security
Secure Port  MaxSecureAddr  CurrentAddr  SecurityViolation  Security Action
          (Count)          (Count)          (Count)
-----
Fa3/1             2             2             0             Restrict
Fa3/2             2             2             0             Restrict
Fa3/3             2             2             0             Shutdown
Fa3/4             2             2             0             Shutdown
Fa3/5             2             2             0             Shutdown
```



```

Fa3/6          2          2          0          Shutdown
Fa3/7          2          2          0          Shutdown
Fa3/8          2          2          0          Shutdown
Fa3/10         1          0          0          Shutdown
Fa3/11         1          0          0          Shutdown
Fa3/12         1          0          0          Restrict
Fa3/13         1          0          0          Shutdown
Fa3/14         1          0          0          Shutdown
Fa3/15         1          0          0          Shutdown
Fa3/16         1          0          0          Shutdown
Po2            3          0          0          Shutdown
-----
Total Addresses in System (excluding one mac per port) :8
Max Addresses limit in System (excluding one mac per port) :3072
Global SNMP trap control for port-security :20 (traps per second)

```

Example 2: Displaying Security Settings for an Interface

This example shows how to display port security settings for Fast Ethernet interface 5/1:

```

Switch# show port-security interface fastethernet 5/1
Port Security          : Enabled
Port Status            : Secure-up
Violation Mode         : Shutdown
Aging Time             : 0 mins
Aging Type             : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses  : 1
Total MAC Addresses    : 1
Configured MAC Addresses : 0
Sticky MAC Addresses   : 1
Last Source Address:Vlan : 0000.0001.001a:1
Security Violation Count : 0

```

Example 3: Displaying All Secure Addresses for the Entire Switch

This example shows how to display all secure MAC addresses configured on all switch interfaces:

```

Switch# show port-security address
Secure Mac Address Table
-----
Vlan    Mac Address      Type                Ports    Remaining Age
      (mins)
-----
1       0000.0001.0000   SecureConfigured    Fa3/1    15 (I)
1       0000.0001.0001   SecureConfigured    Fa3/1    14 (I)
1       0000.0001.0100   SecureConfigured    Fa3/2    -
1       0000.0001.0101   SecureConfigured    Fa3/2    -
1       0000.0001.0200   SecureConfigured    Fa3/3    -
1       0000.0001.0201   SecureConfigured    Fa3/3    -
1       0000.0001.0300   SecureConfigured    Fa3/4    -
1       0000.0001.0301   SecureConfigured    Fa3/4    -
1       0000.0001.1000   SecureDynamic       Fa3/5    -
1       0000.0001.1001   SecureDynamic       Fa3/5    -
1       0000.0001.1100   SecureDynamic       Fa3/6    -
1       0000.0001.1101   SecureDynamic       Fa3/6    -
1       0000.0001.1200   SecureSticky        Fa3/7    -

```

```

1      0000.0001.1201      SecureSticky      Fa3/7      -
1      0000.0001.1300      SecureSticky      Fa3/8      -
1      0000.0001.1301      SecureSticky      Fa3/8      -
1      0000.0001.2000      SecureSticky      Po2        -
-----
Total Addresses in System (excluding one mac per port)      :8
Max Addresses limit in System (excluding one mac per port) :3072

```

Example 4: Displaying a Maximum Number of MAC Addresses on an Interface

This example shows how to display the maximum allowed number of secure MAC addresses and the current number of secure MAC addressees on Gigabit Ethernet interface 1/1:

```

Switch# show port-security interface g1/1 vlan
Default maximum: 22
VLAN  Maximum      Current
2          22          3
3          22          3
4          22          3
5          22          1
6          22          2

```

Example 5: Displaying Security Settings on an Interface for a VLAN Range

This example shows how to display the port security settings on Gigabit Ethernet interface 1/1 for VLANs 2 and 3:

```

Switch# show port-security interface g1/1 vlan 2-3
Default maximum: 22
VLAN  Maximum      Current
2          22          3
3          22          3

```

Example 6: Displaying Secured MAC Addresses and Aging Information on an Interface

This example shows how to display all secure MAC addresses configured on Gigabit Ethernet interface 1/1 with aging information for each address.

```

Switch# show port-security interface g1/1 address

```

```

          Secure Mac Address Table
-----
Vlan      Mac Address      Type      Ports      Remaining Age(mins)
----      -
2      0001.0001.0001      SecureConfigured      Gi1/1      -
2      0001.0001.0002      SecureSticky      Gi1/1      -
2      0001.0001.0003      SecureSticky      Gi1/1      -
3      0001.0001.0001      SecureConfigured      Gi1/1      -
3      0001.0001.0002      SecureSticky      Gi1/1      -
3      0001.0001.0003      SecureSticky      Gi1/1      -
4      0001.0001.0001      SecureConfigured      Gi1/1      -
4      0001.0001.0002      SecureSticky      Gi1/1      -
4      0001.0001.0003      SecureSticky      Gi1/1      -
5      0001.0001.0001      SecureConfigured      Gi1/1      -
6      0001.0001.0001      SecureConfigured      Gi1/1      -
6      0001.0001.0002      SecureConfigured      Gi1/1      -
-----
Total Addresses: 12

```

Example 7: Displaying Secured MAC Addresses for a VLAN Range on an Interface

This example shows how to display all secure MAC addresses configured on VLANs 2 and 3 on Gigabit Ethernet interface 1/1 with aging information for each address:

```
Switch# show port-security interface g1/1 address vlan 2-3
```

```

Secure Mac Address Table
-----
Vlan    Mac Address      Type                Ports    Remaining Age(mins)
-----
2       0001.0001.0001   SecureConfigured    Gi1/1    -
2       0001.0001.0002   SecureSticky         Gi1/1    -
2       0001.0001.0003   SecureSticky         Gi1/1    -
3       0001.0001.0001   SecureConfigured    Gi1/1    -
3       0001.0001.0002   SecureSticky         Gi1/1    -
3       0001.0001.0003   SecureSticky         Gi1/1    -
-----
Total Addresses: 12
Switch#
```

Configuring Port Security with Other Features/Environments

The following topics are discussed:

- [DHCP and IP Source Guard, page 55-31](#)
- [802.1X Authentication, page 55-32](#)
- [Configuring Port Security in a Wireless Environment, page 55-32](#)

DHCP and IP Source Guard

You might want to configure port security with DHCP and IP Source Guard to prevent IP spoofing by unsecured MAC addresses. IP Source Guard supports two levels of IP traffic filtering:

- Source IP address filtering
- Source IP and MAC address filtering

When used in source IP and MAC address filtering, IP Source Guard uses private ACLs to filter traffic based on the source IP address, and uses port security to filter traffic based on the source MAC address. Port security must be enabled on the access port in this mode.

When both features are enabled, the following limitations apply:

- The DHCP packet is not subject to port security dynamic learning.
- If multiple IP clients are connected to a single access port, port security cannot enforce exact binding of source IP and MAC address for each client.

For example, these clients reside on an access port with the following IP and MAC address:

- client1: MAC1 <---> IP1
 - client2: MAC2 <---> IP2
- Any combination of the source MAC and IP address traffic will be allowed as shown here:
- MAC1 <---> IP1, valid
 - MAC2 <---> IP2, valid

- MAC1 <---> IP2, invalid
- MAC2 <---> IP1, invalid

IP traffic with the correct source IP and MAC address binding will be permitted and port security will dynamically learn its MAC address. IP traffic with source addresses that are not in the binding will be treated as invalid packets and dropped by port security. To prevent a denial of service attack, you must configure port security rate limiting for the invalid source MAC address.

802.1X Authentication

You might want to configure port security with 802.1X authentication to prevent MAC spoofing. 802.1X is not supported on regular or private VLAN trunks. On access ports and PVLAN host or promiscuous ports, both port security and 802.1X can be configured simultaneously. When both are configured, hosts must be 802.1X authenticated before port security can secure the MAC address of the host. Both 802.1X and port security must approve of the host or a security violation will be triggered. The type of security violation will depend on which feature rejects the port: if the host is allowed by 802.1X (for example, because the port is in multihost mode) but is disallowed by port security, the port-security violation action will be triggered. If the host is allowed by port security but rejected by 802.1X (for example, because the host is not authorized on a single-host mode port) then the 802.1X security violation action will be triggered.



Note

802.1X, port-security and VVID can all be configured on the same port.

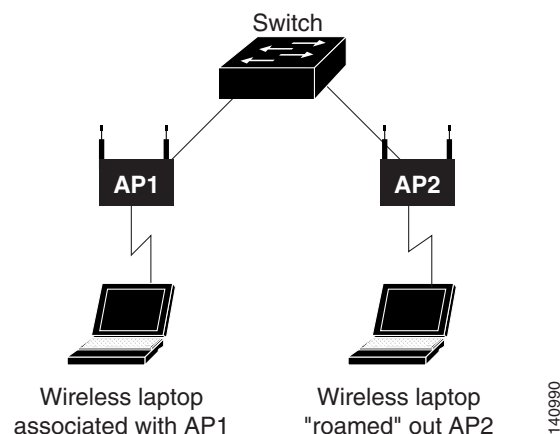
For more information on the interaction between 802.1X and port security, see “Using 802.1X with Port Security” on page 19.

Configuring Port Security in a Wireless Environment

If access points are connected to a secure port, do not configure a static MAC address for your users. A MAC address might move from one access point to another and might cause security violations if both the access points are connected on the same switch.

Figure 55-3 illustrates a typical topology of port security in a wireless environment.

Figure 55-3 Port Security in a Wireless Environment



Port Security Configuration Guidelines and Restrictions

When using (or configuring) port security, consider these guidelines and restrictions:

- After port security is configured on a port along with a "denying" PACL, the CPU will neither see any of the PACL packets denied from the given port nor learn the source MAC addresses from the denied packets. Therefore, the port security feature will not be aware of such packets.
- A secure port cannot be a destination port for the Switch Port Analyzer (SPAN).
- A secure port and a static MAC address configuration for an interface are mutually exclusive.
- When you enter a maximum secure address value for an interface, and the new value is greater than the previous value, the new value overwrites the previously configured value. If the new value is less than the previous value and the number of configured secure addresses on the interface exceeds the new value, the command is rejected.
- While configuring trunk port security on a trunk port, you do not need to account for the protocol packets such as CDP and BPDU) because they are not learned and secured.
- You cannot enable port security aging on sticky secure MAC addresses.
- To restrict MAC spoofing using port security, you must enable 802.1X authentication.
- You cannot configure port security on dynamic ports. You must change the mode to access before you enable port security.
- Port Security over EtherChannels is not supported.
- Wired guest access does not work on Supervisor Engine 8-E or 9-E, in multiple-host mode or in multi- authentication mode.



Configuring Auto Security

This chapter describes how to configure auto security on the Catalyst 4500 series switch.

It consists of these sections:

- [About Auto Security, page 56-1](#)
- [Feature Interaction, page 56-1](#)
- [Configuring Auto Security, page 56-2](#)
- [Guidelines and Restrictions, page 56-6](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About Auto Security

Prior to Release IOS XE 3.6.0E and IOS 15.2(2)E, the Catalyst 4500 series switch offered IPv4 baseline security features (like Port Security), which must be enabled globally and on per port basis. Moreover, the baseline security feature CLIs for uplink ports differ from those for downlink CLIs.

Beginning with Release IOS XE 3.6.0E and IOS 15.2(2)E, the Catalyst 4500 series switch supports Auto Security (AS), which provides a single line CLI, to enable base line security features.

AS supports the IPv4 baseline security features: DHCP Snooping, Dynamic ARP Inspection, and Port Security.

Feature Interaction

Auto security interacts with Port Security, DHCP snooping, DAI modules.

DHCP Snooping

Auto Security (AS) enables DHCP Snooping globally (with the **ip dhcp snooping** command) and also on VLANs 2-1005 (with the **ip dhcp snooping vlan *vlanid*** command).

AS configures trunk or DHCP server-facing port(s) as trusted (with the **ip dhcp-snooping trust** command).

Dynamic ARP Inspection

AS enables this feature globally on all VLANs present on the switch (with the **ip arp inspection vlan vlanid** command).

AS configures the trunk port as trusted (with the **ip arp inspection trust** command).

Port Security

AS enables this feature on all the switch's access ports (with the **switchport port-security** command).]

Configuring Auto Security

Enabling auto security globally

To enable auto security globally, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# auto security	Enables auto security globally.
Step 3	Switch(config)# end	Returns to privileged EXEC mode.
Step 4	Switch# show running-config i security	(Optional) Saves your entries in the configuration file.

This example shows how to enable auto security globally:

```
Switch(config)# auto security
Switch# show running-config | i security
auto security
```

Relevant baseline security feature CLI as shown in the output of the show auto security command is applied on or removed from access and trunk ports.

Disabling auto security globally

To disable auto security globally, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# no auto security	Dis-enables auto security globally.
Step 3	Switch(config)# end	Returns to privileged EXEC mode.
Step 4	Switch# show running-config isecurity	(Optional) Saves your entries in the configuration file.

This example show how to dis-enable auto security globally:

```
Switch(config)# no auto security
Switch# show auto security
Auto Security is Disabled globally

AutoSecure is Enabled on below interface(s):
-----

Switch#
```

Enabling Auto Security Feature for Access (End Hosts) or Trunk (Uplink) Ports

Use the **auto security-port [host | uplink]** command, to enable auto security for access (end hosts) and uplink ports:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface interface	Enters interface mode
Step 3	Switch(config-if)# auto security-port [host uplink]	Enables auto security on host or uplink ports.
Step 4	Switch(config)# end	Returns to privileged EXEC mode.
Step 5	Switch# show auto security	Displays the status of auto security.

This example displays how to enable auto security for an uplink port:

This example shows how to configure a port as auto security-port uplink.

```
Switch# conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# int g1/0/15
Switch(config-if)# switchport mode trunk
Switch(config-if)# auto security-port uplink
Switch(config-if)# end
```

Use the **show auto security** and **show running-config** commands confirm the prior configuration.

```
Switch# show auto security
Auto Security is Enabled globally

AutoSecure is Enabled on below interface(s):
-----
GigabitEthernet1/0/2
GigabitEthernet1/0/3
GigabitEthernet1/0/15

Switch# show running-config int g1/0/15
Building configuration...

Current configuration : 127 bytes
!
interface GigabitEthernet1/0/15
 switchport trunk encapsulation dot1q
 switchport mode trunk
 auto security-port uplink
end
```

This example shows how to configure a port as an auto-security port host.

```
Switch# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# int g1/0/18
Switch(config-if)# switchport mode access
Switch(config-if)# auto security-port host
Switch(config-if)# end
Switch#
```

Use the **auto security** and **show running-config** commands to confirm the prior configuration.

```
Switch# show auto security
Auto Security is Enabled globally

AutoSecure is Enabled on below interface(s):
-----
    GigabitEthernet1/0/2
    GigabitEthernet1/0/3
    GigabitEthernet1/0/15
    GigabitEthernet1/0/18
```

```
Switch# show run int g1/0/18
Building configuration...

Current configuration : 165 bytes
!
interface GigabitEthernet1/0/18
 switchport access vlan 20
 switchport mode access
 switchport voice vlan 40
 auto security-port host
 spanning-tree portfast
```

Disabling Auto Security Feature for Access (End Hosts) or Uplink Ports

Use the **no auto security-port** command to disable auto security on a port:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface interface	Enters interface mode
Step 3	Switch(config-if)# no auto security-port	Disables auto security on a port.
Step 4	Switch(config-if)# end	Exits to EXEC mode.
Step 5	Switch(config)# do show run int interface	Verifies auto security-port being disabled.
Step 6	Switch(config)# end	Returns to privileged EXEC mode.

This example shows how to disable auto security:

```
Switch# show run int g1/0/15
Building configuration...

Current configuration : 137 bytes
!
interface GigabitEthernet1/0/15
 switchport trunk encapsulation dot1q
```

```

switchport mode trunk
auto security-port uplink
end
Switch# conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# int g1/0/15
Switch(config-if)# no auto security-port
Switch(config-if)# end
Switch# show run int g1/0/15
Building configuration...

Current configuration : 110 bytes
!
interface GigabitEthernet1/0/15
switchport trunk encapsulation dot1q
switchport mode trunk
end

```

show command

Use the **show auto security** command, verify the status of auto-security on the interface and global level.

Use the **show auto security [configuration]** command, to view the CLIs that are applied with AS.

This example shows the output of the **show auto security** command when AS is enabled:

```

Switch# show auto security
Auto Security is Enabled globally
AutoSecurity is Enabled on below interface(s):
-----
GigabitEthernet2/0/2
GigabitEthernet2/0/3
GigabitEthernet2/0/4
GigabitEthernet2/0/5
GigabitEthernet2/0/6
GigabitEthernet2/0/7
GigabitEthernet2/0/8
GigabitEthernet2/0/9

```

This example shows the output of the **show auto security configuration** command when AS is enabled:

```

Switch# show auto security configuration
%AutoSecurity provides a single CLI config 'auto security'
to enable Base-line security Features like
DHCP snooping, ARP inspection and Port-Security
Auto Security CLIs applied globally:
-----
ip dhcp snooping
ip dhcp snooping vlan 2-1005
no ip dhcp snooping information option
ip arp inspection vlan 2-1005
ip arp inspection validate src-mac dst-mac ip

Auto Security CLIs applied on Access Port:
-----
switchport port-security
switchport port-security maximum 2
switchport port-security maximum vlan access 1
switchport port-security maximum vlan voice 1
switchport port-security violation restrict
switchport port-security aging time 2
switchport port-security aging type inactivity
ip arp inspection limit rate 100
ip dhcp snooping limit rate 100

```

```

Auto Security CLIs applied on Trunk Port:
-----
ip dhcp snooping trust
ip arp inspection trust
switchport port-security
switchport port-security maximum 100
switchport port-security violation restrict

```

Sample Output when Auto Security is Enabled

This example shows the output of the **show auto security** command when AS is enabled:

```

Switch# show auto security
Auto Security is Enabled globally

AutoSecure is Enabled on below interface(s):
-----
GigabitEthernet1/0/2
GigabitEthernet1/0/3
GigabitEthernet1/0/14

```

Sample Output when Auto Security is Disabled

This example shows the output of the **show auto security** command when AS is disabled:

```

Switch# show auto security
Auto Security is Disabled globally

AutoSecure is Enabled on below interface(s):
-----
none
Switch#

```

Guidelines and Restrictions

- The **auto security** command has no parameters.
- Base line security CLIs (like port security) are not individually nvgen'd on interfaces that have auto security-port configured. This allows you to maintain consistency over reboots.
- After auto security-port is enabled on a port, you cannot change the CLIs of the baseline security features (Port Security, DAI, and DHCP Snooping).

For example, if you enter the following:

```

interface GigabitEthernet2/0/24
switchport mode access
auto security-port host

```

The port security configuration is rejected on the auto security port:

```

Switch(config)# int g2/0/24
Switch(config-if)# switchport port-security maximum 4
%Command Rejected: 'auto security' enabled port

```

- Because you might need a different set of features on uplink ports, such as marking the port as a DHCP trusted port, you need to identify uplink and downlink ports and apply port mode specific configuration.

- Starting with Cisco IOS XE 3.6.0E (IOS 15.2.(2)E), all trunk ports are treated as uplink ports and all access port are treated as host ports.
- AS assumes that you will configure the port with data and voice VLANs.
- AS is not supported on routed or Layer 3 ports, dynamic ports, or VSL links.
- Enabling auto security should elicit system confirmation because the current baseline security configuration will be removed as the auto security configuration is applied. When auto security is globally enabled, existing configurations related to DAI, DHCP, and PSEC are removed and security violation may be triggered on the auto-security enabled port when incoming MACs exceed the limit.

When we issue **auto security** in global or interface config mode, any baseline security configuration on the interfaces or on the switch is removed and auto security configuration is applied. Disabling auto security does not restore the previous security configuration.



Configuring Control Plane Policing and Layer 2 Control Packet QoS



Note

CoPP is supported on Supervisor 6-E starting with Cisco IOS Release 12.2(50)SG; Supervisor 6L-E starting with Cisco IOS Release 12.2(52)X0; Supervisor Engine 7-E starting with Cisco IOS XE 3.1.0SG; Supervisor Engine 7L-E starting with Cisco IOS XE 3.2.0X0; Supervisor Engine 8-E starting with Cisco IOS XE 3.3.0SG; Supervisor 9-E starting with Cisco IOS XE 3.10.0E

This chapter contains information on how to protect your Catalyst 4500 series switch using control plane policing (CoPP). The information covered in this chapter is unique to the Catalyst 4500 series switches, and it supplements the network security information and procedures in [Chapter 62, “Configuring Network Security with ACLs.”](#) This information also supplements the network security information and procedures in these publications:

- *Cisco IOS Security Configuration Guide, Cisco IOS Release 12.4*, at this URL:
http://www.cisco.com/en/US/docs/ios/security/configuration/guide/12_4/sec_12_4_book.html
- *Cisco IOS Security Command Reference, Cisco IOS Release 12.4*, at this URL:
http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html

This chapter includes the following major sections:

- [Configuring Control Plane Policing, page 57-2](#)
- [Monitoring CoPP, page 57-9](#)
- [Configuring Layer 2 Control Packet QoS, page 57-11](#)
- [Configuring Layer 2 Control Packet QoS, page 57-11](#)
- [Policing IPv6 Control Traffic, page 57-16](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

Configuring Control Plane Policing

This section includes these topics:

- [About Control Plane Policing, page 57-2](#)
- [General Guidelines for Control Plane Policing, page 57-3](#)
- [Default Configuration, page 57-4](#)
- [Configuring CoPP for Control Plane Traffic, page 57-4](#)
- [Configuring CoPP for Data Plane and Management Plane Traffic, page 57-6](#)
- [Control Plane Policing Configuration Guidelines and Restrictions, page 57-8](#)
- [Policing IPv6 Control Traffic, page 57-16](#)

About Control Plane Policing



Note

Catalyst 4500 switch support hardware CoPP for all IPv6 First Hop Security Features (DHCPv6 Inspection/Guard, DHCPv6 remote-ID option for Layer 2, IPv6 full RA Guard, ...) However, due to inability of VFE to match ICMP v6 packets for policing in the outward direction, hardware CoPP does not work on Supervisor 6-E and Supervisor 6L-E, because “wireless” is supported only in Supervisor 8-E and 9-E, wireless related CoPP will work only on Supervisor 8-E and 9-E in wireless mode.

The control plane policing (CoPP) feature increases security on the Catalyst 4500 series switch by protecting the CPU from unnecessary or DoS traffic and giving priority to important control plane and management traffic. The classification TCAM and QoS policers provide CoPP hardware support.

Traffic managed by the CPU is divided into three functional components or *planes*:

- Data plane
- Management plane
- Control plane

You can use CoPP to protect most of CPU-bound traffic and to ensure routing stability, reachability, and packet delivery. Most importantly, you can use CoPP to protect the CPU from a DoS attack.

By default, you receive a list of predefined ACLs matching a selected set of Layer 2 and Layer 3 control plane packets. You can further define your preferred policing parameters for each of these packets and modify the matching criteria of these ACLs.

The following table lists the predefined ACLs.

Predefined Named ACL	Description
system-cpp-dot1x	MAC DA = 0180.C200.0003
system-cpp-lddp	MAC DA = 0180.C200.000E
system-cpp-mcast-cfm	MAC DA = 0100.0CCC.CCCC - 0100.0CCC.CCCC7
system-cpp-ucast-cfm	MAC DA = 0100.0CCC.CCCC
system-cpp-bpdu-range	MAC DA = 0180.C200.0000 - 0180.C200.000F
system-cpp-cdp	MAC DA = 0100.0CCC.CCCC (UDLD/DTP/VTP/Pagp)

Predefined Named ACL	Description
system-cpp-sstp	MAC DA = 0100.0CCC.CCCD
system-cpp-cgmp	MAC DA = 01.00.0C.DD.DD.DD
system-cpp-hsrpv2	IP Protocol = UDP, IPDA = 224.0.0.102
system-cpp-ospf	IP Protocol = OSPF, IP DA matches 224.0.0.0/24
system-cpp-igmp	IP Protocol = IGMP, IP DA matches 224.0.0.0/3
system-cpp-pim	IP Protocol = PIM, IP DA matches 224.0.0.0/24
system-cpp-all-systems-on-subnet	IP DA = 224.0.0.1
system-cpp-all-routers-on-subnet	IP DA = 224.0.0.2
system-cpp-ripv2	IP DA = 224.0.0.9
system-cpp-ip-mcast-linklocal	IP DA = 224.0.0.0/24
system-cpp-dhcp-cs	IP Protocol = UDP, L4SrcPort = 68, L4DstPort = 67
system-cpp-dhcp-sc	IP Protocol = UDP, L4SrcPort = 67, L4DstPort = 68
system-cpp-dhcp-ss	IP Protocol = UDP, L4SrcPort = 67, L4DstPort = 67

For the data and management plane traffic, you can define your own ACLs to match the traffic class that you want to police.

CoPP uses MQC to define traffic classification criteria and to specify the configurable policy actions for the classified traffic. MQC uses class maps to define packets for a particular traffic class. After you have classified the traffic, you can create policy maps to enforce policy actions for the identified traffic. The **control-plane global** configuration command allows you to directly attach a CoPP service policy to the control plane.

The policy map `system-cpp-policy` must contain the predefined class maps in the predefined order at the beginning of the policy map. The policy map `system-cpp-wireless-policy` must contain the predefined class-maps in predefined order; adding user defined class-map to this policy-map is dis-advised. We recommend that you use the global macro `system-cpp` to create `system-cpp-policy` and `system-cpp-wireless-policy` policy maps.

The `system-cpp-policy` policy map contains the predefined class maps for the control plane traffic. The names of all system-defined CoPP class maps and their matching ACLs contain the prefix `system-cpp-`. By default, no action is specified for each traffic class. You can define your own class maps matching CPU-bound data plane and management plane traffic. You can also add your defined class maps to `system-cpp-policy`.

General Guidelines for Control Plane Policing

Guidelines for control plane policing include the following:

- If a given traffic class does not have a designated class map, and you want to protect this traffic, we recommend that you:
 - Create specific class maps for such unknown traffic packets and add the user-defined class maps to **system-cpp-policy**.
 - Or, rate-limit such traffic to prevent CPU hogging.

For instance, in a VSS setup, if you have defined class map *cpp-vsl-mgmt* for VSL management traffic (exclusively Layer 2 packets), do not use the *cpp-vsl-mgmt* class map to protect supervisor keep-alive traffic (IP packets), or BFD packets. This can cause VSL link failures. Instead, create separate class maps, such as *cpp-ip* for supervisor keep-alive traffic, and *cpp-bfd* for BFD packets. VSL link failures may also ensue if you enter *class-default* as the class name for traffic that does not have a designated class map.

- Port security might cancel the effect of CoPP for non-IP control packets.

Although source MAC learning on a Catalyst 4500 series switch is performed in software, learning control packets' source MAC addresses (for example, IEEE BPDU, CDP, SSTP BPDU, GARP/) is not allowed. After you configure port security on a port where you expect a high rate of potentially unanticipated control packets, the system generates a copy of the packet to the CPU (until the source address is learned), instead of forwarding it.

The current architecture of the Catalyst 4500 supervisor engine does not allow you to apply policing on the copy of packets sent to the CPU. You can only apply policing on packets that are forwarded to the CPU. Copies of packets are sent to the CPU at the same rate as control packets, and port security is not triggered because learning from control packets is not allowed. Policing is not applied because the packet copy, not the original, is sent to the CPU.

- ARP policing is not supported on either the classic series supervisor engines (i.e., supervisor engines prior to Supervisor Engine 7-E) or fixed configuration switches. It is supported on Supervisor Engine 6-E, and Supervisor Engine 6L-E (use “match protocol arp” to classify).
- Only ingress CoPP is supported; control-plane related CLIs support only the **input** keyword.
- Use ACLs and class-maps to identify data plane and management plane traffic that are handled by CPU.
- “police” is the only action supported in CoPP policy-map.
- Avoid using the **log** keyword in the CoPP policy ACLs.

Default Configuration

CoPP is disabled by default.

Configuring CoPP for Control Plane Traffic

To configure CoPP for control plane traffic, perform this task:

	Command	Purpose
Step 1	Switch# config terminal	Enters global configuration mode.
Step 2	Switch(config)# qos	(Optional) Enables QoS globally.
Step 3	Switch(config)# macro global apply system-cpp	(Optional) Creates the system-cpp-policy policy map and attaches it to the control plane.

	Command	Purpose
Step 4	<pre>Switch(config)# policy-map system-cpp-policy Switch(config-pmap)# class {system-cpp-dot1x system-cpp-bpdu-range system-cpp-cdp service system-cpp-sstp system-cpp-cgmp system-cpp-ospf system-cpp-igmp system-cpp-pim system-cpp-all-systems-on-subnet system-cpp-all-routers-on-subnet system-cpp-ripv2 system-cpp-hsrpv2 system-cpp-ip-mcast-linklocal system-cpp-dhcp-cs system-cpp-dhcp-sc system-cpp-dhcp-ss} Switch(config-pmap-c)# police [aggregate name] rate burst [conform-action {drop transmit}] [{exceed-action {drop transmit}}}]</pre>	Associates actions to one or multiple system-defined control plane traffic in the service policy map. Repeat this step if necessary.
Step 5	<pre>Switch# show policy-map system-cpp-policy</pre>	(Optional) Verifies the configuration.

The following example shows how to police CDP packets:

```
Switch# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# qos
Switch(config)# macro global apply system-cpp
Switch(config)# policy-map system-cpp-policy
Switch(config-pmap)# class system-cpp-cdp
Switch(config-pmap-c)# police 32000 1000 conform-action transmit exceed-action drop
Switch(config-pmap-c)# end
Switch# show policy-map system-cpp-policy
Policy Map system-cpp-policy
Class system-cpp-dot1x
Class system-cpp-bpdu-range
* Class system-cpp-cdp
  police 32000 bps 1000 byte conform-action transmit exceed-action drop
Class system-cpp-sstp
Class system-cpp-cgmp
Class system-cpp-ospf
Class system-cpp-hsrpv2
Class system-cpp-igmp
Class system-cpp-pim
Class system-cpp-all-systems-on-subnet
Class system-cpp-all-routers-on-subnet
Class system-cpp-ripv2
Class system-cpp-ip-mcast-linklocal
Class system-cpp-dhcp-cs
Class system-cpp-dhcp-sc
Class system-cpp-dhcp-ss
Switch#
```

Configuring CoPP for Data Plane and Management Plane Traffic

To configure CoPP for data plane and management plane traffic, perform this task:

	Command	Purpose
Step 1	Switch(config)# qos	(Optional) Enables QoS globally.
Step 2	Switch(config)# macro global apply system-cpp	(Optional) Attaches the system-cpp-policy policy map to the control plane and the system-cpp-wireless-policy policy to the wireless control plane.
Step 3	<p>Switch(config)# {ip mac} access-list extended {access-list-name}</p> <p>For an ip access list, issue Switch(config-ext-nacl)#{permit deny} {protocol} source {source-wildcard} destination {destination-wildcard}</p> <p>For a mac access list, issue Switch(config-ext-macl)#{permit deny} source {source-wildcard} destination {destination-wildcard} [protocol-family]</p> <p>OR</p> <p>Switch(config)# access-list {access-list-name} {permit deny} {type-code wild-mask address mask}</p>	<p>Defines ACLs to match traffic:</p> <ul style="list-style-type: none"> permit—Sets the conditions under which a packet passes a named ACL deny—Sets the conditions under which a packet does not pass a name ACL <p>Note You must configure ACLs in most cases to identify the important or unimportant traffic.</p> <ul style="list-style-type: none"> type-code—16-bit hexadecimal number written with a leading 0x; for example, 0x6000. Specify either a Link Service Access Point (LSAP) type code for 802-encapsulated packets or a SNAP type code for SNAP-encapsulated packets. (LSAP, sometimes called SAP, refers to the type codes found in the DSAP and SSAP fields of the 802 header.) wild-mask—16-bit hexadecimal number whose ones bits correspond to bits in the type-code argument. The wild-mask indicates which bits in the type-code argument should be ignored when making a comparison. (A mask for a DSAP/SSAP pair should always be 0x0101 because these two bits are used for purposes other than identifying the SAP code.) address—48-bit Token Ring address written as a dotted triple of four-digit hexadecimal numbers. This field is used for filtering by vendor code. mask—48-bit Token Ring address written as a dotted triple of four-digit hexadecimal numbers. The ones bits in the mask are the bits to be ignored in address. This field is used for filtering by vendor code.

	Command	Purpose
Step 4	Switch(config)# class-map { <i>traffic-class-name</i> } Switch(config-cmap)# match access-group { <i>access-list-number</i> <i>name</i> { <i>access-list-name</i> }}	Defines the packet classification criteria. To identify the traffic associated with the class, use the match statements.
Step 5	Switch(config-cmap)# exit	Returns to global configuration mode.
Step 6	Switch(config)# policy-map system-cpp-policy Switch(config-pmap)# class { <i>class-map-name</i> } Switch(config-pmap-c)# police [<i>aggregate name</i>] <i>rate burst</i> [conform-action { <i>drop</i> <i>transmit</i> }] [exceed-action { <i>drop</i> <i>transmit</i> }]	Adds the traffic classes to the CoPP policy map. Uses the police statement to associate actions to the traffic class.
Step 7	Switch(config)# end	Returns to privileged EXEC mode.
Step 8	Switch# show policy-map system-cpp-policy	Verifies your entries.

The following example shows how to configure trusted hosts with source addresses 10.1.1.1 and 10.1.1.2 to forward Telnet packets to the control plane without constraint, while allowing all remaining Telnet packets to be policed at the specific rate. This example assumes that global QoS is enabled and that the system-cpp-policy policy map was created.

```
Switch# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# qos
Switch(config)# macro global apply system-cpp

! Allow 10.1.1.1 trusted host traffic.
Switch(config)# access-list 140 deny tcp host 10.1.1.1 any eq telnet

! Allow 10.1.1.2 trusted host traffic.
Switch(config)# access-list 140 deny tcp host 10.1.1.2 any eq telnet

! Rate limit all other Telnet traffic.
Switch(config)# access-list 140 permit tcp any any eq telnet

! Define class-map "telnet-class."
Switch(config)# class-map telnet-class
Switch(config-cmap)# match access-group 140
Switch(config-cmap)# exit

! Add the class-map "telnet-class" to "system-cpp-policy" and define the proper action
Switch(config)# policy-map system-cpp-policy
Switch(config-pmap)# class telnet-class
Switch(config-pmap-c)# police 80000 1000 conform transmit exceed drop
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit

! Verify the above configuration steps
Switch# show policy-map system-cpp-policy
Policy Map system-cpp-policy
  Class system-cpp-dot1x
  Class system-cpp-bpdu-range
  Class system-cpp-cdp
    police 32000 bps 1000 byte conform-action transmit exceed-action drop
  Class system-cpp-sstp
```

```

Class system-cpp-cgmp
Class system-cpp-ospf
Class system-cpp-hsrpv2
Class system-cpp-igmp
Class system-cpp-pim
Class system-cpp-all-systems-on-subnet
Class system-cpp-all-routers-on-subnet
Class system-cpp-ripv2
Class system-cpp-ip-mcast-linklocal
Class system-cpp-dhcp-cs
Class system-cpp-dhcp-sc
Class system-cpp-dhcp-ss
* Class telnet-class
  police 80000 1000 byte conform-action drop exceed-action drop

```

Control Plane Policing Configuration Guidelines and Restrictions

When using (or configuring) control plane policing, consider these guidelines and restrictions:

All supervisor engines

When configuring CoPP, consider these guidelines:

- Only ingress CoPP is supported. Only the **input** keyword is supported in control plane-related CLIs.
- Control plane traffic can be policed only through CoPP. Traffic cannot be policed at the input interface or VLAN even though a policy map containing the control plane traffic is accepted when the policy map is attached to an interface or VLAN.
- Use ACLs and class maps to identify data plane and management plane traffic that are handled by the CPU. User defined class maps should be added to the system-cpp-policy policy map for CoPP.
- The default system-cpp-policy policy map does not define actions for the system-defined class maps (no policing).
- The only action supported in system-cpp-policy is police.
- You can use both MAC and IP ACLs to define data plane and management plane traffic classes. However, if a packet also matches a predefined ACL for the control plane traffic, a police (or no police) action will operate on the control plane class because the control plane classes appear above the user-defined classes in the service policy.
- The exceeding action **policed-dscp-transmit** is not supported for CoPP.
- Do not use the **log** keyword in CoPP policy ACLs. Instead, if you want to determine if rogue packets are arriving, view the output of the **show policy-map interface** command or use the span feature.

Do not apply to Supervisor Engine 6-E, and Supervisor Engine 6L-E

- To police control plane traffic, use the system-defined class maps.
- System-defined class maps cannot be used in policy maps for regular QoS.
- The policy map named system-cpp-policy is dedicated for CoPP.
- CoPP is not enabled unless global QoS is enabled and a police action is specified.

Do not apply to Supervisor Engine 6-E, Supervisor Engine 6L-E, Supervisor 7-E and Supervisor 7L-E

- System-cpp-wireless-policy is generated only on Supervisor Engine 8-E and 9-E when booted in wireless mode.

Monitoring CoPP

You can enter the **show policy-map control-plane** command to develop site-specific policies, to monitor statistics for the control plane policy, and to troubleshoot CoPP. This command displays dynamic information about the actual policy applied, including rate information and the number of bytes (and packets) that conformed or exceeded the configured policies both in hardware and in software.

The output of the **show policy-map control-plane** command is similar to the following:

```
Switch# show policy-map control-plane

Control Plane

Service-policy input: system-cpp-policy

  Class-map: system-cpp-dot1x (match-all)
    0 packets
    Match: access-group name system-cpp-dot1x

  Class-map: system-cpp-bpdu-range (match-all)
    0 packets
    Match: access-group name system-cpp-bpdu-range

*   Class-map: system-cpp-cdp (match-all)
    160 packets
    Match: access-group name system-cpp-cdp
**  police: Per-interface
    Conform: 22960 bytes Exceed: 0 bytes
*

  Class-map: system-cpp-sstp (match-all)
    0 packets
    Match: access-group name system-cpp-sstp

  Class-map: system-cpp-cgmp (match-all)
    0 packets
    Match: access-group name system-cpp-cgmp

  Class-map: system-cpp-hsrpv2 (match-all)
    0 packets
    Match: access-group name system-cpp-hsrpv2

  Class-map: system-cpp-ospf (match-all)
    0 packets
    Match: access-group name system-cpp-ospf

  Class-map: system-cpp-igmp (match-all)
    0 packets
    Match: access-group name system-cpp-igmp

  Class-map: system-cpp-pim (match-all)
    0 packets
    Match: access-group name system-cpp-pim

  Class-map: system-cpp-all-systems-on-subnet (match-all)
    0 packets
```

```

Match: access-group name system-cpp-all-systems-on-subnet

Class-map: system-cpp-all-routers-on-subnet (match-all)
  0 packets
Match: access-group name system-cpp-all-routers-on-subnet

Class-map: system-cpp-ripv2 (match-all)
  0 packets
Match: access-group name system-cpp-ripv2

Class-map: system-cpp-ip-mcast-linklocal (match-all)
  0 packets
Match: access-group name system-cpp-ip-mcast-linklocal

Class-map: system-cpp-dhcp-cs (match-all)
  83 packets
Match: access-group name system-cpp-dhcp-cs

Class-map: system-cpp-dhcp-sc (match-all)
  0 packets
Match: access-group name system-cpp-dhcp-sc

Class-map: system-cpp-dhcp-ss (match-all)
  0 packets
Match: access-group name system-cpp-dhcp-ss

Class-map: telnet-class (match-all)
  92 packets
Match: access-group 140
police:
  cir 32000 bps, bc 1500 bytes
  conformed 5932 bytes; actions:
    transmit
  exceeded 0 bytes; actions:
    drop
  conformed 0000 bps, exceed 0000 bps

Class-map: class-default (match-any)
  0 packets
Match: any
  0 packets
Switch#

```

To clear the counters on the control plane, enter the **clear control-plane *** command:

```

Switch# clear control-plane *
Switch#

```

To display all the CoPP access list information, enter the **show access-lists** command:

```

Switch# show access-lists
Extended IP access list system-cpp-all-routers-on-subnet
10 permit ip any host 224.0.0.2
Extended IP access list system-cpp-all-systems-on-subnet
10 permit ip any host 224.0.0.1
Extended IP access list system-cpp-dhcp-cs
10 permit udp any eq bootpc any eq bootps Extended IP access list
system-cpp-dhcp-sc
10 permit udp any eq bootps any eq bootpc Extended IP access list
system-cpp-dhcp-ss
10 permit udp any eq bootps any eq bootps Extended IP access list
system-cpp-igmp
10 permit igmp any 224.0.0.0 31.255.255.255 Extended IP access list
system-cpp-ip-mcast-linklocal

```



```
10 permit ip any 224.0.0.0 0.0.0.255 Extended IP access list
system-cpp-ospf
10 permit ospf any 224.0.0.0 0.0.0.255 Extended IP access list
system-cpp-pim
10 permit pim any 224.0.0.0 0.0.0.255 Extended IP access list
system-cpp-ripv2
10 permit ip any host 224.0.0.9
Extended MAC access list system-cpp-bpdu-range
permit any 0180.c200.0000 0000.0000.000f Extended MAC access list
system-cpp-cdp
permit any host 0100.0ccc.cccc
Extended MAC access list system-cpp-cgmp
permit any host 0100.0cdd.dddd
Extended MAC access list system-cpp-dot1x
permit any host 0180.c200.0003
system-cpp-sstp
permit any host 0100.0ccc.cccd
```

To display one CoPP access list, enter the **show access-lists system-cpp-cdp** command:

```
Switch# show access-list system-cpp-cdp
Extended MAC access list system-cpp-cdp
permit any host 0100.0ccc.cccc
Switch#
```

Configuring Layer 2 Control Packet QoS

Layer 2 control packet QoS enables you to police control packets arriving on a physical port or LAN.

This section includes these topics:

- [Understanding Layer 2 Control Packet QoS, page 57-11](#)
- [Default Configuration, page 57-12](#)
- [Enabling Layer 2 Control Packet QoS, page 57-12](#)
- [Disabling Layer 2 Control Packet QoS, page 57-13](#)
- [Layer 2 Control Packet QoS Configuration Examples, page 57-14](#)
- [Layer 2 Control Packet QoS Guidelines and Restrictions, page 57-16](#)

Understanding Layer 2 Control Packet QoS

You might want to police incoming Layer 2 control packets such as STP, CDP, VTP, SSTP, BPDU, EAPOL and LLDP on a specific port before the packets reach CPU. This could serve as a first line of defense before aggregate traffic is subjected to policing (through CoPP). By default, policers cannot be applied to Layer 2 control packets in the input direction. This prevents users from inadvertently policing or dropping critical Layer 2 control packets.

While this approach protects a user who is wrongly policing control packets, it introduces a more serious problem. If a flood of Layer 2 control packets is received on any of the switch interfaces at a very high rate due to a DoS attack or to a loop introduced in the customer network because of misconfiguration, CPU utilization can increase quickly. This can have adverse impacts such as loss of protocol keep-alives and routing protocol updates. The Layer 2 control packet QoS feature allows you to police Layer 2 control packets at the port, VLAN, or port- VLAN level in the input direction.

Default Configuration

Layer 2 control packet QoS is disabled by default.

Enabling Layer 2 Control Packet QoS

To enable Layer 2 control packet QoS, perform this task:

	Command	Purpose
Step 1	Switch# config terminal	Enters configuration mode.
Step 2	Switch(config)# [no] qos control-packets [bpd-range cdp-vtp eapol sstp protocol-tunnel llbp]	Enables QoS on all or a specific packet type. Use the no keyword to disable QoS on all or a specific packet type.
Step 3	Switch(config)# end	Exits configuration mode.
Step 4	Switch# show run inc qos control-packets	Verifies the configuration.

Table 57-1 lists the types of packets impacted by this feature.

Table 57-1 Packet Type and Actionable Address Range

Type of Packet that the Feature is Enabled On	Range of Address the Feature Acts On
BPDURange	0180.C200.0000 BPDURange 0180.C200.0002 OAM, LACP 0180.C200.0003 EAPOL
CDP-VTP	0100.0CCC.CCCC
SSTP	0100.0CCC.CCCD
LLDP	0180.C200.000E

The following example shows how to enable QoS for CDP packets and to apply a policer to CDP packets arriving on interface gi3/1 and VLAN 1:

```
Switch# config terminal
Switch(config)# qos control-packets cdp-vtp
Switch(config)# end
Switch# show run | inc qos control-packets
qos control-packets cdp-vtp
Switch# show class-map
Class Map match-any system-control-packet-cdp-vtp (id 1)

      Match access-group name system-control-packet-cdp-vtp

Create a policy map and attach it to interface gi3/1 , vlan 1
Switch# config terminal
Switch(config)# policy-map police_cdp
Switch(config-pmap)# class system-control-packet-cdp-vtp
Switch(config-pmap-c)# police 32k
Switch(config-pmap-c)# end

Switch(config)# interface gi3/1
```

```

Switch(config-if)# vlan 1
Switch(config-if-vlan-range)# service-policy in police_cdp
Switch(config-if-vlan-range)# exit
Switch(config-if)# exit
Switch(config)# exit
Switch# show policy-map interface gi3/1

GigabitEthernet3/1 vlan 1

Service-policy input: police_cdp

Class-map: system-control-packet-cdp-vtp (match-any)
  0 packets
  Match: access-group name system-control-packet-cdp-vtp
    0 packets
  police:
    cir 32000 bps, bc 1500 bytes
    conformed 0 packets, 0 bytes; actions:
      transmit
    exceeded 0 packets, 0 bytes; actions:
      drop
    conformed 0000 bps, exceed 0000 bps

Class-map: class-default (match-any)
  0 packets

```

Disabling Layer 2 Control Packet QoS

The **no qos control-packet** command disables QoS for all packet types.

The following example shows how to disable QoS for CDP packets after QoS is enabled for all packet types:

```

Switch# show running-configuration | include qos control-packets
qos control-packets bpdu-range
qos control-packets cdp-vtp
qos control-packets llDP
qos control-packets eapOl
qos control-packets sstp
qos control-packets protocol-tunnel

```



Note

When all control packets (CDP/VTP, bpdu-range, Sstp, LLDP, and protocol-tunnel), are enabled only qos control-packets is nevgen'd. Individual protocol names mentioned in the previous output are nvegen'd only if the some of the control packets are configured.

```

Switch# config terminal
Switch(config)# no qos control-packets cdp-vtp
Switch(config)# end
Switch# show running-configuration | include qos control-packets
qos control-packets bpdu-range
qos control-packets llDP
qos control-packets sstp
qos control-packets protocol-tunnel

```



Note

When you unconfigure this feature for a specified protocol type, the user-configured policies handling that protocol type immediately become ineffective. To save TCAM resources, remove the policies as well as MACs and class maps (auto-generated or user-defined).

**Note**

TCAM resources are not consumed when the interface is in a down state.

Table 57-2 displays the auto-generated MACLs and class maps that are created when you enable the feature on the corresponding packet type.

Table 57-2 Packet Types and Auto-Generated MACL/Class Maps

Packet Type	Auto-Generated MACL/Class Map
BPDU-range	mac access-list extended system-control-packet-bpdu-range permit any 0180.c200.0000 0000.0000.000c class-map match-any system-control-packet-bpdu-range match access-group name system-control-packet-bpdu-range
SSTP	mac access-list extended system-control-packet-sstp permit any host 0100.0ccc.cccd class-map match-any system-control-packet-sstp match access-group name system-control-packet-sstp
CDP-VTP	mac access-list extended system-control-packet-cdp-vtp permit any host 0100.0ccc.cccc class-map match-any system-control-packet-cdp-vtp match access-group name system-control-packet-cdp-vtp
EAPOL	mac access-list extended system-control-packet-eapol permit any any 0x888E class-map match-any system-control-packet-eapol match access-group name system-control-packet-eapol
LLDP	mac access-list extended system-control-packet-lldp permit any host 0180.c200.000e class-map match-any system-control-packet-lldp match access-group name system-control-packet-lldp
PROTOCOL TUNNEL	mac access-list extended system-control-packet-protocol-tunnel permit any host 0100.0ccd.cdd0 class-map match-any system-control-packet-protocol-tunnel match access-group name system-control-packet-protocol-tunnel

Layer 2 Control Packet QoS Configuration Examples

You can use CoPP and Layer 2 control packet QoS together to prevent DoS attacks to the CPU. In the following example, BPDUs arriving on interface gi3/1, VLAN 1 and VLAN 2 are limited to 32 Kbps and 34 Kbps, respectively. Aggregate BPDU traffic to CPU then is further rate-limited to 50 Kbps using CoPP.

```
Switch(config)# qos control-packets
```

```
Switch(config)# policy-map police_bpdu_1
Switch(config-pmap)# class system-control-packet-bpdu-range
Switch(config-pmap-c)# police 32k 1000
Switch(config-pmap-c-police)# exit
Switch(config-pmap-c)# exit
Switch(config-pmap)# policy-map police_bpdu_2
Switch(config-pmap)# class system-control-packet-bpdu-range
Switch(config-pmap-c)# police 34k
Switch(config-pmap-c-police)# exit
```

Configuring Layer 2 Control Packet QoS

```
Switch(config)# interface gi3/1
Switch(config-if)# vlan-range 1
Switch(config-if-vlan-range)# service-policy in police_bpdu_1
Switch(config-if-vlan-range)# exit
Switch(config-if)# interface gi3/2
Switch(config-if)# vlan-range 2
Switch(config-if-vlan-range)# service-policy in police_bpdu_1
Switch(config-if-vlan-range)# exit
```

Configuring Control Plane Policy

```
Switch(config)# macro global apply system-cpp
Switch(config)# policy-map system-cpp-policy
Switch(config-pmap)# class system-cpp-bpdu-range
Switch(config-pmap-c)# police 50k
Switch(config-pmap-c-police)# exit
Switch(config-pmap-c)# exit
```



Note

To reduce the consumption of policer resources, you can also use named-aggregate policers applied to a group of ports or VLANs.



Note

Do not modify class maps and MACLs that are auto-generated by the system. This action can cause unexpected behavior when the switch reloads or when the running configuration is updated from a file.

To refine or modify system-generated class maps or MACLs, apply user-defined class maps and MACLs.



Note

User defined class map names must begin with the prefix system-control-packet. If not, certain hardware (Supervisor Engine II-Plus, Supervisor Engine II+10GE, Supervisor Engine V, and Supervisor Engine V-10GE) might not perform the configured QoS action.

For example, the following are valid user-defined class map names to police Layer 2 control packets because they begin with the prefix system-control-packet:

```
system-control-packet-bpdu1
system-control-packet-control-packet
```

No such restrictions exist on the names you can use for user-defined MACLs (access-groups).

The following example shows how to create user-defined MACLs and class maps to identify EAPOL and BPDU packets. Because the auto-generated class map system-control-packet-bpdu range matches three packet types (BPDU, EAPOL, and OAM), policing this traffic class affects all three packet types. To police BPDU and EAPOL packets at different rates, you can set user-defined MACL and class map as follows:

```
Switch(config)# mac access-list extended system-control-packet-bpdu
Switch(config-ext-macl)# permit any host 0180.c200.0000
Switch(config-ext-macl)# exit
```

```

Switch(config)# class-map match-any system-control-packet-bpdu
Switch(config-cmap)# match access-group name system-control-packet-bpdu
Switch(config-cmap)# exit

Switch(config)# mac access-list extended system-control-packet-eapol
Switch(config-ext-macl)# permit any host 0180.c200.0003
Switch(config-ext-macl)# exit
Switch(config)# class-map match-any system-control-packet-eapol
Switch(config-cmap)# match access-group name system-control-packet-eapol
Switch(config-cmap)# exit

```

Layer 2 Control Packet QoS Guidelines and Restrictions

When using (or configuring) Layer 2 control packet QoS, consider these guidelines and restrictions:

- When you enable Layer 2 control packet QoS, it applies to all ports on the switch. If Layer 2 control packets are not explicitly classified in the policy attached to port or VLAN, the actions in class-default will be applied as per normal QoS rules.
- Place classifiers that match control packets at the beginning of a policy map followed by other traffic classes, ensuring that Layer 2 control packets are not subjected to inadvertent QoS actions.
- The application of default class (class-default) actions depends on the type of supervisor engine:
 - Supervisor Engine V-10GE with NetFlow support—Actions associated with class-default are never applied on unmatched control packets; a default permit action is applied. Only actions associated with class maps that begin with system-control-packet are applied on control packets.
 - All other supervisor engines—Actions associated with class-default are applied on unmatched control packets.
- If you enable the feature on a BPDU range, EAPOL packets are policed only after the initial 802.1X authentication phase completes.

Policing IPv6 Control Traffic

On Supervisor Engine 6-E, and Supervisor Engine 6L-E, IPv6 control packets such as OSPF, PIM and MLD can be policed on a physical port, VLAN, or control plane by configuring IPv6 ACLs to classify such traffic and then applying a QoS policy to police such traffic.

The following examples show how to police OSPFv6, PIMv6 and MLD control traffic received on a port.

This example shows how to configure a traffic class to identify OSPFv6 control packets by its destination IP v6 address:

```

Switch# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# ipv6 access-list ospfv6
Switch(config-ipv6-acl)# permit ipv6 any host ff02:5
Switch(config-ipv6-acl)# exit
Switch(config)# class-map ospfv6Class
Switch(config-camp)# match access-group name ospfv6
Switch(config-camp)# exit

```

The following example shows how to configure a traffic class to identify PIMv6 control packets by its destination IPv6 address:

```

Switch(config)# ipv6 access-list pimv6

```

```
Switch(config-ipv6-acl)# permit ipv6 any host ff02::d
Switch(config-ipv6-acl)# exit
Switch(config)# class-map pimv6Class
Switch(config-cmap)# match access-group name pimv6
Switch(config-cmap)# exit
```

The following example shows how to configure a traffic class to identify MLD protocol control packets:

```
Switch(config)# ipv6 access-list mldv1
Switch(config-ipv6-acl)# permit icmp any any mld-query
Switch(config-ipv6-acl)# permit icmp any any mld-report
Switch(config-ipv6-acl)# permit icmp any any mld-reduction
Switch(config-ipv6-acl)# exit
Switch(config)# class-map mldClass
Switch(config-cmap)# match access-group name mldv1
Switch(config-cmap)# exit
```

The following example shows how to configure a QoS policy to police OSPFv6, PIMv6 and MLD traffic classes:

```
Switch(config)# policy-map v6_control_packet_policy
Switch(config-pmap)# class mldClass
Switch(config-pmap-c)# police 32k
Switch(config-pmap-c-police)# class ospfv6Class
Switch(config-pmap-c)# police 32k
Switch(config-pmap-c)# class pimv6Class
Switch(config-pmap-c)# police 32k
Switch(config-pmap-c-police)# exit
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# exit
Switch# show policy-map
```

```
Policy Map v6_control_packet_policy
  Class mldClass
    police cir 32000 bc 1500
      conform-action transmit
      exceed-action drop
  Class ospfv6Class
    police cir 32000 bc 1500
      conform-action transmit
      exceed-action drop
  Class pimv6class
    police cir 32000 bc 1500
      conform-action transmit
      exceed-action drop
```

The following example shows how to policy to interface gi2/2 in the input direction:

```
Switch# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface gi2/2
Switch(config-if)# service-policy in v6_control_packet_policy
Switch(config-if)# exit
```




Configuring Dynamic ARP Inspection

This chapter describes how to configure Dynamic ARP Inspection (DAI) on the Catalyst 4500 series switch.

This chapter includes the following major sections:

- [About Dynamic ARP Inspection, page 58-1](#)
- [Configuring Dynamic ARP Inspection, page 58-5](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About Dynamic ARP Inspection

Dynamic ARP Inspection (DAI) is a security feature that validates Address Resolution Protocol (ARP) packets in a network. DAI allows a network administrator to intercept, log, and discard ARP packets with invalid MAC-IP pairs. This capability protects the network from certain “man-in-the-middle” attacks.

This section contains the following subsections:

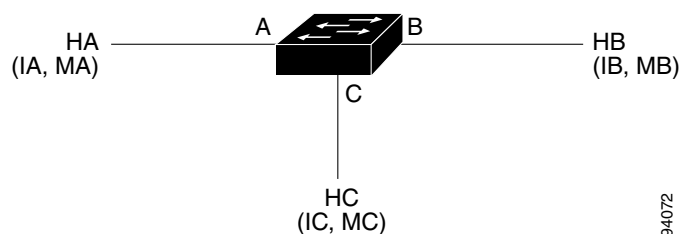
- [ARP Cache Poisoning, page 58-2](#)
- [Purpose of Dynamic ARP Inspection, page 58-2](#)
- [Interface Trust State, Security Coverage and Network Configuration, page 58-3](#)
- [Relative Priority of Static Bindings and DHCP Snooping Entries, page 58-4](#)
- [Logging of Dropped Packets, page 58-4](#)
- [Rate Limiting of ARP Packets, page 58-4](#)
- [Port Channels Function, page 58-5](#)

ARP Cache Poisoning

You can attack hosts, switches, and routers connected to your Layer 2 network by “poisoning” their ARP caches. For example, a malicious user might intercept traffic intended for other hosts on the subnet by poisoning the ARP caches of systems connected to the subnet.

Figure 58-1 shows an example of cache poisoning.

Figure 58-1 ARP Cache Poisoning



Hosts HA, HB, and HC are connected to the switch on interfaces A, B and C, all of which are on the same subnet. Their IP and MAC addresses are shown in parentheses; for example, Host HA uses IP address IA and MAC address MA. When HA needs to communicate to HB at the IP Layer, HA broadcasts an ARP request for the MAC address associated with IB. As soon as HB receives the ARP request, the ARP cache on HB is populated with an ARP binding for a host with the IP address IA and a MAC address MA. When HB responds to HA, the ARP cache on HA is populated with a binding for a host with the IP address IB and a MAC address MB.

Host HC can “poison” the ARP caches of HA and HB by broadcasting forged ARP responses with bindings for a host with an IP address of IA (or IB) and a MAC address of MC. Hosts with poisoned ARP caches use the MAC address MC as the destination MAC address for traffic intended for IA or IB. This means that HC intercepts that traffic. Because HC knows the true MAC addresses associated with IA and IB, HC can forward the intercepted traffic to those hosts using the correct MAC address as the destination. HC has inserted itself into the traffic stream from HA to HB, the classic “man in the middle” attack.

Purpose of Dynamic ARP Inspection

To prevent ARP poisoning attacks, a switch must ensure that only valid ARP requests and responses are relayed. DAI prevents these attacks by intercepting all ARP requests and responses. Each of these intercepted packets is verified for valid MAC address to IP address bindings before the local ARP cache is updated or the packet is forwarded to the appropriate destination. Invalid ARP packets are dropped.

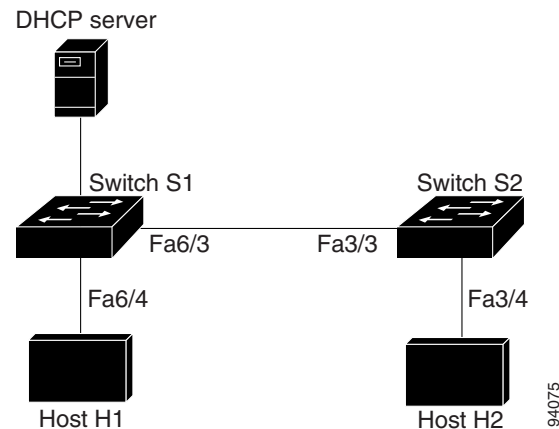
DAI determines the validity of an ARP packet based on valid MAC address to IP address bindings stored in a trusted database. This database is built at runtime by DHCP snooping, provided this feature is enabled on VLANs and on the switch. In addition, in order to handle hosts that use statically configured IP addresses, DAI can also validate ARP packets against user-configured ARP ACLs.

DAI can also be configured to drop ARP packets when the IP addresses in the packet are invalid or when the MAC addresses in the body of the ARP packet do not match the addresses specified in the Ethernet header.

Interface Trust State, Security Coverage and Network Configuration

DAI associates a trust state with each interface on the system. Packets arriving on trusted interfaces bypass all DAI validation checks, while those arriving on untrusted interfaces go using the DAI validation process. In a typical network configuration for DAI, all ports connected to host ports are configured as untrusted, while all ports connected to switches are configured as trusted. With this configuration, all ARP packets entering the network from a given switch pass the security check.

Figure 58-2 Validation of ARP Packets on a DAI-Enabled VLAN



Use the trust state configuration carefully. Configuring interfaces as untrusted when they should be trusted can result in a loss of connectivity. If we assume that both S1 and S2 (in Figure 58-2) run DAI on the VLAN ports that contains H1 and H2, and if H1 and H2 were to acquire their IP addresses from the DHCP server connected to S1, then only S1 binds the IP to MAC address of H1. If the interface between S1 and S2 is untrusted, the ARP packets from H1 get dropped on S2. This condition would result in a loss of connectivity between H1 and H2.

Configuring interfaces to be trusted when they are actually untrusted leaves a security hole in the network. If S1 were not running DAI, then H1 can easily poison the ARP of S2 (and H2, if the inter-switch link is configured as trusted). This condition can occur even though S2 is running DAI.

DAI ensures that hosts (on untrusted interfaces) connected to a switch running DAI do not poison the ARP caches of other hosts in the network. It does not, however, ensure that hosts from other portions of the network do not poison the caches of the hosts connected to it.

To handle cases in which some switches in a VLAN run DAI and other switches do not, the interfaces connecting such switches should be configured as untrusted. To validate the bindings of packets from non-DAI switches, however, the switch running DAI should be configured with ARP ACLs. When it is not feasible to determine such bindings, switches running DAI should be isolated from non-DAI switches at Layer 3.



Note

Depending on the set up of the DHCP server and the network, it may not be possible to perform validation of a given ARP packet on all switches in the VLAN.

Relative Priority of Static Bindings and DHCP Snooping Entries

As mentioned previously, DAI populates its database of valid MAC address to IP address bindings through DHCP snooping. It also validates ARP packets against statically configured ARP ACLs. It is important to note that ARP ACLs have precedence over entries in the DHCP snooping database. ARP packets are first compared to user-configured ARP ACLs. If the ARP ACL denies the ARP packet, then the packet is denied even if a valid binding exists in the database populated by DHCP snooping.

Logging of Dropped Packets

When the switch drops a packet, it places an entry in the log buffer and then generates system messages on a rate-controlled basis. After the message is generated, the switch clears the entry from the log buffer. Each log entry contains flow information, such as the receiving VLAN, the port number, the source and destination IP addresses, and the source and destination MAC addresses.

You use the **ip arp inspection log-buffer** global configuration command to configure the number of entries in the buffer and the number of entries needed in the specified interval to generate system messages. You specify the type of packets that are logged by using the **ip arp inspection vlan logging** global configuration command. For configuration information, see the [“Configuring the Log Buffer” section on page 58-14](#).

Rate Limiting of ARP Packets

DAI performs validation checks in the CPU, so the number of incoming ARP packets is rate-limited to prevent a denial of service attack. By default, the rate for untrusted interfaces is set to 15 pps second but trusted interfaces have no rate limit. When the rate of incoming ARP packets exceeds the configured limit, the port is placed in the error-disable state. The port remains in that state until an administrator intervenes. With the **errdisable recovery** global configuration command, you can enable error-disable recovery so that ports emerge from this state automatically after a specified timeout period.

You use the **ip arp inspection limit** global configuration command to limit the rate of incoming ARP requests and responses on the interface. Unless a rate limit is explicitly configured on an interface, changing the trust state of the interface also changes its rate limit to the default value for that trust state; that is, 15 packets per second for untrusted interfaces and unlimited for trusted interfaces. Once a rate limit is configured explicitly, the interface retains the rate limit even when its trust state is changed. At any time, the interface reverts to its default rate limit if the **no** form of the **rate limit** command is applied. For configuration information, see the [“Limiting the Rate of Incoming ARP Packets” section on page 58-16](#).



Note

When you enable DAI, all ARP packets are forwarded by CPU (software forwarding, the slow path). With this mechanism, whenever a packet exits through multiple ports, the CPU must create as many copies of the packet as there are egress ports. The number of egress ports is a multiplying factor for the CPU. When QoS policing is applied on egress packets that were forwarded by CPU, QoS must be applied in the CPU as well. (You cannot apply QoS in hardware on CPU generated packets because the hardware forwarding path is turned off for CPU generated packets.) Both factors can drive the CPU to a very high utilization level.

Port Channels Function

A given physical port can join a channel only when the trust state of the physical port and of the channel match. Otherwise, the physical port remains suspended in the channel. A channel inherits its trust state from the first physical port that joined the channel. Consequently, the trust state of the first physical port need not match the trust state of the channel.

Conversely, when the trust state is changed on the channel, the new trust state is configured on all the physical ports that comprise the channel.

The rate limit check on port channels is unique. The rate of incoming packets on a physical port is checked against the port channel configuration rather than the physical ports' configuration.

The rate limit configuration on a port channel is independent of the configuration on its physical ports.

The rate limit is cumulative across all physical ports; that is, the rate of incoming packets on a port channel equals the sum of rates across all physical ports.

When you configure rate limits for ARP packets on trunks, you must account for VLAN aggregation because a high rate limit on one VLAN can cause a denial of service attack to other VLANs when the port is error-disabled by software. Similarly, when a port channel is error-disabled, a high rate limit on one physical port can cause other ports in the channel to go down.

Configuring Dynamic ARP Inspection

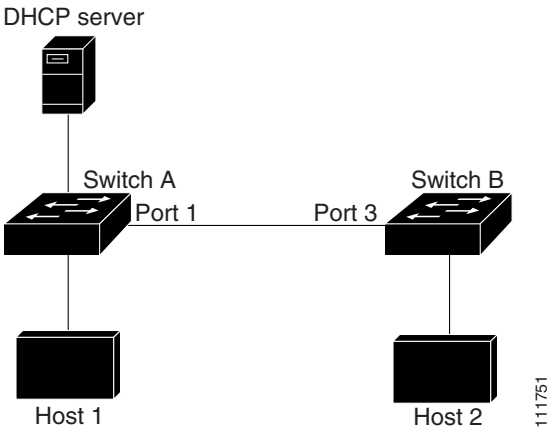
These sections describe how to configure DAI on your switch:

- [Configuring Dynamic ARP Inspection in DHCP Environments, page 58-5](#) (required)
- [DAI Configuration Example, page 58-7](#)
- [Configuring ARP ACLs for Non-DHCP Environments, page 58-11](#) (optional)
- [Configuring the Log Buffer, page 58-14](#) (optional)
- [Limiting the Rate of Incoming ARP Packets, page 58-16](#) (optional)
- [Performing Validation Checks, page 58-19](#) (optional)

Configuring Dynamic ARP Inspection in DHCP Environments

This procedure shows how to configure dynamic ARP inspection when two switches support this feature. Host 1 is connected to Switch A, and Host 2 is connected to Switch B as shown in Figure 58-3. Both switches are running DAI on VLAN 100 where the hosts are located. A DHCP server is connected to Switch A. Both hosts acquire their IP addresses from the same DHCP server. Switch A has the bindings for Host 1, and Switch B has the bindings for Host 2.

Figure 58-3 ARP Packet Validation on a VLAN Enabled for DAI



Note

DAI depends on the entries in the DHCP snooping binding database to verify IP-to-MAC address bindings in incoming ARP requests and ARP responses. Make sure to enable DHCP snooping to permit ARP packets that have dynamically assigned IP addresses. For configuration information, see [Chapter 60, “Configuring DHCP Snooping, IP Source Guard, and IPSG for Static Hosts.”](#)

For information on how to configure DAI when only one switch supports the feature, see the [“Configuring ARP ACLs for Non-DHCP Environments”](#) section on page 58-11.

To configure DAI, perform this task on both switches:

	Command	Purpose
Step 1	Switch# show cdp neighbors	Verifies the connection between the switches.
Step 2	Switch# configure terminal	Enters global configuration mode.
Step 3	Switch(config)# [no] ip arp inspection vlan <i>vlan-range</i>	<p>Enables DAI on a per-VLAN basis. By default, DAI is disabled on all VLANs.</p> <p>To disable DAI, use the no ip arp inspection vlan <i>vlan-range</i> global configuration command.</p> <p>For <i>vlan-range</i>, specify a single VLAN identified by VLAN ID number, a range of VLANs separated by a hyphen, or a series of VLANs separated by a comma. The range is 1 to 4094.</p> <p>Specify the same VLAN ID for both switches.</p>
Step 4	Switch(config)# interface <i>interface-id</i>	Specifies the interface connected to the other switch, and enter interface configuration mode.

	Command	Purpose
Step 5	Switch(config-if)# ip arp inspection trust	Configures the connection between the switches as trusted. To return the interfaces to an untrusted state, use the no ip arp inspection trust interface configuration command. By default, all interfaces are untrusted. The switch does not check ARP packets that it receives from the other switch on the trusted interface. It forwards the packets. For untrusted interfaces, the switch intercepts all ARP requests and responses. It verifies that the intercepted packets have valid IP-to-MAC address bindings before updating the local cache and before forwarding the packet to the appropriate destination. The switch drops invalid packets and logs them in the log buffer according to the logging configuration specified with the ip arp inspection vlan logging global configuration command. For more information, see the “Configuring the Log Buffer” section on page 58-14 .
Step 6	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 7	Switch# show ip arp inspection interfaces Switch# show ip arp inspection vlan <i>vlan-range</i>	Verifies the DAI configuration.
Step 8	Switch# show ip dhcp snooping binding	Verifies the DHCP bindings.
Step 9	Switch# show ip arp inspection statistics vlan <i>vlan-range</i>	Checks the DAI statistics.
Step 10	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

DAI Configuration Example

This example shows how to configure DAI on Switch A in VLAN 100. You would perform a similar procedure on Switch B.

Switch A

```
SwitchA# show cdp neighbors
```

```
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge  
S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone
```

```
Device ID           Local Intrfce    Holdtme    Capability    Platform    Port ID  
SwitchB             Gig 3/48        179        R S I        WS-C4506    Gig 3/46
```

```
SwitchA# configure terminal
```

```
SwitchA(config)# ip arp inspection vlan 100
```

```
SwitchA(config)# interface g3/48
```

```
SwitchA(config-if)# ip arp inspection trust
```

```
SwitchA(config-if)# end
```

```
SwitchA# show ip arp inspection interfaces
```

```
Interface           Trust State      Rate (pps)      Burst Interval  
-----  
Gil/1               Untrusted       15              1  
Gil/2               Untrusted       15              1
```

Gi3/1	Untrusted	15	1
Gi3/2	Untrusted	15	1
Gi3/3	Untrusted	15	1
Gi3/4	Untrusted	15	1
Gi3/5	Untrusted	15	1
Gi3/6	Untrusted	15	1
Gi3/7	Untrusted	15	1
Gi3/8	Untrusted	15	1
Gi3/9	Untrusted	15	1
Gi3/10	Untrusted	15	1
Gi3/11	Untrusted	15	1
Gi3/12	Untrusted	15	1
Gi3/13	Untrusted	15	1
Gi3/14	Untrusted	15	1
Gi3/15	Untrusted	15	1
Gi3/16	Untrusted	15	1
Gi3/17	Untrusted	15	1
Gi3/18	Untrusted	15	1
Gi3/19	Untrusted	15	1
Gi3/20	Untrusted	15	1
Gi3/21	Untrusted	15	1
Gi3/22	Untrusted	15	1
Gi3/23	Untrusted	15	1
Gi3/24	Untrusted	15	1
Gi3/25	Untrusted	15	1
Gi3/26	Untrusted	15	1
Gi3/27	Untrusted	15	1
Gi3/28	Untrusted	15	1
Gi3/29	Untrusted	15	1
Gi3/30	Untrusted	15	1
Gi3/31	Untrusted	15	1
Gi3/32	Untrusted	15	1
Gi3/33	Untrusted	15	1
Gi3/34	Untrusted	15	1
Gi3/35	Untrusted	15	1
Gi3/36	Untrusted	15	1
Gi3/37	Untrusted	15	1
Gi3/38	Untrusted	15	1
Gi3/39	Untrusted	15	1
Gi3/40	Untrusted	15	1
Gi3/41	Untrusted	15	1
Gi3/42	Untrusted	15	1
Gi3/43	Untrusted	15	1
Gi3/44	Untrusted	15	1
Gi3/45	Untrusted	15	1
Gi3/46	Untrusted	15	1
Gi3/47	Untrusted	15	1
Gi3/48	Trusted	None	N/A

SwitchA# **show ip arp inspection vlan 100**

Source Mac Validation : Disabled

Destination Mac Validation : Disabled

IP Address Validation : Disabled

Vlan	Configuration	Operation	ACL Match	Static ACL
----	-----	-----	-----	-----
100	Enabled	Active		
Vlan	ACL Logging	DHCP Logging		
----	-----	-----		
100	Deny	Deny		

SwitchA# **show ip dhcp snooping binding**

MacAddress	IpAddress	Lease(sec)	Type	VLAN	Interface
00:01:00:01:00:01	170.1.1.1	3597	dhcp-snooping	100	GigabitEthernet3/27

Total number of bindings: 1

SwitchA# **show ip arp inspection statistics vlan 100**

Vlan	Forwarded	Dropped	DHCP Drops	ACL Drops
100	15	0	0	0

Vlan	DHCP Permits	ACL Permits	Source MAC Failures
100	0	0	0

Vlan	Dest MAC Failures	IP Validation Failures	Invalid Protocol Data
100	0	0	0

SwitchA#

Switch B

SwitchB# **show cdp neighbors**

Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone

Device ID	Local Intrfce	Holdtme	Capability	Platform	Port ID
SwitchA	Gig 3/46	163	R S I	WS-C4507R	Gig 3/48

SwitchB#

SwitchB# **configure terminal**

Enter configuration commands, one per line. End with CNTL/Z.

SwitchB(config)# **ip arp inspection vlan 100**

SwitchB(config)# **interface g3/46**

SwitchB(config-if)# **ip arp inspection trust**

SwitchB(config-if)# **end**

SwitchB#

SwitchB# **show ip arp inspection interfaces**

Interface	Trust State	Rate (pps)	Burst Interval
Gi1/1	Untrusted	15	1
Gi1/2	Untrusted	15	1
Gi3/1	Untrusted	15	1
Gi3/2	Untrusted	15	1
Gi3/3	Untrusted	15	1
Gi3/4	Untrusted	15	1
Gi3/5	Untrusted	15	1
Gi3/6	Untrusted	15	1
Gi3/7	Untrusted	15	1
Gi3/8	Untrusted	15	1
Gi3/9	Untrusted	15	1
Gi3/10	Untrusted	15	1
Gi3/11	Untrusted	15	1
Gi3/12	Untrusted	15	1
Gi3/13	Untrusted	15	1
Gi3/14	Untrusted	15	1
Gi3/15	Untrusted	15	1
Gi3/16	Untrusted	15	1
Gi3/17	Untrusted	15	1
Gi3/18	Untrusted	15	1
Gi3/19	Untrusted	15	1

Gi3/20	Untrusted	15	1
Gi3/21	Untrusted	15	1
Gi3/22	Untrusted	15	1
Gi3/23	Untrusted	15	1
Gi3/24	Untrusted	15	1
Gi3/25	Untrusted	15	1
Gi3/26	Untrusted	15	1
Gi3/27	Untrusted	15	1
Gi3/28	Untrusted	15	1
Gi3/29	Untrusted	15	1
Gi3/30	Untrusted	15	1
Gi3/31	Untrusted	15	1
Gi3/32	Untrusted	15	1
Gi3/33	Untrusted	15	1
Gi3/34	Untrusted	15	1
Gi3/35	Untrusted	15	1
Gi3/36	Untrusted	15	1
Gi3/37	Untrusted	15	1
Gi3/38	Untrusted	15	1
Gi3/39	Untrusted	15	1
Gi3/40	Untrusted	15	1
Gi3/41	Untrusted	15	1
Gi3/42	Untrusted	15	1
Gi3/43	Untrusted	15	1
Gi3/44	Untrusted	15	1
Gi3/45	Untrusted	15	1
Gi3/46	Trusted	None	N/A
Gi3/47	Untrusted	15	1
Gi3/48	Untrusted	15	1

SwitchB# **show ip arp inspection vlan 100**

Source Mac Validation : Disabled

Destination Mac Validation : Disabled

IP Address Validation : Disabled

Vlan	Configuration	Operation	ACL Match	Static ACL
----	-----	-----	-----	-----
100	Enabled	Active		
Vlan	ACL Logging	DHCP Logging		
----	-----	-----		
100	Deny	Deny#		

SwitchB# **show ip dhcp snooping binding**

MacAddress	IpAddress	Lease(sec)	Type	VLAN	Interface
-----	-----	-----	-----	----	-----
00:02:00:02:00:02	170.1.1.2	3492	dhcp-snooping	100	GigabitEthernet3/31
Total number of bindings: 1					

SwitchB# **show ip arp insp statistics vlan 100**

Vlan	Forwarded	Dropped	DHCP Drops	ACL Drops
----	-----	-----	-----	-----
100	2398	0	0	0
Vlan	DHCP Permits	ACL Permits	Source MAC Failures	
----	-----	-----	-----	
100	2398	0	0	
Vlan	Dest MAC Failures	IP Validation Failures	Invalid Protocol Data	
----	-----	-----	-----	
100	0	0	0	

SwitchB#

Configuring ARP ACLs for Non-DHCP Environments

This procedure shows how to configure DAI when Switch B shown in Figure 58-3 does not support DAI or DHCP snooping.

If you configure port 1 on Switch A as trusted, a security hole is created because both Switch A and Host 1 could be attacked by either Switch B or Host 2. To prevent this possibility, you must configure port 1 on Switch A as untrusted. To permit ARP packets from Host 2, you must set up an ARP ACL and apply it to VLAN 100. If the IP address of Host 2 is not static, such that it is impossible to apply the ACL configuration on Switch A, you must separate Switch A from Switch B at Layer 3 and use a router to route packets between them.

To configure an ARP ACL (on switch A in a non-DHCP environment), perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# arp access-list <i>acl-name</i>	Defines an ARP ACL, and enter ARP access-list configuration mode. By default, no ARP access lists are defined. Note At the end of the ARP access list, there is an implicit deny ip any mac any command.
Step 3	Switch(config-arp-nac)# permit ip host <i>sender-ip</i> mac <i>host sender-mac</i> [log]	Permits ARP packets from the specified host (Host 2). <ul style="list-style-type: none"> For <i>sender-ip</i>, enter the IP address of Host 2. For <i>sender-mac</i>, enter the MAC address of Host 2. (Optional) Specify log to log a packet in the log buffer when it matches the access control entry (ACE). Matches are logged if you also configure the matchlog keyword in the ip arp inspection vlan logging global configuration command. For more information, see the “Configuring the Log Buffer” section on page 58-14.
Step 4	Switch(config-arp-nac)# exit	Returns to global configuration mode.

	Command	Purpose
Step 5	Switch(config)# ip arp inspection filter <i>arp-acl-name</i> vlan <i>vlan-range</i> [static]	<p>Applies the ARP ACL to the VLAN. By default, no defined ARP ACLs are applied to any VLAN.</p> <ul style="list-style-type: none"> For <i>arp-acl-name</i>, specify the name of the ACL created in Step 2. For <i>vlan-range</i>, specify the VLAN that the switches and hosts are in. You can specify a single VLAN identified by VLAN ID number, a range of VLANs separated by a hyphen, or a series of VLANs separated by a comma. The range is 1 to 4094. (Optional) Specify static to treat implicit denies in the ARP ACL as explicit denies and to drop packets that do not match any previous clauses in the ACL. DHCP bindings are not used. <p>If you do not specify this keyword, it means that there is no explicit deny in the ACL that denies the packet, and DHCP bindings determine whether a packet is permitted or denied if the packet does not match any clauses in the ACL.</p> <p>ARP packets containing only IP-to-MAC address bindings are compared against the ACL. Packets are permitted only if the access list permits them.</p>
Step 6	Switch(config)# interface <i>interface-id</i>	Specifies the Switch A interface that is connected to Switch B, and enter interface configuration mode.
Step 7	Switch(config-if)# no ip arp inspection trust	<p>Configures the Switch A interface that is connected to Switch B as untrusted.</p> <p>By default, all interfaces are untrusted.</p> <p>For untrusted interfaces, the switch intercepts all ARP requests and responses. It verifies that the intercepted packets have valid IP-to-MAC address bindings before updating the local cache and before forwarding the packet to the appropriate destination. The switch drops invalid packets and logs them in the log buffer according to the logging configuration specified with the ip arp inspection vlan logging global configuration command. For more information, see the “Configuring the Log Buffer” section on page 58-14.</p>
Step 8	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 9	Switch# show arp access-list [<i>acl-name</i>] Switch# show ip arp inspection vlan <i>vlan-range</i> Switch# show ip arp inspection interfaces	Verifies the DAI configuration.
Step 10	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To remove the ARP ACL, use the **no arp access-list** global configuration command. To remove the ARP ACL attached to a VLAN, use the **no ip arp inspection filter arp-acl-name vlan vlan-range** global configuration command.

This example shows how to configure an ARP ACL called host2 on Switch A, to permit ARP packets from HostB (IP address 170.1.1.2 and MAC address 2.2.2), to apply the ACL to VLAN 100, and to configure port 1 on Switch A as untrusted:

```
SwitchA# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SwitchA(config)# arp access-list hostB
SwitchA(config-arp-nacl)# permit ip host 170.1.1.2 mac host 2.2.2 log
SwitchA(config-arp-nacl)# exit
SwitchA(config)# ip arp inspection filter hostB vlan 100 static
SwitchA(config)# interface g3/48
SwitchA(config-if)# no ip arp inspection trust
SwitchA(config-if)# end
SwitchA# show arp access-list hostB
ARP access list hostB
    permit ip host 170.1.1.2 mac host 0002.0002.0002 log
```

```
SwitchA# show ip arp inspection interfaces
```

Interface	Trust State	Rate (pps)	Burst Interval
-----	-----	-----	-----
Gi1/1	Untrusted	15	1
Gi1/2	Untrusted	15	1
Gi3/1	Untrusted	15	1
Gi3/2	Untrusted	15	1
Gi3/3	Untrusted	15	1
Gi3/4	Untrusted	15	1
Gi3/5	Untrusted	15	1
Gi3/6	Untrusted	15	1
Gi3/7	Untrusted	15	1
Gi3/8	Untrusted	15	1
Gi3/9	Untrusted	15	1
Gi3/10	Untrusted	15	1
Gi3/11	Untrusted	15	1
Gi3/12	Untrusted	15	1
Gi3/13	Untrusted	15	1
Gi3/14	Untrusted	15	1
Gi3/15	Untrusted	15	1
Gi3/16	Untrusted	15	1
Gi3/17	Untrusted	15	1
Gi3/18	Untrusted	15	1
Gi3/19	Untrusted	15	1
Gi3/20	Untrusted	15	1
Gi3/21	Untrusted	15	1
Gi3/22	Untrusted	15	1
Gi3/23	Untrusted	15	1
Gi3/24	Untrusted	15	1
Gi3/25	Untrusted	15	1
Gi3/26	Untrusted	15	1
Gi3/27	Untrusted	15	1
Gi3/28	Untrusted	15	1
Gi3/29	Untrusted	15	1
Gi3/30	Untrusted	15	1
Gi3/31	Untrusted	15	1
Gi3/32	Untrusted	15	1
Gi3/33	Untrusted	15	1
Gi3/34	Untrusted	15	1

Gi3/35	Untrusted	15	1
Gi3/36	Untrusted	15	1
Gi3/37	Untrusted	15	1
Gi3/38	Untrusted	15	1
Gi3/39	Untrusted	15	1
Gi3/40	Untrusted	15	1
Gi3/41	Untrusted	15	1
Gi3/42	Untrusted	15	1
Gi3/43	Untrusted	15	1
Gi3/44	Untrusted	15	1
Gi3/45	Untrusted	15	1
Gi3/46	Untrusted	15	1
Gi3/47	Untrusted	15	1
Gi3/48	Untrusted	15	1

SwitchA# **show ip arp inspection statistics vlan 100**

Vlan	Forwarded	Dropped	DHCP Drops	ACL Drops
100	15	169	160	9

Vlan	DHCP Permits	ACL Permits	Source MAC Failures
100	0	0	0

Vlan	Dest MAC Failures	IP Validation Failures	Invalid Protocol Data
100	0	0	0

SwitchA#

Configuring the Log Buffer

When the switch drops a packet, it places an entry in the log buffer and then generates system messages on a rate-controlled basis. After the message is generated, the switch clears the entry from the log buffer. Each log entry contains flow information, such as the receiving VLAN, the port number, the source and destination IP addresses, and the source and destination MAC addresses.

A log-buffer entry can represent more than one packet. For example, if an interface receives many packets on the same VLAN with the same ARP parameters, the switch combines the packets as one entry in the log buffer and generates a single system message for the entry.

If the log buffer overflows, it means that a log event does not fit into the log buffer, and the display for the **show ip arp inspection log** privileged EXEC command is affected. No other statistics are provided for the entry.

To configure the log buffer, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# ip arp inspection log-buffer {entries number logs number interval seconds}	<p>Configures the DAI logging buffer.</p> <p>By default, when DAI is enabled, denied or dropped ARP packets are logged. The number of log entries is 32. The number of system messages is limited to 5 per second. The logging-rate interval is 1 second.</p> <p>The keywords have these meanings:</p> <ul style="list-style-type: none"> For entries number, specify the number of entries to be logged in the buffer. The range is 0 to 1024. For logs number interval seconds, specify the number of entries to generate system messages in the specified interval. <p>For logs number, the range is 0 to 1024. A 0 value means that the entry is placed in the log buffer, but a system message is not generated.</p> <p>For interval seconds, the range is 0 to 86400 seconds (1 day). A 0 value means that a system message is immediately generated (and the log buffer is always empty).</p> <p>An interval setting of 0 overrides a log setting of 0.</p> <p>The logs and interval settings interact. If the logs number X is greater than interval seconds Y, X divided by Y (X/Y) system messages are sent every second. Otherwise, one system message is sent every Y divided by X (Y/X) seconds.</p>
Step 3	Switch(config)# [no] ip arp inspection vlan vlan-range logging {acl-match {matchlog none} dhcp-bindings {all none permit}}	<p>Controls the type of packets that are logged per-VLAN. By default, all denied or all dropped packets are logged. The term <i>logged</i> means the entry is placed in the log buffer and a system message is generated.</p> <p>The keywords have these meanings:</p> <ul style="list-style-type: none"> For vlan-range, specify a single VLAN identified by VLAN ID number, a range of VLANs separated by a hyphen, or a series of VLANs separated by a comma. The range is 1 to 4094. For acl-match matchlog, log packets based on the ACE logging configuration. If you specify the matchlog keyword in this command and the log keyword in the permit or deny ARP access-list configuration command, ARP packets permitted or denied by ACEs with log keyword are logged. For acl-match none, do not log packets that match ACLs. For dhcp-bindings all, log all packets that match DHCP bindings. For dhcp-bindings none, do not log packets that match DHCP bindings. For dhcp-bindings permit, log DHCP-binding permitted packets.
Step 4	Switch(config)# exit	Returns to privileged EXEC mode.

	Command	Purpose
Step 5	Switch# show ip arp inspection log	Verifies your settings.
Step 6	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To return to the default log buffer settings, use the **no ip arp inspection log-buffer** global configuration command. To return to the default VLAN log settings, use the **no ip arp inspection vlan *vlan-range* logging {acl-match | dhcp-bindings}** global configuration command. To clear the log buffer, use the **clear ip arp inspection log** privileged EXEC command.

This example shows how to configure the number of entries for the log buffer to 1024. It also shows how to configure your Catalyst 4500 series switch so that the logs must be generated from the buffer at the rate of 100 per 10 seconds.

```
SwitchB# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SwitchB(config)# ip arp inspection log-buffer entries 1024
SwitchB(config)# ip arp inspection log-buffer logs 100 interval 10
SwitchB(config)# end
SwitchB# show ip arp inspection log
Total Log Buffer Size : 1024
Syslog rate : 100 entries per 10 seconds.
```

```
Interface   Vlan   Sender MAC      Sender IP      Num Pkts   Reason      Time
-----
Gi3/31     100    0002.0002.0003  170.1.1.2      5    DHCP Deny   02:05:45 UTC
Fri Feb 4 2005
SwitchB#
```

Limiting the Rate of Incoming ARP Packets

The switch CPU performs DAI validation checks; therefore, the number of incoming ARP packets is rate-limited to prevent a denial-of-service attack.



Note

Unless you explicitly configure a rate limit on an interface, changing the trust state of the interface also changes its rate limit to the default value for that trust state. After you configure the rate limit, the interface retains the rate limit even when its trust state is changed. If you enter the **no ip arp-inspection limit** interface configuration command, the interface reverts to its default rate limit.

By default, the switch places the port in the error-disabled state when the rate of incoming ARP packets exceeds the configured limit. To prevent the port from shutting down, you can use the **errdisable detect cause arp-inspection action shutdown vlan** global configuration command to shut down just the offending VLAN on the port where the violation occurred.

When a port is in the error-disabled state, you can bring it out of this state automatically by configuring the **errdisable recovery cause arp-inspection** global configuration command or you can manually reenale it by entering the **shutdown** and **no shut down** interface configuration commands. If a port is in per-VLAN error-disable mode, you can also use **clear errdisable interface *name* *vlan range*** command to reenale the VLAN on the port.

To limit the rate of incoming ARP packets, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# errdisable detect cause arp-inspection [action shutdown vlan]	Enables per-VLAN error-disable detection. Note By default this command is enabled, and when a violation occurs the interface is shutdown.
Step 3	Switch(config)# interface interface-id	Specifies the interface to be rate-limited, and enters interface configuration mode.
Step 4	Switch(config-if)# [no] ip arp inspection limit {rate pps [burst interval second] none}	Limits the rate of incoming ARP requests and responses on the interface. The default rate is 15 pps on untrusted interfaces and unlimited on trusted interfaces. The burst interval is 1 second. The keywords have these meanings: <ul style="list-style-type: none"> For rate pps, specify an upper limit for the number of incoming packets processed per second. The range is 0 to 2048 pps. (Optional) For burst interval seconds, specify the consecutive interval in seconds, over which the interface is monitored for a high rate of ARP packets. The range is 1 to 15. For rate none, specify no upper limit for the rate of incoming ARP packets that can be processed.
Step 5	Switch(config-if)# exit	Returns to global configuration mode.
Step 6	Switch(config)# errdisable recovery {cause arp-inspection interval interval}	(Optional) Enables error recovery from the DAI error-disable state. By default, recovery is disabled, and the recovery interval is 300 seconds. For interval interval , specify the time in seconds to recover from the error-disable state. The range is 30 to 86400.
Step 7	Switch(config)# exit	Returns to privileged EXEC mode.
Step 8	Switch# show ip arp inspection interfaces	Verifies your settings.
Step 9	Switch# show errdisable recovery	Verifies your settings.
Step 10	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To return to the default rate-limit configuration, use the **no ip arp inspection limit** interface configuration command. To disable error recovery for DAI, use the **no errdisable recovery cause arp-inspection** global configuration command.

This example shows how to set an upper limit for the number of incoming packets (100 pps) and to specify a burst interval (1 second):

```
SwitchB# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SwitchB(config)# interface g3/31
SwitchB(config-if)# ip arp inspection limit rate 100 burst interval 1
SwitchB(config-if)# exit
SwitchB(config)# errdisable recovery cause arp-inspection
SwitchB(config)# exit
```

SwitchB# **show ip arp inspection interfaces**

Interface	Trust State	Rate (pps)	Burst Interval
-----	-----	-----	-----
Gi1/1	Untrusted	15	1
Gi1/2	Untrusted	15	1
Gi3/1	Untrusted	15	1
Gi3/2	Untrusted	15	1
Gi3/3	Untrusted	15	1
Gi3/4	Untrusted	15	1
Gi3/5	Untrusted	15	1
Gi3/6	Untrusted	15	1
Gi3/7	Untrusted	15	1
Gi3/8	Untrusted	15	1
Gi3/9	Untrusted	15	1
Gi3/10	Untrusted	15	1
Gi3/11	Untrusted	15	1
Gi3/12	Untrusted	15	1
Gi3/13	Untrusted	15	1
Gi3/14	Untrusted	15	1
Gi3/15	Untrusted	15	1
Gi3/16	Untrusted	15	1
Gi3/17	Untrusted	15	1
Gi3/18	Untrusted	15	1
Gi3/19	Untrusted	15	1
Gi3/20	Untrusted	15	1
Gi3/21	Untrusted	15	1
Gi3/22	Untrusted	15	1
Gi3/23	Untrusted	15	1
Gi3/24	Untrusted	15	1
Gi3/25	Untrusted	15	1
Gi3/26	Untrusted	15	1
Gi3/27	Untrusted	15	1
Gi3/28	Untrusted	15	1
Gi3/29	Untrusted	15	1
Gi3/30	Untrusted	15	1
Gi3/31	Untrusted	100	1
Gi3/32	Untrusted	15	1
Gi3/33	Untrusted	15	1
Gi3/34	Untrusted	15	1
Gi3/35	Untrusted	15	1
Gi3/36	Untrusted	15	1
Gi3/37	Untrusted	15	1
Gi3/38	Untrusted	15	1
Gi3/39	Untrusted	15	1
Gi3/40	Untrusted	15	1
Gi3/41	Untrusted	15	1
Gi3/42	Untrusted	15	1
Gi3/43	Untrusted	15	1
Gi3/44	Untrusted	15	1
Gi3/45	Untrusted	15	1
Gi3/46	Trusted	None	N/A
Gi3/47	Untrusted	15	1
Gi3/48	Untrusted	15	1

SwitchB# **show errdisable recovery**

ErrDisable Reason	Timer Status
-----	-----
udld	Disabled
bpduguard	Disabled
security-violatio	Disabled
channel-misconfig	Disabled

```

vmps                Disabled
pagp-flap           Disabled
dtp-flap            Disabled
link-flap           Disabled
l2ptguard           Disabled
psecure-violation   Disabled
gbic-invalid        Disabled
dhcp-rate-limit     Disabled
unicast-flood       Disabled
storm-control       Disabled
arp-inspection      Enabled

```

Timer interval: 300 seconds

Interfaces that will be enabled at the next timeout:

```

SwitchB#
1w2d: %SW_DAI-4-PACKET_RATE_EXCEEDED: 101 packets received in 739 milliseconds on Gi3/31.
1w2d: %PM-4-ERR_DISABLE: arp-inspection error detected on Gi3/31, putting Gi3/31 in
err-disable state
SwitchB# show clock
*02:21:43.556 UTC Fri Feb 4 2005
SwitchB#
SwitchB# show interface g3/31 status

Port      Name      Status      Vlan      Duplex  Speed  Type
Gi3/31                               err-disabled 100          auto   auto  10/100/1000-TX
SwitchB#
SwitchB#
1w2d: %PM-4-ERR_RECOVER: Attempting to recover from arp-inspection err-disable state on
Gi3/31
SwitchB# show interface g3/31 status

Port      Name      Status      Vlan      Duplex  Speed  Type
Gi3/31                               connected   100      a-full  a-100  10/100/1000-TX
SwitchB# show clock
*02:27:40.336 UTC Fri Feb 4 2005
SwitchB#

```

Performing Validation Checks

DAI intercepts, logs, and discards ARP packets with invalid IP-to-MAC address bindings. You can configure the switch to perform additional checks on the destination MAC address, the sender and target IP addresses, and the source MAC address.

To perform specific checks on incoming ARP packets, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# ip arp inspection validate {[src-mac] [dst-mac] [ip]}	<p>Performs a specific check on incoming ARP packets. By default, no additional checks are performed.</p> <p>The keywords have these meanings:</p> <ul style="list-style-type: none"> For src-mac, check the source MAC address in the Ethernet header against the sender MAC address in the ARP body. This check is performed on both ARP requests and responses. When enabled, packets with different MAC addresses are classified as invalid and are dropped. For dst-mac, check the destination MAC address in the Ethernet header against the target MAC address in ARP body. This check is performed for ARP responses. When enabled, packets with different MAC addresses are classified as invalid and are dropped. For ip, check the ARP body for invalid and unexpected IP addresses. Addresses include 0.0.0.0, 255.255.255.255, and all IP multicast addresses. Sender IP addresses are checked in all ARP requests and responses, and target IP addresses are checked only in ARP responses. <p>You must specify at least one of the keywords. Each command overrides the configuration of the previous command; that is, if a command enables src and dst mac validations, and a second command enables IP validation only, the src and dst mac validations are disabled as a result of the second command.</p>
Step 3	Switch(config)# exit	Returns to privileged EXEC mode.
Step 4	Switch# show ip arp inspection vlan vlan-range	Verifies your settings.
Step 5	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To disable checking, use the **no ip arp inspection validate [src-mac] [dst-mac] [ip]** global configuration command. To display statistics for forwarded, dropped, MAC validation failure, and IP validation failure packets, use the **show ip arp inspection statistics** privileged EXEC command.

This example shows how to configure source mac validation. Packets are dropped and an error message may be generated when the source address in the Ethernet header does not match the sender hardware address in the ARP body.

```
SwitchB# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SwitchB(config)# ip arp inspection validate src-mac
SwitchB(config)# exit
SwitchB# show ip arp inspection vlan 100

Source Mac Validation      : Enabled
Destination Mac Validation : Disabled
IP Address Validation      : Disabled
```

Vlan	Configuration	Operation	ACL Match	Static ACL
----	-----	-----	-----	-----
100	Enabled	Active		

Vlan	ACL Logging	DHCP Logging
----	-----	-----
100	Deny	Deny

SwitchB#

1w2d: %SW_DAI-4-INVALID_ARP: 9 Invalid ARPs (Req) on Gi3/31, vlan

100. ([0002.0002.0002/170.1.1.2/0001.0001.0001/170.1.1.1/02:30:24 UTC Fri Feb 4 2005])



Configuring the Cisco IOS DHCP Server

Cisco devices running Cisco software include Dynamic Host Configuration Protocol (DHCP) server and the relay agent software. The Cisco IOS DHCP server is a full DHCP server implementation that assigns and manages IP addresses from specified address pools within the device to DHCP clients. The DHCP server can be configured to assign additional parameters such as the IP address of the Domain Name System (DNS) server and the default device.

This module describes the concepts and the tasks needed to configure the Cisco IOS DHCP server.

- [Prerequisites for Configuring the DHCP Server, page 59-1](#)
- [Information About Cisco IOS DHCP Server, page 59-2](#)
- [How to Configure the Cisco IOS DHCP Server, page 59-9](#)
- [Configuration Examples for the Cisco IOS DHCP Server, page 59-24](#)

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the [“Feature Information for the IOS DHCP Server”](#) section on page 59-34.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Prerequisites for Configuring the DHCP Server

- Before you configure a Cisco Dynamic Host Control Protocol (DHCP) server, you must understand the concepts documented in the “Overview of the DHCP Server” section.
- The Cisco DHCP server and the relay agent services are enabled by default. Use the **no service dhcp** command to disable the Cisco DHCP server and the relay agent and the **service dhcp** command to reenble the functionality.

- Port 67 (the DHCP server port) is closed in the Cisco DHCP/BOOTP default configuration. There are two logical parts to the **service dhcp** command: service-enabled and service running. The DHCP service is enabled by default, but port 67 does not open until the DHCP service is running. If the DHCP service is running, the **show ip sockets details** or the **show sockets detail** command displays port 67 as open.
- The Cisco DHCP relay agent is enabled on an interface only when you configure the **ip helper-address** command. This command enables a DHCP broadcast to be forwarded to the configured DHCP server.

Information About Cisco IOS DHCP Server

- [Overview of the DHCP Server, page 59-2](#)
- [DHCP Attribute Inheritance, page 59-2](#)
- [DHCP Server Address Allocation Using Option 82, page 59-3](#)
- [Disabling Conflict Logging, page 59-4](#)
- [DHCP Address Pools, page 59-5](#)
- [Manual Bindings, page 59-6](#)
- [DHCP Static Mapping, page 59-7](#)
- [DHCP Server Operation, page 59-8](#)
- [Static Route with the Next-Hop Dynamically Obtained Through DHCP, page 59-9](#)

Overview of the DHCP Server

The Cisco DHCP server accepts address assignment requests and renewals from the client and assigns the addresses from predefined groups of addresses within DHCP address pools. These address pools can also be configured to supply additional information to the requesting client such as the IP address of the Domain Name System (DNS) server, the default device, and other configuration parameters. The Cisco DHCP server can accept broadcasts from locally attached LAN segments or from DHCP requests that have been forwarded by other DHCP relay agents within the network.

DHCP Attribute Inheritance

The DHCP server database is organized as a tree. The root of the tree is the address pool for natural networks, branches are subnetwork address pools, and leaves are manual bindings to clients. Subnetworks inherit network parameters and clients inherit subnetwork parameters. Therefore, common parameters (for example, the domain name) should be configured at the highest (network or subnetwork) level of the tree.

Inherited parameters can be overridden. For example, if a parameter is defined in both the natural network and a subnetwork, the definition of the subnetwork is used.

Address leases are not inherited. If a lease is not specified for an IP address, by default, the DHCP server assigns a one-day lease for the address.

DHCP Server Address Allocation Using Option 82

The Cisco IOS DHCP server can allocate dynamic IP addresses based on the relay information option (option 82) sent by the relay agent.

DHCP provides a framework for passing configuration information to hosts on a TCP/IP network. Configuration parameters and other control information are carried in tagged data items that are stored in the options field of the DHCP message. The data items are also called options. Option 82 is organized as a single DHCP option that contains information known by the relay agent.

Automatic DHCP address allocation is based on an IP address. This IP address can either be the gateway address (giaddr field of the DHCP packet) or the IP address of an incoming interface. In some networks, it is necessary to use additional information to further determine which IP addresses to allocate. By using option 82, the Cisco IOS DHCP relay agent can include additional information about itself when forwarding client-originated DHCP packets to a DHCP server. The Cisco IOS DHCP server can also use option 82 to provide additional information to properly allocate IP addresses to DHCP clients. The information sent via option 82 is used to identify the port where the DHCP request arrives. Automatic DHCP address allocation does not parse out the individual suboptions contained in option 82. Rather, the address allocation is done by matching a configured pattern byte by byte.

This feature introduces a new DHCP class capability, which is a method to group DHCP clients based on some shared characteristics other than the subnet in which the clients reside.

For example, DHCP clients are connected to two ports of a single switch. Each port can be configured to be a part of two VLANs: VLAN1 and VLAN2. DHCP clients belong to either VLAN1 or VLAN2 and the switch can differentiate the VLAN that a particular DHCP Discover message belongs to (possibly through Layer 2 encapsulation). Each VLAN has its own subnet and all DHCP messages from the same VLAN (same switch) have the giaddr field set to the same value indicating the subnet of the VLAN.

Problems can occur while allocating IP addresses to DHCP clients that are connected to different ports of the same VLAN. These IP addresses must be part of the same subnet but the range of IP addresses must be different. In the preceding example, when a DHCP client that is connected to a port of VLAN1 must be allocated an IP address from a range of IP addresses within the VLAN's subnet, whereas a DHCP client connecting to port 2 of VLAN1 must be allocated an IP address from another range of IP addresses. The two range of IP addresses are part of the same subnet (and have the same subnet mask). Generally, during DHCP address allocation, the DHCP server refers only to the giaddr field and is unable to differentiate between the two ranges.

To solve this problem, a relay agent residing at the switch inserts the relay information option (option 82), which carries information specific to the port, and the DHCP server inspects both the giaddr field and the inserted option 82 during the address selection process.

When you enable option 82 on a device, the following sequence of events occurs:

1. The host (DHCP client) generates a DHCP request and broadcasts it on the network.
2. When the device receives the DHCP request, it adds the option 82 information in the packet. The option 82 information contains the device MAC address (the remote ID suboption) and the port identifier, vlan-mod-port, from which the packet is received (the circuit ID suboption).
3. The device adds the IP address of the relay agent to the DHCP packet.
4. The device forwards the DHCP request that includes the option 82 field to the DHCP server.
5. The DHCP server receives the packet. If the server is option 82 capable, it uses the remote ID, the circuit ID, or both to assign IP addresses and implement policies, such as restricting the number of IP addresses that can be assigned to a single remote ID or circuit ID. The DHCP server echoes the option 82 field in the DHCP reply.

6. The DHCP server unicasts the reply to the device if the request is relayed to the server by the device. The device verifies that it originally inserted the option 82 data by inspecting remote ID and possibly circuit ID fields. The device removes the option 82 field and forwards the packet to the interface that connects to the DHCP client that sent the DHCP request.

The Cisco software refers to a pool of IP addresses (giaddr or incoming interface IP address) and matches the request to a class or classes configured in the pool in the order the classes are specified in the DHCP pool configuration.

When a DHCP address pool is configured with one or more DHCP classes, the pool becomes a restricted access pool, which means that no addresses are allocated from the pool unless one or more classes in the pool matches. This design allows DHCP classes to be used either for access control (no default class is configured on the pool) or to provide further address range partitions within the subnet of the pool.

Multiple pools can be configured with the same class, eliminating the need to configure the same pattern in multiple pools.

The following capabilities are supported for DHCP class-based address allocation:

- Specifying the full relay agent information option value as a raw hexadecimal string by using the **relay-information hex** command in new relay agent information configuration mode.
- Support for bit-masking the raw relay information hexadecimal value.
- Support for a wildcard at the end of a hexadecimal string specified by the **relay-information hex** command.

If the relay agent inserts option 82 but does not set the giaddr field in the DHCP packet, the DHCP server interface must be configured as a trusted interface by using the **ip dhcp relay information trusted** command. This configuration prevents the server from dropping the DHCP message.

Disabling Conflict Logging

A DHCP database agent is any host (for example, an FTP, a TFTP, or a remote copy protocol [RCP] server) or storage media on a DHCP server (for example, disk0) that stores the DHCP bindings database. You can configure multiple DHCP database agents, and the interval between database updates and transfers for each agent.

Automatic bindings are IP addresses that are automatically mapped to the MAC addresses of hosts that are found in the DHCP database. Automatic binding information (such as lease expiration date and time, interface index, and VPN routing and forwarding [VRF] name) is stored in a database agent. The bindings are saved as text records for easy maintenance.

An address conflict occurs when two hosts use the same IP address. During address assignment, DHCP checks for conflicts by using ping and gratuitous Address Resolution Protocol (ARP). If a conflict is detected, the address is removed from the pool. The address is not assigned until the administrator resolves the conflict.



Note

We strongly recommend using database agents. However, the Cisco DHCP server can run without database agents. If you choose not to configure a DHCP database agent, disable the recording of DHCP address conflicts on the DHCP server by using the **no ip dhcp conflict logging** command in global configuration mode. If there is a conflict logging but no database agent is configured, bindings during a switchover are lost when a device reboots. Possible false conflicts can occur causing the address to be removed from the address pool.

DHCP Address Pools

You can configure a DHCP address pool with a name that is a string (such as “engineering”) or an integer (such as 0). Configuring a DHCP address pool also puts the device into DHCP pool configuration mode—identified by the `(dhcp-config)#` prompt—from which you can configure pool parameters (for example, the IP subnet number and default device list).

DHCP defines a process by which the DHCP server knows the IP subnet in which the DHCP client resides, and it can assign an IP address from a pool of valid IP addresses in that subnet. The process by which the DHCP server identifies the DHCP address pool to use for a client request is described in the Configuring Manual Bindings section.

The DHCP server identifies and uses DHCP address pools for a client request, in the following manner:

- If the client is not directly connected to the DHCP server (the `giaddr` field of the DHCPDISCOVER broadcast message is nonzero), the server matches the DHCPDISCOVER with the DHCP pool that has the subnet that contains the IP address in the `giaddr` field.
- If the client is directly connected to the DHCP server (the `giaddr` field is zero), the DHCP server matches the DHCPDISCOVER with DHCP pools that contain the subnets configured on the receiving interface. If the interface has secondary IP addresses, subnets associated with the secondary IP addresses are examined for possible allocation only after the subnet associated with the primary IP address (on the interface) is exhausted.

Cisco DHCP server software supports advanced capabilities for IP address allocation. See the Configuring DHCP Address Allocation Using Option 82 section for more information.

DHCP Address Pool with Secondary Subnets

Each subnet is a range of IP addresses that the device uses to allocate an IP address to a DHCP client. The DHCP server multiple subnet functionality enables a Cisco DHCP server address pool to manage additional IP addresses by adding the addresses to a secondary subnet of an existing DHCP address pool (instead of using a separate address pool).

Configuring a secondary DHCP subnetwork places the device in DHCP pool secondary subnet configuration mode—identified by the `(config-dhcp-subnet-secondary)#` prompt—where you can configure a default address list that is specific to the secondary subnet. You can also specify the utilization rate of the secondary subnet, which allows pools of IP addresses to dynamically increase or reduce in size depending on the address utilization level. This setting overrides the global utilization rate.

- If the DHCP server selects an address pool that contains multiple subnets, the DHCP server allocates an IP address from the subnets as follows:
- When the DHCP server receives an address assignment request, it looks for an available IP address in the primary subnet.
- When the primary subnet is exhausted, the DHCP server automatically looks for an available IP address in any of the secondary subnets maintained by the DHCP server (even though the `giaddr` does not necessarily match the secondary subnet). The server inspects the subnets for address availability in the order of subnets that were added to the pool.

**Note**

If the giaddr matches a secondary subnet in the pool, the DHCP server allocates an IP address from that particular secondary subnet (even if IP addresses are available in the primary subnet and irrespective of the order of secondary subnets that were added).

The secondary subnet in the pool is supported only for directly connected clients. To avoid multiple IP address allocation from multiple subnets, you should configure secondary IP address on the interface connected to clients. Note that the secondary subnets should not be used in pools that are used for servicing requests from DHCP relay.

Manual Bindings

An address binding is a mapping between the IP address and MAC address of a client. The IP address of a client can be assigned manually by an administrator or assigned automatically from a pool by a DHCP server.

Manual bindings are IP addresses that are manually mapped to MAC addresses of hosts that are found in the DHCP database. Manual bindings are stored in the NVRAM of the DHCP server. Manual bindings are just special address pools. There is no limit to the number of manual bindings, but you can configure only one manual binding per host pool.

Automatic bindings are IP addresses that have been automatically mapped to MAC addresses of hosts that are found in the DHCP database. Because the bindings are stored in the volatile memory of the DHCP server, binding information is lost in the event of power failures or on device reloads. To prevent the loss of automatic binding information, a copy of the automatic binding information is stored on a remote host called the DHCP database agent. The bindings are periodically written to the database agent. When the device reloads, the bindings are read from the database agent to the DHCP database in the DHCP server.

**Note**

We strongly recommend that you use database agents. However, Cisco DHCP server can function even without database agents.

Some DHCP clients send a client identifier (DHCP option 61) in the DHCP packet. To configure manual bindings for such clients, you must enter the client-identifier command with the hexadecimal values that identify the DHCP client. To configure manual bindings for clients that do not send a client identifier option, you must enter the hardware-address DHCP pool configuration command with the hexadecimal hardware address of the client.

Depending on your release, the DHCP server sends infinite lease time to the clients for which manual bindings are configured.

Depending on your release, the DHCP server sends lease time that is configured using the lease command to clients for which manual bindings are configured.

**Note**

You cannot configure manual bindings within the same pool that is configured with the network command in DHCP pool configuration mode. See the Configuring DHCP Address Pools section for information about DHCP address pools and the network command.

DHCP Static Mapping

The DHCP Static Mapping feature enables the assignment of static IP addresses (without creating numerous host pools with manual bindings) by using a customer-created text file that the DHCP server reads. The benefit of this feature is that it eliminates the need for a long configuration file and reduces the space required in NVRAM to maintain address pools.

A DHCP database contains the mappings between a client IP address and the hardware address, which is referred to as a binding. There are two types of bindings: manual bindings that map a single hardware address to a single IP address, and automatic bindings that dynamically map a hardware address to an IP address from a pool of IP addresses. Manual (also known as static) bindings can be configured individually directly on the device or by using the DHCP Static Mapping feature. These static bindings can be read from a separate static mapping text file. The static mapping text files are read when a device reloads or the DHCP service restarts. These files are read-only.

The read static bindings are treated just like the manual bindings, in that they are:

- Retained across DHCPRELEASEs from the clients.
- Not timed out.
- Deleted only upon deletion of the pool.
- Provided appropriate exclusions for the contained addresses, which are created at the time of the read.

Just like automatic bindings, manual (or static) bindings from the static mapping text file are also displayed by using the **show ip dhcp binding** command.

To create a static mapping text file, input your addresses in the text file that is stored in the DHCP database for the DHCP server to read. There is no limit to the number of addresses that can be stored in the file. The file format has the following elements:

- Database version number
- End-of-file designator
- Hardware type
- Hardware address
- IP address
- Lease expiration
- Time the file was created

The following is a sample static mapping text file:

```
*time* Jan 21 2005 03:52 PM
*version* 2
!IP address      Type      Hardware address      Lease expiration
10.0.0.4 /24     1         0090.bff6.081e        Infinite
10.0.0.5 /28     id        00b7.0813.88f1.66     Infinite
10.0.0.2 /21     1         0090.bff6.081d        Infinite
*end*
```

Table 59-1 Static Mapping Text File Field Descriptions

Field	Description
time	Specifies the time the file was created. This field allows DHCP to differentiate between the new and old database versions when multiple agents are configured. The valid format of the time is mm dd yyyy hh:mm AM/PM.
version 2	Specifies the database version number.
IP address	Specifies the static IP address. If the subnet mask is not specified, a mask is automatically assigned depending on the IP address. The IP address and the mask is separated by a space.
Type	Specifies the hardware type. For example, type “1” indicates Ethernet. The type “id” indicates that the field is a DHCP client identifier. Legal values can be found online at http://www.iana.org/assignments/arp-parameters in the “Number Hardware Type” list.
Hardware address	<p>Specifies the hardware address.</p> <p>When the type is numeric, the type refers to the hardware media. Legal values can be found online at http://www.iana.org/assignments/arp-parameters in the “Number Hardware Type” list.</p> <p>When the type is “id,” the type refers to a match on the client identifier.</p> <p>For more information about the client identifier, see RFC 2132, DHCP Options and BOOTP Vendor Extensions, section 9.14, located at http://www.ietf.org/rfc/rfc2132.txt, or the client-identifier command.</p> <p>If you are unsure about the client identifier to match with the hardware type, use the debug dhcp detail command to display the client identifier being sent to the DHCP server from the client.</p>

DHCP Server Operation

By default, the DHCP server pings a pool address twice before assigning a particular address to a requesting client. If the ping is unanswered, the DHCP server assumes (with a high probability) that the address is not in use and assigns the address to the requesting client.

By default, the DHCP server waits for 2 seconds before timing out a ping packet.

You can configure the DHCP server to ignore and not reply to any BOOTP requests that the server receives. This functionality is beneficial when there is a mix of BOOTP and DHCP clients in a network segment and there is a BOOTP server and a Cisco DHCP server servicing the network segment. The BOOTP server is configured with static bindings for the BOOTP clients and the BOOTP clients must obtain their addresses from the BOOTP server. However, DHCP servers can also respond to BOOTP requests and the DHCP server may offer an address that causes the BOOTP clients to boot with the address from the DHCP server, instead of the address from the BOOTP server. Configuring the DHCP server to ignore BOOTP requests ensures that the BOOTP clients will receive address information from the BOOTP server and will not accept an address from a DHCP server.

Cisco software can forward these ignored BOOTP request packets to another DHCP server if the **ip helper-address** command is configured on the incoming interface.

Static Route with the Next-Hop Dynamically Obtained Through DHCP

Static routes are updated in the routing table when the default gateway is assigned by the DHCP server. These routes remain in the routing table until the DHCP lease expires and then the routes are removed.

When a DHCP client releases an address, the corresponding static route (the route configured using the **ip route** command) is automatically removed from the routing table. If the DHCP router option (option 3 of the DHCP packet) changes during the client renewal, the DHCP default gateway changes to the new IP address supplied after the renewal.

This feature is particularly useful for VPN deployments such as Dynamic Multipoint VPNs (DMVPNs). It is useful when a nonphysical interface, such as a multipoint generic routing encapsulation (mGRE) tunnel, is configured on a device and certain traffic must be excluded from entering the tunnel interface.



Note

- If the DHCP client is not able to obtain an IP address or the default device IP address, the static route is not installed in the routing table.
- If the lease has expired and the DHCP client cannot renew the address, the DHCP IP address assigned to the client is released and any associated static routes are removed from the routing table

How to Configure the Cisco IOS DHCP Server

- [Configuring a DHCP Database Agent or Disabling Conflict Logging, page 59-10](#)
- [Excluding IP Addresses, page 59-10](#)
- [Configuring Manual Bindings, page 59-17](#)
- [Configuring the DHCP Server to Read a Static Mapping Text File, page 59-18](#)
- [Customizing DHCP Server Operation, page 59-19](#)
- [Configuring a Remote Device to Import DHCP Server Options from a Central DHCP Server, page 59-19](#)
- [Configuring DHCP Address Allocation Using Option 82, page 59-21](#)
- [Configuring Static Route with the Next-Hop Dynamically Obtained Through DHCP, page 59-23](#)
- [Clearing DHCP Server Variables, page 59-24](#)

Configuring a DHCP Database Agent or Disabling Conflict Logging

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	Switch# configure terminal	Enters global configuration mode.
Step 3	Switch(config)# ip dhcp database url [timeout seconds write-delay seconds] or Switch(config)# no ip dhcp conflict logging	Configures a DHCP server to save automatic bindings on a remote host called a database agent. Or Disables DHCP address conflict logging.
Step 4	Switch(config)# exit	Exits global configuration mode and returns to privileged EXEC mode.

Excluding IP Addresses

The IP address configured on a device interface is automatically excluded from the DHCP address pool. The DHCP server assumes that all other IP addresses in a DHCP address pool subnet are available for assigning to DHCP clients.

You must exclude addresses from the pool if the DHCP server does not allocate those IP addresses to DHCP clients. Consider a scenario where two DHCP servers are set up for the same network segment (subnet) for redundancy. If DHCP servers do not coordinate their services with each other using a protocol such as DHCP failover, each DHCP server must be configured to allocate addresses from a nonoverlapping set of addresses in the shared subnet. See the Example: Configuring Manual Bindings section for a configuration example.

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	Switch# configure terminal	Enters global configuration mode.
Step 3	Switch(config)# ip dhcp excluded-address <i>low-address</i> [<i>high-address</i>]	Specifies IP addresses that the DHCP server should not assign to DHCP clients.
Step 4	Switch(config)# exit	Exits global configuration mode and returns to privileged EXEC mode.

Configuring DHCP Address Pools

- [Configuring a DHCP Address Pool, page 59-11](#)
- [Configuring a DHCP Address Pool with Secondary Subnets, page 59-13](#)
- [Troubleshooting Tips, page 59-15](#)
- [Verifying the DHCP Address Pool Configuration, page 59-16](#)

Configuring a DHCP Address Pool

On a per-address pool basis, specify DHCP options for the client as necessary.

Before you configure the DHCP address pool, you must:

- Identify DHCP options for devices where necessary, including the following:
 - Default boot image name
 - Default devices
 - Domain Name System (DNS) servers
 - Network Basic Input/Output System (NetBIOS) name server
 - Primary subnet
 - Secondary subnets and subnet-specific default device lists (see Configuring a DHCP Address Pool with Secondary Subnets section for information on secondary subnets).
- Decide on a NetBIOS node type (b, p, m, or h).
- Decide on a DNS domain name.



Note

You cannot configure manual bindings within the same pool that is configured with the network DHCP pool configuration command. To configure manual bindings, see the Configuring Manual Bindings section.

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	Switch# configure terminal	Enters global configuration mode.
Step 3	Switch(config)# ip dhcp pool <i>name</i>	Assigns a name to a DHCP pool and enters DHCP configuration mode.
Step 4	Switch(dhcp-config)# utilization mark <i>high percentage-number</i> [log]	(Optional) Configures the high utilization mark of the current address pool size. <ul style="list-style-type: none"> • The log keyword enables the logging of a system message. A system message will be generated for a DHCP pool when the pool utilization exceeds the configured high utilization threshold.
Step 5	Switch(dhcp-config)# utilization mark low <i>percentage-number</i> [log]	(Optional) Configures the low utilization mark of the current address pool size. <ul style="list-style-type: none"> • The log keyword enables the logging of a system message. A system message will be generated for a DHCP pool when the pool utilization falls below the configured low utilization threshold.
Step 6	Switch(dhcp-config)# network <i>network-number</i> [<i>mask</i> <i>/prefix-length</i>] [secondary]	Specifies the subnet network number and mask of the DHCP address pool.
Step 7	Switch(dhcp-config)# domain-name <i>domain</i>	Specifies the domain name for the client.

	Command or Action	Purpose
Step 8	Switch(dhcp-config)# dns-server <i>address</i> [<i>address2</i> ... <i>address8</i>]	Specifies the IP address of a DNS server that is available to a DHCP client. <ul style="list-style-type: none"> One IP address is required; however, you can specify up to eight IP addresses in one command. Servers should be listed in order of preference.
Step 9	Switch(dhcp-config)# bootfile <i>filename</i>	(Optional) Specifies the name of the default boot image for a DHCP client. <ul style="list-style-type: none"> The boot file is used to store the boot image for the client. The boot image is generally the operating system that the client uses to load.
Step 10	Switch(dhcp-config)# next-server <i>address</i> [<i>address2</i> ... <i>address8</i>]	(Optional) Configures the next server in the boot process of a DHCP client. <ul style="list-style-type: none"> One address is required; however, you can specify up to eight addresses in one command line. If multiple servers are specified, DHCP assigns them to clients in a round-robin order. The first client gets address 1, the next client gets address 2, and so on. If this command is not configured, DHCP uses the server specified by the ip helper address command as the boot server.
Step 11	Switch(dhcp-config)# netbios-name-server <i>address</i> [<i>address2</i> ... <i>address8</i>]	(Optional) Specifies the NetBIOS WINS server that is available to a Microsoft DHCP client. <ul style="list-style-type: none"> One address is required; however, you can specify up to eight addresses in one command line. Servers should be listed in order of preference.
Step 12	Switch(dhcp-config)# netbios-node-type <i>type</i>	(Optional) Specifies the NetBIOS node type for a Microsoft DHCP client.
Step 13	Switch(dhcp-config)# default-router <i>address</i> [<i>address2</i> ... <i>address8</i>]	(Optional) Specifies the IP address of the default device for a DHCP client. <ul style="list-style-type: none"> The IP address should be on the same subnet as the client. One IP address is required; however, you can specify up to eight IP addresses in one command line. These default devices are listed in order of preference; that is, address is the most preferred device, address2 is the next most preferred device, and so on. When a DHCP client requests an IP address, the device—acting as a DHCP server—accesses the default device list to select another device that the DHCP client will use as the first hop for forwarding messages. After a DHCP client has booted, the client begins sending packets to its default device.
Step 14	Switch(dhcp-config)# option <i>code</i> [<i>instance number</i>] { <i>ascii string</i> <i>hex string</i> <i>ip-address</i> }	(Optional) Configures DHCP server options.

	Command or Action	Purpose
Step 15	Switch(dhcp-config)# lease {days [hours [minutes]] infinite }	Optional) Specifies the duration of the lease. <ul style="list-style-type: none"> The default is a one-day lease. The infinite keyword specifies that the duration of the lease is unlimited.
Step 16	Switch(dhcp-config)# end	Exits DHCP configuration mode and returns to privileged EXEC mode.

Configuring a DHCP Address Pool with Secondary Subnets

For any DHCP pool, you can configure a primary subnet and any number of secondary subnets.

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	Switch# configure terminal	Enters global configuration mode.
Step 3	Switch(config)# ip dhcp pool name	Assigns a name to a DHCP pool and enters DHCP configuration mode.
Step 4	Switch(dhcp-config)# utilization mark high percentage-number [log]	(Optional) Configures the high utilization mark of the current address pool size. <ul style="list-style-type: none"> The log keyword enables the logging of a system message. A system message will be generated for a DHCP pool when the pool utilization exceeds the configured high utilization threshold.
Step 5	Switch(dhcp-config)# utilization mark low percentage-number [log]	(Optional) Configures the low utilization mark of the current address pool size. <ul style="list-style-type: none"> The log keyword enables the logging of a system message. A system message will be generated for a DHCP pool when the pool utilization falls below the configured low utilization threshold.
Step 6	Switch(dhcp-config)# network network-number [mask /prefix-length] [secondary]	Specifies the subnet network number and mask of the DHCP address pool.
Step 7	Switch(dhcp-config)# domain-name domain	Specifies the domain name for the client.
Step 8	Switch(dhcp-config)# dns-server address [address2 ... address8]	Specifies the IP address of a DNS server that is available to a DHCP client. <ul style="list-style-type: none"> One IP address is required; however, you can specify up to eight IP addresses in one command. Servers should be listed in order of preference.
Step 9	Switch(dhcp-config)# bootfile filename	(Optional) Specifies the name of the default boot image for a DHCP client. <ul style="list-style-type: none"> The boot file is used to store the boot image for the client. The boot image is generally the operating system that the client uses to load.

Command or Action	Purpose
Step 10 Switch(dhcp-config)# next-server <i>address</i> [<i>address2</i> ... <i>address8</i>]	(Optional) Configures the next server in the boot process of a DHCP client. <ul style="list-style-type: none"> One address is required; however, you can specify up to eight addresses in one command line. If multiple servers are specified, DHCP assigns them to clients in a round-robin order. The first client gets address 1, the next client gets address 2, and so on. If this command is not configured, DHCP uses the server specified by the ip helper address command as the boot server.
Step 11 Switch(dhcp-config)# netbios-name-server <i>address</i> [<i>address2</i> ... <i>address8</i>]	(Optional) Specifies the NetBIOS WINS server that is available to a Microsoft DHCP client. <ul style="list-style-type: none"> One address is required; however, you can specify up to eight addresses in one command line. Servers should be listed in order of preference.
Step 12 Switch(dhcp-config)# netbios-node-type <i>type</i>	(Optional) Specifies the NetBIOS node type for a Microsoft DHCP client.
Step 13 Switch(dhcp-config)# default-router <i>address</i> [<i>address2</i> ... <i>address8</i>]	(Optional) Specifies the IP address of the default device for a DHCP client. <ul style="list-style-type: none"> The IP address should be on the same subnet as the client. One IP address is required; however, you can specify up to eight IP addresses in one command line. These default devices are listed in order of preference; that is, address is the most preferred device, address2 is the next most preferred device, and so on. When a DHCP client requests an IP address, the device—acting as a DHCP server—accesses the default device list to select another device that the DHCP client will use as the first hop for forwarding messages. After a DHCP client has booted, the client begins sending packets to its default device.
Step 14 Switch(dhcp-config)# option <i>code</i> [<i>instance number</i>] { <i>ascii string</i> <i>hex string</i> <i>ip-address</i> }	(Optional) Configures DHCP server options.
Step 15 Switch(dhcp-config)# lease { <i>days</i> [<i>hours</i> [<i>minutes</i>]] infinite }	(Optional) Specifies the duration of the lease. <ul style="list-style-type: none"> The default is a one-day lease. The infinite keyword specifies that the duration of the lease is unlimited.

	Command or Action	Purpose
Step 16	Switch(dhcp-config)# network <i>network-number [mask /prefix-length]</i> [secondary]	(Optional) Specifies the network number and mask of a secondary DHCP server address pool. <ul style="list-style-type: none"> Any number of secondary subnets can be added to a DHCP server address pool. During execution of this command, the configuration mode changes to DHCP pool secondary subnet configuration mode, which is identified by (config-dhcp-subnet-secondary)# prompt. In this mode, the administrator can configure a default device list that is specific to the subnet. See Troubleshooting Tips section if you are using secondary IP addresses under a loopback interface with DHCP secondary subnets.
Step 17	Switch(config-dhcp-subnet-secondary)# override default-router <i>address [address2 ... address8]</i>	(Optional) Specifies the default device list that is used when an IP address is assigned to a DHCP client from a particular secondary subnet. <ul style="list-style-type: none"> If the subnet-specific override value is configured, this override value is used when assigning an IP address from the subnet; the network-wide default device list is used only to set the gateway device for the primary subnet. If this subnet-specific override value is not configured, the network-wide default device list is used when assigning an IP address from the subnet.
Step 18	Switch(config-dhcp-subnet-secondary)# override utilization high <i>percentage-number</i>	(Optional) Sets the high utilization mark of the subnet size. <ul style="list-style-type: none"> This command overrides the global default setting specified by the utilization mark high command.
Step 19	Switch(config-dhcp-subnet-secondary)# override utilization low <i>percentage-number</i>	(Optional) Sets the low utilization mark of the subnet size. <ul style="list-style-type: none"> This command overrides the global default setting specified by the utilization mark low command.
Step 20	Switch(config-dhcp-subnet-secondary)# end	Exits DHCP secondary subnet configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

If you are using secondary IP addresses under a single loopback interface and using secondary subnets under a DHCP pool, use one DHCP pool to configure networks for all the secondary subnets instead of using one pool per secondary subnet. The **network** *network-number [mask | /prefix-length] [secondary]* command must be configured under a single DHCP address pool rather than multiple DHCP address pools.

The following is the correct configuration:

```
ip dhcp pool dhcp_1
 network 172.16.1.0 255.255.255.0
 network 172.16.2.0 255.255.255.0 secondary
 network 172.16.3.0 255.255.255.0 secondary
 network 172.16.4.0 255.255.255.0 secondary
!
interface Loopback111
 ip address 172.16.1.1 255.255.255.255 secondary
```

```
ip address 172.16.2.1 255.255.255.255 secondary
ip address 172.16.3.1 255.255.255.255 secondary
ip address 172.16.4.1 255.255.255.255 secondary
```



The following is the incorrect configuration:

```
ip dhcp pool dhcp_1
network 172.16.1.0 255.255.255.0
lease 1 20 30
accounting default
!
ip dhcp pool dhcp_2
network 172.16.2.0 255.255.255.0
lease 1 20 30
accounting default
!
ip dhcp pool dhcp_3
network 172.16.3.0 255.255.255.0
lease 1 20 30
accounting default
!
ip dhcp pool dhcp_4
network 172.16.4.0 255.255.255.0
lease 1 20 30
accounting default
!
interface Loopback111
ip address 172.16.1.1 255.255.255.255 secondary
ip address 172.16.2.1 255.255.255.255 secondary
ip address 172.16.3.1 255.255.255.255 secondary
ip address 172.16.4.1 255.255.255.255 secondary
```

Verifying the DHCP Address Pool Configuration

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	Switch# show ip dhcp pool [<i>name</i>]	(Optional) Displays information about DHCP address pools.
Step 3	Switch(config)# show ip dhcp binding [<i>address</i>]	(Optional) Displays a list of all bindings created on a specific DHCP server. <ul style="list-style-type: none"> Use the show ip dhcp binding command to display the IP addresses that have already been assigned. Verify that the address pool is not exhausted. If necessary, recreate the pool to create a larger pool of addresses. Use the show ip dhcp binding command to display the lease expiration date and time of the IP address of the host.
Step 4	Switch(dhcp-config)# show ip dhcp conflict [<i>address</i>]	(Optional) Displays a list of all IP address conflicts.
Step 5	Switch(config)# show ip dhcp database [<i>url</i>]	(Optional) Displays recent activity on the DHCP database.
Step 6	Switch(config)# show ip dhcp server statistics [<i>type-number</i>]	(Optional) Displays count information about server statistics and messages sent and received.

Configuring Manual Bindings

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	Switch# configure terminal	Enters global configuration mode.
Step 3	Switch(dhcp-config)# ip dhcp pool <i>pool-name</i>	Assigns a name to a DHCP pool and enters DHCP configuration mode.
Step 4	Switch(dhcp-config)# host <i>address</i> [<i>mask</i> <i>/prefix-length</i>]	Specifies the IP address and subnet mask of the client. <ul style="list-style-type: none"> There is no limit to the number of manual bindings you can configure. However, you can configure only one manual binding per host pool.
Step 5	Switch(dhcp-config)# client-identifier <i>unique-identifier</i>	Specifies the unique identifier for DHCP clients. <ul style="list-style-type: none"> This command is used for DHCP requests. DHCP clients require client identifiers. You can specify the unique identifier for the client in either of the following ways: <ul style="list-style-type: none"> A 7-byte dotted hexadecimal notation. For example, 01b7.0813.8811.66, where 01 represents the Ethernet media type and the remaining bytes represent the MAC address of the DHCP client. A 27-byte dotted hexadecimal notation. For example, 7665.6e64.6f72.2d30.3032.342e.3937.6230.2e33.3734.312d.4661.302f.31. The equivalent ASCII string for this hexadecimal value is vendor-0024.97b0.3741-fa0/1, where vendor represents the vendor, 0024.97b0.3741 represents the MAC address of the source interface, and fa0/1 represents the source interface of the DHCP client. See the Troubleshooting section for information about how to determine the client identifier of the DHCP client. <div>  <p>Note The identifier specified here is considered for a DHCP client that sends a client identifier in the packet.</p> </div>
Step 6	Switch(dhcp-config)# hardware-address <i>hardware-address</i> [<i>protocol-type</i> <i>hardware-number</i>]	Specifies a hardware address for the client. <ul style="list-style-type: none"> This command is used for BOOTP requests. <div>  <p>Note The hardware address specified here is considered for a DHCP client that does not send a client identifier in the packet.</p> </div>

	Command or Action	Purpose
Step 7	Switch(dhcp-config)# client-name <i>name</i>	(Optional) Specifies the name of the client using any standard ASCII character. <ul style="list-style-type: none"> The client name should not include the domain name. For example, the name client1 should not be specified as client1.cisco.com.
Step 8	Switch(dhcp-config)# end	Exits DHCP configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

You can determine the client identifier by using the **debug ip dhcp server packet** command. In the following sample output, the client is identified by the value 0b07.1134.a029:

```
Switch# debug ip dhcp server packet
```

```
DHCPD:DHCPDISCOVER received from client 0b07.1134.a029 through relay 10.1.0.253.
DHCPD:assigned IP address 10.1.0.3 to client 0b07.1134.a029.
.
.
.
```


Configuring the DHCP Server to Read a Static Mapping Text File

Before You Begin

The administrator must create the static mapping text file in the correct format and configure the address pools before performing this task.

Before editing the file, you must disable the DHCP server using the **no service dhcp** command.

The static bindings must not be deleted when a DHCPRELEASE is received or must not be timed out by the DHCP timer. The static bindings should be created by using the **ip dhcp pool** command.

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	Switch# configure terminal	Enters global configuration mode.
Step 3	Switch(config)# ip dhcp pool <i>pool-name</i>	Assigns a name to a DHCP pool and enters DHCP configuration mode. <div data-bbox="893 1501 1461 1753">  <p>Note If you have already configured the IP DHCP pool name using the ip dhcp pool command and the static file URL using the origin file command, you must perform a fresh read using the no service dhcp command and the service dhcp command.</p> </div>
Step 4	Switch(dhcp-config)# origin file <i>url</i>	Specifies the URL that the DHCP server can access to locate the text file.

	Command or Action	Purpose
Step 5	Switch(dhcp-config) # end	Exits DHCP configuration mode and returns to privileged EXEC mode.
Step 6	Switch# show ip dhcp binding [address]	(Optional) Displays a list of all bindings created on a specific DHCP server.

Customizing DHCP Server Operation

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	Switch# configure terminal	Enters global configuration mode.
Step 3	Switch(config) # ip dhcp ping packets <i>number</i>	(Optional) Specifies the number of ping packets the DHCP server sends to a pool address before assigning the address to a requesting client. <ul style="list-style-type: none"> The default is two packets. Setting the <i>number</i> argument to a value of 0 disables the DHCP server ping operation.
Step 4	Switch(config) # ip dhcp ping timeout <i>milliseconds</i>	(Optional) Specifies the duration the DHCP server waits for a ping reply from an address pool.
Step 5	Switch(config) # ip dhcp bootp ignore	(Optional) Allows the DHCP server to selectively ignore and not reply to received BOOTP requests. <ul style="list-style-type: none"> The ip dhcp bootp ignore command applies to all DHCP pools configured on the device. BOOTP requests cannot be selectively ignored on a per-DHCP pool basis.
Step 6	Switch(config) # end	Exits global configuration mode and returns to privileged EXEC mode.

Configuring a Remote Device to Import DHCP Server Options from a Central DHCP Server

The Cisco DHCP server can dynamically configure options such as the Domain Name System (DNS) and Windows Internet Name Service (WINS) addresses to respond to DHCP requests from local clients behind the customer premises equipment (CPE). Earlier, network administrators configured the Cisco DHCP server on each device manually. Now, the Cisco DHCP server is enhanced to allow configuration information to be updated automatically. Network administrators can configure one or more centralized DHCP servers to update specific DHCP options within the DHCP pools. The remote servers can request or “import” these option parameters from centralized servers.

This section contains the following tasks:

- [Configuring the Central DHCP Server to Update DHCP Options, page 59-20](#)
- [Configuring a Remote Device to Import DHCP Server Options from a Central DHCP Server, page 59-19](#)

Configuring the Central DHCP Server to Update DHCP Options

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	Switch# configure terminal	Enters global configuration mode.
Step 3	Switch(config)# ip dhcp pool <i>name</i>	Assigns a name to a DHCP pool and enters DHCP configuration mode.
Step 4	Switch(dhcp-config)# network <i>network-number</i> [<i>mask</i> <i>/prefix-length</i>]	Specifies the subnet number and mask of the DHCP address pool.
Step 5	Switch(dhcp-config)# dns-server <i>address</i> [<i>address2</i> ... <i>address8</i>]	(Optional) Specifies the IP address of a DNS server that is available to a DHCP client. <ul style="list-style-type: none"> One IP address is required; however, you can specify up to eight IP addresses in one command line. Servers should be listed in the order of preference.
Step 6	Switch(dhcp-config)# end	Exits DHCP configuration mode and returns to privileged EXEC mode.

Configuring the Remote Device to Import DHCP Options

Perform the following task to configure the remote device to import DHCP options:



Note

When two servers provide DHCP addresses to a single device configured with `ip address dhcp` on two different interfaces, the imported information is merged and, for those options that take a single value, the last known option value will be used.

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	Switch# configure terminal	Enters global configuration mode.
Step 3	Switch(config)# ip dhcp pool <i>name</i>	Assigns a name to a DHCP pool and enters DHCP configuration mode.
Step 4	Switch(dhcp-config)# network <i>network-number</i> [<i>mask</i> <i>/prefix-length</i>]	Specifies the subnet number and mask of the DHCP address pool.
Step 5	Switch(dhcp-config)# import all	(Optional) Specifies the IP address of a DNS server that is available to a DHCP client. <ul style="list-style-type: none"> One IP address is required; however, you can specify up to eight IP addresses in one command line. Servers should be listed in the order of preference.
Step 6	Switch(dhcp-config)# exit	Exits DHCP configuration mode and returns to privileged EXEC mode.
Step 7	Switch(config)# interface <i>type number</i>	Configures an interface and enters interface configuration mode.
Step 8	Switch(config-if)# ip address dhcp	Specifies that the interface acquires an IP address through DHCP.

	Command or Action	Purpose
Step 9	Switch(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode.
Step 10	Switch# show ip dhcp import	Displays the options that are imported from the central DHCP server.

Configuring DHCP Address Allocation Using Option 82

- [Enabling Option 82 for DHCP Address Allocation, page 59-21](#)
- [Defining the DHCP Class and Relay Agent Information Patterns, page 59-21](#)

Enabling Option 82 for DHCP Address Allocation

By default, the Cisco DHCP server uses information provided by option 82 to allocate IP addresses. If the DHCP address allocation is disabled, perform the task described in this section to reenable this capability.

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	Switch# configure terminal	Enters global configuration mode.
Step 3	Switch(config)# ip dhcp use class	Controls DHCP classes that are used for address allocation. <ul style="list-style-type: none"> • This functionality is enabled by default. • Use the no form of this command to disable this functionality without deleting the DHCP class configuration.
Step 4	Switch(config)# end	Exits global configuration mode and returns to privileged EXEC mode.

Troubleshooting Tips

If DHCP classes are configured in the pool, but the DHCP server does not use the classes, verify if the **no ip dhcp use class** command was configured.

Defining the DHCP Class and Relay Agent Information Patterns

Before You Begin

You must know the hexadecimal value of each byte location in option 82 to configure the relay-information hex command. The option 82 format may vary from product to product. Contact the relay agent vendor for this information.

Perform this task to define the DHCP class and relay agent information patterns.

■ Configuring DHCP Address Pools

Step 1	Switch> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	Switch# configure terminal	Enters global configuration mode.
Step 3	Switch(config)# ip dhcp class name	Defines a DHCP class and enters DHCP class configuration mode.
Step 4	Switch(dhcp-class)# relay agent information	Enters relay agent information option configuration mode. • If you omit this step, the DHCP class matches any relay agent information option, whether the relay agent information option value is available or not.
Step 5	Switch(dhcp-class-relayinfo)# relay-information hex pattern [*] [bitmask mask]	(Optional) Specifies a hexadecimal value for full relay information option. • The pattern argument creates a pattern that is used to match the DHCP class. • If you omit this step, no pattern is configured and it is considered a match to any relay agent information option value, but the relay information option must be available in the DHCP packet. • You can configure multiple relay-information hex commands in a DHCP class.
Step 6	Repeat Steps 3 through 5 for each DHCP class you need to configure.	—
Step 7	Switch(dhcp-class-relayinfo)# end	Exits relay agent information option mode and returns to privileged EXEC mode.

Troubleshooting Tips

Use the **debug ip dhcp server class** command to display the class matching results.

Defining the DHCP Address Pool

Step 1	Switch> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	Switch# configure terminal	Enters global configuration mode.
Step 3	Switch(config)# ip dhcp pool name	Assigns a name to a DHCP pool and enters DHCP configuration mode. • Multiple pools can be configured with the same class, eliminating the need to configure the same pattern in multiple pools.
Step 4	Switch(dhcp-config)# network network-number [mask /prefix-length]	Configures the subnet and mask for a DHCP address pool on a Cisco IOS DHCP server.

Step 5	Switch(dhcp-config)# class <i>class-name</i>	Associates a class with a pool and enters DHCP pool class configuration mode. <ul style="list-style-type: none"> This command also creates a DHCP class if the DHCP class is not yet defined.
Step 6	Switch(dhcp-pool-class)# address range <i>start-ip end-ip</i>	(Optional) Sets an address range for the DHCP class in a DHCP server address pool. <ul style="list-style-type: none"> If this command is not configured for a class, the default value is the entire subnet of the pool. Each class in the DHCP pool is examined for a match in the order configured.
Step 7	Repeat Steps 5 and 6 for each DHCP class you need to associate with the DHCP pool.	—
Step 8	Switch(dhcp-pool-class)# end	Exits DHCP pool class option mode and returns to privileged EXEC mode.

Configuring Static Route with the Next-Hop Dynamically Obtained Through DHCP

This task enables static routes to be assigned using a DHCP default gateway as the next-hop device. Without this feature the gateway IP address is not known until after the DHCP address assignment. You cannot configure a static route with the CLI without knowing that DHCP-supplied address.

Before You Begin

Verify all DHCP client and server configuration steps. Ensure that the DHCP client and server are properly defined to supply a DHCP device option 3 of the DHCP packet.

Step 1	Switch> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	Switch# configure terminal	Enters global configuration mode.
Step 3	Switch(config)# ip route <i>prefix mask</i> { <i>ip-address</i> <i>interface-type interface-number</i> [<i>ip-address</i>]} dhcp [<i>distance</i>]	Assigns a static route for the default next-hop device when the DHCP server is accessed for an IP address. <ul style="list-style-type: none"> If more than one interface is configured to obtain an IP address from a DHCP server, use the ip route prefix mask interface-type interface-number dhcp command for each interface. If the interface is not specified, the route is added to the routing table as soon as any of the interfaces obtain an IP address and a default device.
Step 4	Switch(config)# end	Exits global configuration mode and returns to privileged EXEC mode.
Step 5	Switch# show ip route	(Optional) Displays the current state of the routing table.

Clearing DHCP Server Variables

Step 1	Switch> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	Switch# clear ip dhcp binding {address *}	Deletes an automatic address binding from the DHCP database. <ul style="list-style-type: none"> Specifying the address argument clears the automatic binding for a specific (client) IP address, whereas specifying an asterisk (*) clears all automatic bindings.
Step 3	Switch# clear ip dhcp conflict {address *}	Clears an address conflict from the DHCP database. <ul style="list-style-type: none"> Specifying the address argument clears the conflict for a specific IP address, whereas specifying an asterisk (*) clears conflicts for all addresses.
Step 4	Switch# clear ip dhcp server statistics	Resets all DHCP server counters to 0.

Configuration Examples for the Cisco IOS DHCP Server

- [Example: Configuring a DHCP Database Agent or Disabling Conflict Logging, page 59-24](#)
- [Example: Excluding IP Addresses, page 59-24](#)
- [Example: Configuring a DHCP Address Pool, page 59-25](#)
- [Example: Configuring Manual Bindings, page 59-27](#)
- [Example: Configuring Static Mapping, page 59-28](#)
- [Example: Customizing DHCP Server Operation, page 59-29](#)
- [Example: Configuring a Remote Device to Import DHCP Server Options from a Central DHCP Server, page 59-30](#)
- [Example: Configuring DHCP Address Allocation Using Option 82, page 59-31](#)
- [Example: Configuring a Static Route with the Next-Hop Dynamically Obtained Through DHCP, page 59-32](#)

Example: Configuring a DHCP Database Agent or Disabling Conflict Logging

The following example shows how to store bindings on host 172.16.4.253. The file transfer protocol is FTP. The server waits for 2 minutes (120 seconds) before performing database changes.

```
Switch> enable
Switch# configure terminal
Switch(config)# ip dhcp database ftp://user:password@172.16.4.253/router-dhcp write-delay 120
Switch(config)# exit
```

Example: Excluding IP Addresses

In the following example, server A and server B service the subnet 10.0.20.0/24. If the subnet is split equally between the two servers, server A is configured to allocate IP addresses 10.0.20.1 to 10.0.20.125 and server B is configured to allocate IP addresses 10.0.20.126 to 10.0.20.254.

Server A

```
Switch# configure terminal
switch(config)# ip dhcp excluded-address 10.0.20.126 10.0.20.255
Switch(config)# ip dhcp pool A
Switch(dhcp-config)# network 10.0.20.0 255.255.255.0
```

Server B

```
Switch# configure terminal
switch(config)# ip dhcp excluded-address 10.0.20.0 10.0.20.125
Switch(config)# ip dhcp pool B
Switch(dhcp-config)# network 10.0.20.0 255.255.255.0
```

Example: Configuring a DHCP Address Pool

In the following example, three DHCP address pools are created: one in network 172.16.0.0, one in subnetwork 172.16.1.0, and one in subnetwork 172.16.2.0. Attributes from network 172.16.0.0—such as the domain name, Domain Name System (DNS) server, (Network Basic Input/Output System) NetBIOS name server, and NetBIOS node type—are inherited in subnetworks 172.16.1.0 and 172.16.2.0. In each pool, clients are granted 30-day leases and all addresses in each subnetwork, except the excluded addresses, are available to the DHCP server for assigning to clients. The table below lists the IP addresses for the devices in three DHCP address pools.

Table 59-2 *DHCP Address Pool Configuration*

	Pool 0 (Network 172.16.0.0)	Pool 1 (Subnetwork 172.16.1.0)	Pool 2 (Subnetwork 172.16.2.0)
Device Type	IP Address	IP Address	IP Address
Default Devices		172.16.1.100 172.16.1.101	172.16.2.100 172.16.2.101
DNS Server	172.16.1.102 172.16.2.102		
NetBIOS Name Server	172.16.1.103 172.16.2.103		
NetBIOS Node Type	h-node		

```
Switch(config)# ip dhcp database ftp://user:password@172.16.4.253/router-dhcp write-delay 120
Switch(config)# ip dhcp excluded-address 172.16.1.100 172.16.1.103
Switch(config)# ip dhcp excluded-address 172.16.2.100 172.16.2.103
!
Switch(config)# ip dhcp pool 0
Switch(dhcp-config)# network 172.16.0.0 /16
Switch(dhcp-config)# domain-name cisco.com
Switch(dhcp-config)# dns-server 172.16.1.102 172.16.2.102
Switch(dhcp-config)# netbios-name-server 172.16.1.103 172.16.2.103
Switch(dhcp-config)# netbios-node-type h-node
!
Switch(config)# ip dhcp pool 1
Switch(dhcp-config)# network 172.16.1.0 /24
Switch(dhcp-config)# default-router 172.16.1.100 172.16.1.101
Switch(dhcp-config)# lease 30
```

```

!
Switch(config)# ip dhcp pool 2
Switch(dhcp-config)# network 172.16.2.0 /24
Switch(dhcp-config)# default-router 172.16.2.100 172.16.2.101
Switch(dhcp-config)# lease 30

```

Example: Configuring a DHCP Address Pool with Multiple Disjoint Subnets

Multiple disjoint subnets in a DHCP pool can be used in any of the following network topologies:

- IP address pooling—The DHCP client and server reside on the same subnet.
- DHCP relay—The DHCP client and DHCP server communicate through a DHCP relay agent where the relay interface is configured with secondary IP addresses.
- Hierarchical DHCP—The DHCP server is configured as the DHCP subnet allocation server. The DHCP client and DHCP subnet allocation server communicate through an on-demand address pool (ODAP) router.

In the following example, one DHCP address pool named pool3 is created; the primary subnet is 172.16.0.0/16, one secondary subnet is 172.16.1.0/24, and the other secondary subnet is 172.16.2.0/24.

- When IP addresses in the primary subnet are exhausted, the DHCP server inspects the secondary subnets in the order in which the subnets were added to the pool.
- When the DHCP server allocates an IP address from the secondary subnet 172.16.1.0/24, the server uses the subnet-specific default device list that consists of IP addresses 172.16.1.100 and 172.16.1.101. However, when the DHCP server allocates an IP address from the subnet 172.16.2.0/24, the server uses the pool-wide list that consists of the four IP addresses from 172.16.0.100 to 172.16.0.103.
- Other attributes from the primary subnet 172.16.0.0/16—such as the domain name, DNS server, NetBIOS name server, and NetBIOS node type—are inherited in both the secondary subnets.
- DHCP clients are granted 30-day leases on IP addresses in the pool. All addresses in each subnet, except the excluded addresses, are available to the DHCP server for assigning to clients.

The table below lists the IP addresses for the devices in the DHCP address pool that consists of three disjoint subnets.

Table 59-3 **DHCP Address Pool Configuration**

	Primary Subnet (172.16.0.0/16)	First Secondary Subnet (172.16.1.0/24)	Second Secondary Subnet (172.16.2.0/24)
Device Type	IP Address	IP Address	IP Address
Default Devices	172.16.0.100	172.16.1.100	172.16.0.100
	172.16.0.101	172.16.1.101	172.16.0.101
	172.16.0.102		172.16.0.102
	172.16.0.103		172.16.0.103
DNS Server	172.16.1.102		
	172.16.2.102		

	Primary Subnet (172.16.0.0/16)	First Secondary Subnet (172.16.1.0/24)	Second Secondary Subnet (172.16.2.0/24)
NetBIOS Name Server	172.16.1.103 172.16.2.103		
NetBIOS Node Type	h-node		

```
Switch(config)# ip dhcp database ftp://user:password@172.16.4.253/router-dhcp write-delay
120
Switch(config)# ip dhcp excluded-address 172.16.0.100 172.16.1.103
Switch(config)# ip dhcp excluded-address 172.16.1.100 172.16.1.101
!
Switch(config)# ip dhcp pool pool3
Switch(dhcp-config)# network 172.16.0.0 /16
Switch(dhcp-config)# default-router 172.16.0.100 172.16.2.101 172.16.0.102 172.16.0.103
Switch(dhcp-config)# domain-name cisco.com
Switch(dhcp-config)# dns-server 172.16.1.102 172.16.2.102
Switch(dhcp-config)# netbios-name-server 172.16.1.103 172.16.2.103
Switch(dhcp-config)# netbios-node-type h-node
Switch(dhcp-config)# lease 30
!
Switch(dhcp-config)# network 172.16.1.0 /24 secondary
Switch(dhcp-config)# override default-router 172.16.1.100 172.16.1.101
!
Switch(dhcp-config)# network 172.16.2.0 /24 secondary
```

Example: Configuring Manual Bindings

The following example shows how to create a manual binding for a client named example1.abc.com that sends a client identifier in the DHCP packet. The MAC address of the client is 02c7.f800.0422 and the IP address of the client is 172.16.2.254.

```
Switch(config)# ip dhcp pool pool1
Switch(dhcp-config)# host 172.16.2.254
Switch(dhcp-config)# client-identifier 01b7.0813.8811.66
Switch(dhcp-config)# client-name example1
```

The following example shows how to create a manual binding for a client named example2.abc.com that does not send a client identifier in the DHCP packet. The MAC address of the client is 02c7.f800.0422 and the IP address of the client is 172.16.2.253.

```
Switch(config)# ip dhcp pool pool2
Switch(dhcp-config)# host 172.16.2.253
Switch(dhcp-config)# hardware-address 02c7.f800.0422 ethernet
Switch(dhcp-config)# client-name example1
```

Because attributes are inherited, the two preceding configurations are equivalent to the following:

```
Switch(config)# ip dhcp pool pool1
Switch(dhcp-config)# host 172.16.2.254 255.255.255.0
Switch(dhcp-config)# hardware-address 02c7.f800.0422 ieee802
Switch(dhcp-config)# client-name client1
Switch(dhcp-config)# default-router 172.16.2.100 172.16.2.101
Switch(dhcp-config)# domain-name abc.com
Switch(dhcp-config)# dns-server 172.16.1.102 172.16.2.102
Switch(dhcp-config)# netbios-name-server 172.16.1.103 172.16.2.103
Switch(dhcp-config)# netbios-node-type h-node
```

Example: Configuring Static Mapping

The following example shows how to restart the DHCP server, configure the pool, and specify the URL where the static mapping text file is stored:

```
Switch# configure terminal
Switch(config)# no service dhcp
Switch(config)# service dhcp
Switch(config)# ip dhcp pool pool1
Switch(dhcp-config)# origin file tftp://10.1.0.1/static-bindings
Switch(dhcp-config)# end
```



Note

The static mapping text file can be copied to flash memory on the device and served by the TFTP process of the device. In this case, the IP address in the original file line must be an address owned by the device and one additional line of configuration is required on the device: tftp-server flash static-filename.

The following sample output from the **show ip dhcp binding** command displays address bindings that are configured:

```
Device# show ip dhcp binding

00:05:14:%SYS-5-CONFIG_I: Configured from console by console
Bindings from all pools not associated with VRF:
IP address  Client-ID/          Ls expir   Type    Hw address        User name
10.9.9.4/8   0063.7363.2d30.3036.  Infinite   Static   302e.3762.2e39.3634.  632d.4574.8892.
10.9.9.1/24  0063.6973.636f.2d30.  Infinite   Static   3036.302e.3437.3165.  2e64.6462.342d.
```

The following sample output displays each entry in the static mapping text file

```
*time* Jan 21 2005 22:52 PM
!IP address      Type      Hardware address          Lease expiration
10.19.9.1 /24    id        0063.6973.636f.2d30.3036.302e.3437
10.9.9.4         id        0063.7363.2d30.3036.302e.3762.2e39.3634.632d  Infinite
*end*
```

The following sample debug output shows the reading of the static mapping text file from the TFTP server:

```
Switch# debug ip dhcp server

Loading abc/static_pool from 10.19.192.33 (via Ethernet0):
[OK - 333 bytes]
*May 26 23:14:21.259: DHCPD: contacting agent tftp://10.19.192.33/abc/static_pool (attempt 0)
*May 26 23:14:21.467: DHCPD: agent tftp://10.19.192.33/abc/static_pool is responding.
*May 26 23:14:21.467: DHCPD: IFS is ready.
*May 26 23:14:21.467: DHCPD: reading bindings from tftp://10.19.192.33/abc/static_pool.
*May 26 23:14:21.707: DHCPD: read 333 / 1024 bytes.
*May 26 23:14:21.707: DHCPD: parsing text line
*time* Apr 22 2002 11:31 AM
*May 26 23:14:21.707: DHCPD: parsing text line ""
*May 26 23:14:21.707: DHCPD: parsing text line
!IP address Type Hardware address Lease expiration
*May 26 23:14:21.707: DHCPD: parsing text line
"10.9.9.1 /24 id 0063.6973.636f.2d30.3036.302e.3437"
*May 26 23:14:21.707: DHCPD: creating binding for 10.9.9.1
*May 26 23:14:21.707: DHCPD: Adding binding to radix tree (10.9.9.1)
*May 26 23:14:21.707: DHCPD: Adding binding to hash tree
*May 26 23:14:21.707: DHCPD: parsing text line
"10.9.9.4 id 0063.7363.2d30.3036.302e.3762.2e39.3634.632d"
*May 26 23:14:21.711: DHCPD: creating binding for 10.9.9.4
```

```
*May 26 23:14:21.711: DHCPD: Adding binding to radix tree (10.9.9.4)
*May 26 23:14:21.711: DHCPD: Adding binding to hash tree
*May 26 23:14:21.711: DHCPD: parsing text line "Infinite"
*May 26 23:14:21.711: DHCPD: parsing text line ""
*May 26 23:14:21.711: DHCPD: parsing text line
!IP address Interface-index Lease expiration VRF
*May 26 23:14:21.711: DHCPD: parsing text line "*end*"
*May 26 23:14:21.711: DHCPD: read static bindings from
tftp://10.19.192.33/abcemp/static_pool.
```

Example: Customizing DHCP Server Operation

```
Switch# configure terminal
Switch(config)# ip dhcp ping packets 5
Switch(config)# ip dhcp ping timeout 850
Switch(config)# ip dhcp bootp ignore
Switch(config)# end
```

Example: Configuring the Option to Ignore all BOOTP Requests

The following example shows two DHCP pools that are configured on the device and that the device's DHCP server is configured to ignore all received BOOTP requests. If a BOOTP request is received from subnet 10.0.18.0/24, the request will be dropped by the device (because the `ip helper-address` command is not configured). If there is a BOOTP request from subnet 192.168.1.0/24, the request will be forwarded to 172.16.1.1 via the `ip helper-address` command.

```
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Router
!
ip subnet-zero
!
ip dhcp bootp ignore
!
ip dhcp pool ABC
    network 192.168.1.0 255.255.255.0
    default-router 192.168.1.3
    lease 2
!
ip dhcp pool DEF
    network 10.0.18.0 255.255.255.0
!
ip cef
!
interface FastEthernet0/0
    no ip address
    shutdown
    duplex half
!
interface Ethernet1/0
    ip address 10.0.18.68 255.255.255.0
    duplex half
!
interface Ethernet1/1
    ip address 192.168.1.1 255.255.255.0
    ip helper-address 172.16.1.1
```

```

duplex half
!
interface Ethernet1/2
shutdown
duplex half
!
interface Ethernet1/3
no ip address
shutdown
duplex half
!
interface FastEthernet2/0
no ip address
shutdown
duplex half
!
ip route 172.16.1.1 255.255.255.255 e1/0
no ip http server
no ip pim bidir-enable
!
call rsvp-sync
!
mgcp profile default
!
dial-peer cor custom
!
gatekeeper
shutdown
!
line con 0
line aux 0
line vty 0 4
!
end

```

Example: Configuring a Remote Device to Import DHCP Server Options from a Central DHCP Server

- [Example: Importing DHCP Options, page 59-30](#)
- [Example: Configuring the Remote Device to Import DHCP Options, page 59-31](#)

Example: Importing DHCP Options

The following example shows how to configure a remote and central server to support the importing of DHCP options. The central server is configured to automatically update DHCP options, such as DNS and WINS addresses, within the DHCP pools. In response to a DHCP request from a local client behind CPE equipment, the remote server can request or “import” these option parameters from the centralized server. See the figure below for a diagram of the network topology.

Central Device

```

!do not assign this range to DHCP clients
ip dhcp-excluded address 10.0.0.1 10.0.0.5
!
ip dhcp pool central
! Specifies network number and mask for DHCP clients
network 10.0.0.0 255.255.255.0
! Specifies the domain name for the client

```

```
domain-name central
! Specifies DNS server that will respond to DHCP clients when they need to correlate host
! name to ip address
dns-server 10.0.0.2
! Specifies the NETBIOS WINS server
netbios-name-server 10.0.0.2
!
interface FastEthernet0/0
 ip address 10.0.0.1 255.255.255.0
 duplex auto
 speed auto
```

Remote Device

```
ip dhcp pool client
! Imports DHCP option parameters into DHCP server database
import all
 network 172.16.2.254 255.255.255.0
!
interface FastEthernet0/0
 ip address dhcp
 duplex auto
 speed auto
```

Example: Configuring the Remote Device to Import DHCP Options

```
Switch# configure terminal
Switch(config)# ip dhcp pool 1
Switch(dhcp-config)# network 172.16.0.0 /16
Switch(dhcp-config)# import all
Switch(dhcp-config)# exit
Switch(config)# interface FastEthernet 0/0
Switch(config-if)# ip address dhcp
Switch(config-if)# end
Switch# show ip dhcp import
```

Example: Configuring DHCP Address Allocation Using Option 82

This example shows how to configure two DHCP classes. CLASS1 defines the group of DHCP clients whose address requests contain the relay agent information option with the specified hexadecimal values. CLASS2 defines the group of DHCP clients whose address requests contain the configured relay agent information suboptions. CLASS3 has no pattern configured and is treated as a “match to any” class. This type of class is useful for specifying a “default” class.

The subnet of pool ABC has been divided into three ranges without further subnetting the 10.0.20.0/24 subnet. If there is a DHCP Discover message from the 10.0.20.0/24 subnet with option 82 matching that of class CLASS1, an available address in the range from 10.0.20.1 to 10.0.20.100 will be allocated. If there is no free address in CLASS address range, the DHCP Discover message will be matched against CLASS2, and so on.

Therefore, each class in the DHCP pool will be examined for a match in the order configured by the user. In pool ABC, the order of matching is CLASS1, CLASS2, and finally CLASS3. In pool DEF, class CLASS2 does not have any address range configured. By default, the address range for a particular class is the pool’s entire subnets. Therefore, clients matching CLASS2 may be allocated addresses from 10.0.20.1 to 10.0.20.254.

Multiple pools can be configured with the same class, eliminating the need to configure the same patterns in multiple pools. For example, there may be a need to specify that one or more pools must be used only to service a particular class of devices (for example, cable modems and IP phones).

```
! Defines the DHCP classes and relay information patterns
ip dhcp class CLASS1
  relay agent information
    relay-information hex 01030a0b0c02050000000123
    relay-information hex 01030a0b0c02*
    relay-information hex 01030a0b0c02050000000000 bitmask 00000000000000000000FF
ip dhcp class CLASS2
  relay agent information
    relay-information hex 01040102030402020102
    relay-information hex 01040101030402020102
ip dhcp class CLASS3
  relay agent information
! Associates the DHCP pool with DHCP classes
ip dhcp pool ABC
  network 10.0.20.0 255.255.255.0
  class CLASS1
    address range 10.0.20.1 10.0.20.100
class CLASS2
  address range 10.0.20.101 10.0.20.200
class CLASS3
  address range 10.0.20.201 10.0.20.254
ip dhcp pool DEF
  network 172.64.2.2 255.255.255.0
  class CLASS1
    address range 172.64.2.3 172.64.2.10
  class CLASS2
```

Example: Configuring a Static Route with the Next-Hop Dynamically Obtained Through DHCP

The following example shows how to configure two Ethernet interfaces to obtain the next-hop device IP address from the DHCP server:

```
Switch(config)# ip route 10.10.10.0 255.255.255.0 dhcp 200
Switch(config)# ip route 10.10.20.1 255.255.255.255 ethernet 1 dhcp
```

Additional References for the Cisco IOS DHCP Server

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Catalyst 4500 commands	Cisco IOS <<technology>> Command Reference

Standards & RFCs

Standard/RFC	Title
RFC 951	Bootstrap Protocol (BOOTP)
RFC 1542	Clarifications and Extensions for the Bootstrap Protocol
RFC 2131	Dynamic Host Configuration Protocol
RFC 2132	DHCP Options and BOOTP Vendor Extensions

MIBs

MIB	MIBs Link
<ul style="list-style-type: none">	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for the IOS DHCP Server

Table 4 lists the features in this module and provides links to specific configuration information.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.



Note

Table 4 lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 4 Feature Information for the IOS DHCP Server

Feature Name	Feature Information
DHCP Server Import All Enhancement	The DHCP Server Import All Enhancement feature is an enhancement to the import all command. Prior to this feature, the options imported through the import all command were overwritten by those imported by another subsystem. Through this feature, options imported by multiple subsystems can coexist in the DHCP address pool. When the session is terminated or the lease is released, the imported options are cleared.
DHCP Server Multiple Subnet	The DHCP Server Multiple Subnet feature enables multiple subnets to be configured under the same DHCP address pool. The following commands were introduced or modified: network(DHCP), override default-router.
DHCP Server Option to Ignore all BOOTP Requests	The DHCP Server Option to Ignore all BOOTP Requests feature allows the Cisco IOS DHCP server to selectively ignore and not reply to received Bootstrap Protocol (BOOTP) request packets. The following command was introduced or modified: ip dhcp bootp ignore.



Configuring DHCP Snooping, IP Source Guard, and IPSPG for Static Hosts

This chapter describes how to configure Dynamic Host Configuration Protocol (DHCP) snooping, IP source guard, and IP source guard (IPSPG) for static hosts on Catalyst 4500 series switches. It provides guidelines, procedures, and configuration examples.

This chapter consists of the following major sections:

- [About DHCP Snooping, page 60-1](#)
- [Configuring DHCP Snooping, page 60-6](#)
- [Displaying DHCP Snooping Information, page 60-18](#)
- [Displaying IP Source Binding Information, page 60-23](#)
- [Configuring IP Source Guard, page 60-20](#)
- [Displaying IP Source Binding Information, page 60-23](#)
- [Configuring IP Source Guard for Static Hosts, page 60-24](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About DHCP Snooping

DHCP snooping is a DHCP security feature that provides security by filtering untrusted DHCP messages and by building and maintaining a DHCP snooping binding table. An untrusted message is a message that is received from outside the network or firewall and that can cause traffic attacks within your network.

The DHCP snooping binding table contains the MAC address, IP address, lease time, binding type, VLAN number, and interface information that corresponds to the local untrusted interfaces of a switch; it does not contain information regarding hosts interconnected with a trusted interface. An untrusted interface is an interface that is configured to receive messages from outside the network or firewall. A trusted interface is an interface that is configured to receive only messages from within the network.

DHCP snooping acts such as a firewall between untrusted hosts and DHCP servers. It also gives you a way to differentiate between untrusted interfaces connected to the end-user and trusted interfaces connected to the DHCP server or another switch.

**Note**

In order to enable DHCP snooping on a VLAN, you must enable DHCP snooping on the switch.

You can configure DHCP snooping for switches and VLANs. When you enable DHCP snooping on a switch, the interface acts as a Layer 2 bridge, intercepting and safeguarding DHCP messages going to a Layer 2 VLAN. When you enable DHCP snooping on a VLAN, the switch acts as a Layer 2 bridge within a VLAN domain.

This section includes these topics:

- [Trusted and Untrusted Sources, page 60-2](#)
- [About the DHCP Snooping Database Agent, page 60-2](#)
- [Option 82 Data Insertion, page 60-3](#)

Trusted and Untrusted Sources

The DHCP snooping feature determines whether traffic sources are trusted or untrusted. An untrusted source may initiate traffic attacks or other hostile actions. To prevent such attacks, the DHCP snooping feature filters messages and rate-limits traffic from untrusted sources.

In an enterprise network, devices under your administrative control are trusted sources. These devices include the switches, routers and servers in your network. Any device beyond the firewall or outside your network is an untrusted source. Host ports are generally treated as untrusted sources.

In a service provider environment, any device that is not in the service provider network is an untrusted source (such as a customer switch). Host ports are untrusted sources.

In the Catalyst 4500 series switch, you indicate that a source is trusted by configuring the trust state of its connecting interface.

The default trust state of all interfaces is untrusted. You must configure DHCP server interfaces as trusted. You can also configure other interfaces as trusted if they connect to devices (such as switches or routers) inside your network. You usually do not configure host port interfaces as trusted.

**Note**

For DHCP snooping to function properly, all DHCP servers must be connected to the switch through trusted interfaces, as untrusted DHCP messages will be forwarded only to trusted interfaces.

About the DHCP Snooping Database Agent

To retain the bindings across switch reloads, you must use the DHCP snooping database agent. Without this agent, the bindings established by DHCP snooping are lost upon switch reload. Connectivity is lost as well.

The mechanism for the database agent stores the bindings in a file at a configured location. Upon reload, the switch reads the file to build the database for the bindings. The switch keeps the file current by writing to the file as the database changes.

The format of the file that contains the bindings is as follows:

```
<initial-checksum>
TYPE DHCP-SNOOPING
VERSION 1
BEGIN
<entry-1> <checksum-1>
<entry-2> <checksum-1-2>
...
...
<entry-n> <checksum-1-2-...-n>
END
```

Each entry in the file is tagged with a checksum that is used to validate the entries whenever the file is read. The <initial-checksum> entry on the first line helps distinguish entries associated with the latest write from entries that are associated with a previous write.

it is a sample bindings file:

```
3ebe1518
TYPE DHCP-SNOOPING
VERSION 1
BEGIN
10.1.1.1 512 0001.0001.0005 3EBE2881 Gi1/1 e5e1e733
10.1.1.1 512 0001.0001.0002 3EBE2881 Gi1/1 4b3486ec
10.1.1.1 1536 0001.0001.0004 3EBE2881 Gi1/1 f0e02872
10.1.1.1 1024 0001.0001.0003 3EBE2881 Gi1/1 ac41adf9
10.1.1.1 1 0001.0001.0001 3EBE2881 Gi1/1 34b3273e
END
```

Each entry holds an IP address, VLAN, MAC address, lease time (in hex), and the interface associated with a binding. At the end of each entry is a checksum that accounts for all the bytes from the start of the file through all the bytes associated with the entry. Each entry consists of 72 bytes of data, followed by a space, followed by a checksum.

Upon bootup, when the calculated checksum equals the stored checksum, a switch reads entries from the file and adds the bindings to the DHCP snooping database. When the calculated checksum does not equal the stored checksum, the entry read from the file is ignored and so are all the entries following the failed entry. The switch also ignores all those entries from the file whose lease time has expired. (This situation is possible because the lease time might indicate an expired time.) An entry from the file is also ignored if the interface referred to in the entry no longer exists on the system or if it is a router port or a DHCP snooping-trusted interface.

When a switch learns of new bindings or when it loses some bindings, the switch writes the modified set of entries from the snooping database to the file. The writes are performed with a configurable delay to batch as many changes as possible before the actual write happens. Associated with each transfer is a timeout after which a transfer is aborted if it is not completed. These timers are referred to as the write delay and abort timeout.

Option 82 Data Insertion

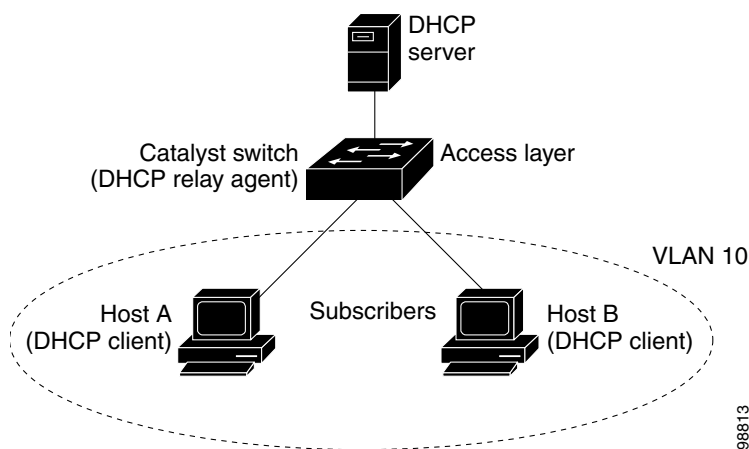
In residential, metropolitan Ethernet-access environments, DHCP can centrally manage the IP address assignments for a large number of subscribers. When the DHCP Option 82 feature is enabled on the switch, a subscriber device is identified by the switch port through which it connects to the network (in addition to its MAC address). Multiple hosts on the subscriber LAN can be connected to the same port on the access switch and are uniquely identified.

**Note**

The DHCP Option 82 feature is supported only when DHCP snooping is globally enabled and on the VLANs to which subscriber devices using this feature are assigned.

Figure 60-1 is an example of a metropolitan Ethernet network in which a centralized DHCP server assigns IP addresses to subscribers connected to the switch at the access layer. Because the DHCP clients and their associated DHCP server do not reside on the same IP network or subnet, a DHCP relay agent (the Catalyst switch) is configured with a helper address to enable broadcast forwarding and to transfer DHCP messages between the clients and the server.

Figure 60-1 DHCP Relay Agent in a Metropolitan Ethernet Network



When you enable the DHCP snooping information Option 82 on the switch, this sequence of events occurs:

- The host (DHCP client) generates a DHCP request and broadcasts it on the network.
- When the switch receives the DHCP request, it adds the Option 82 information in the packet. By default, the remote ID suboption is the switch MAC address, and the circuit ID suboption is the port identifier, **vlan-mod-port**, from which the packet is received. Beginning with Cisco IOS Release 12.2(40)SG, you can configure the remote ID and circuit ID. For information on configuring these suboptions, see the [“Enabling DHCP Snooping and Option 82”](#) section on page 60-10.
- If the IP address of the relay agent is configured, the switch adds this IP address in the DHCP packet.
- The switch forwards the DHCP request that includes the Option 82 field to the DHCP server.
- The DHCP server receives the packet. If the server is Option 82-capable, it can use the remote ID, the circuit ID, or both to assign IP addresses and implement policies, such as restricting the number of IP addresses that can be assigned to a single remote ID or circuit ID. The DHCP server then echoes the Option 82 field in the DHCP reply.
- The DHCP server unicasts the reply to the switch if the request was relayed to the server by the switch. The switch verifies that it originally inserted the Option 82 data by inspecting the remote ID and possibly the circuit ID fields. The switch removes the Option 82 field and forwards the packet to the switch port that connects to the DHCP client that sent the DHCP request.

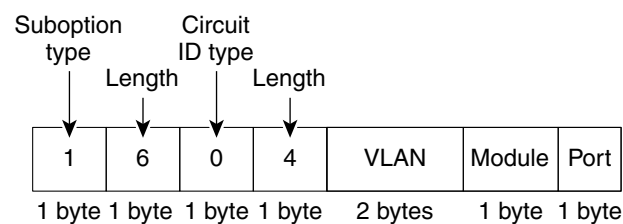
In the default suboption configuration, when the described sequence of events occurs, the values in these fields in [Figure 60-2](#) do not change:

- Circuit ID suboption fields
 - Suboption type
 - Length of the suboption type
 - Circuit ID type
 - Length of the circuit ID type
- Remote ID suboption fields
 - Suboption type
 - Length of the suboption type
 - Remote ID type
 - Length of the remote ID type

[Figure 60-2](#) shows the packet formats for the remote ID suboption and the circuit ID suboption when the default suboption configuration is used. For the circuit ID suboption, the module number corresponds to the switch module number. The switch uses the packet formats when you globally enable DHCP snooping and enter the **ip dhcp snooping information option** global configuration command.

Figure 60-2 Suboption Packet Formats

Circuit ID Suboption Frame Format



Remote ID Suboption Frame Format



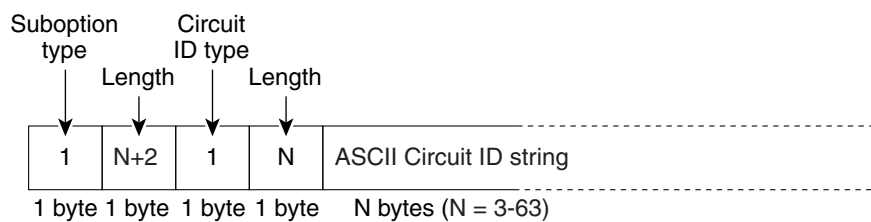
[Figure 60-3](#) shows the packet formats for user-configured remote ID and circuit ID suboptions. The switch uses these packet formats when DHCP snooping is globally enabled and when the **ip dhcp snooping information option format remote-id** global configuration command and the **ip dhcp snooping vlan information option format-type circuit-id string** interface configuration command are entered.

The values for these fields in the packets change from the default values when you configure the remote ID and circuit ID suboptions:

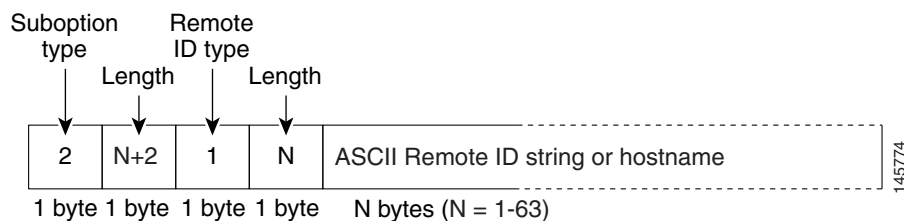
- Circuit ID suboption fields
 - The circuit ID type is 1.
 - The length values are variable, depending on the length of the string that you configure.
- Remote ID suboption fields
 - The remote ID type is 1.
 - The length values are variable, depending on the length of the string that you configure.

Figure 60-3 User-Configured Suboption Packet Formats

Circuit ID Suboption Frame Format (for user-configured string):



Remote ID Suboption Frame Format (for user-configured string):



Configuring DHCP Snooping

When you configure DHCP snooping on your switch, you are enabling the switch to differentiate untrusted interfaces from trusted interfaces. You must enable DHCP snooping globally before you can use DHCP snooping on a VLAN. You can enable DHCP snooping independently from other DHCP features.

These sections describe how to configure DHCP snooping:

- [Default Configuration for DHCP Snooping, page 60-7](#)
- [Enabling DHCP Snooping, page 60-7](#)
- [Enabling DHCP Snooping on the Aggregation Switch, page 60-9](#)
- [Enabling DHCP Snooping and Option 82, page 60-10](#)
- [Enabling DHCP Snooping on Private VLAN, page 60-12](#)
- [Configuring DHCP Snooping on Private VLAN, page 60-12](#)

- [Configuring DHCP Snooping with an Ethernet Channel Group](#), page 60-12
- [Enabling the DHCP Snooping Database Agent](#), page 60-13
- [Limiting the Rate of Incoming DHCP Packets](#), page 60-13
- [Configuration Examples for the Database Agent](#), page 60-15

**Note**

For DHCP server configuration information, refer to “Configuring DHCP” in the *Cisco IOS IP and IP Routing Configuration Guide* at:

http://www.cisco.com/en/US/docs/ios/12_2/ip/configuration/guide/1cfdhcp.html

Default Configuration for DHCP Snooping

DHCP snooping is disabled by default. [Table 60-1](#) shows all the default configuration values for each DHCP snooping option.

Table 60-1 **Default Configuration Values for DHCP Snooping**

Option	Default Value/State
DHCP snooping	Disabled
DHCP snooping information option	Enabled
DHCP snooping information option allow-untrusted	Disabled
DHCP snooping limit rate	Infinite (functions as if rate limiting were disabled)
DHCP snooping trust	Untrusted
DHCP snooping vlan	Disabled

If you want to change the default configuration values, see the “[Enabling DHCP Snooping](#)” section.

Enabling DHCP Snooping

**Note**

When DHCP snooping is enabled globally, DHCP requests are dropped until the ports are configured. Consequently, you should probably configure this feature during a maintenance window and not during production.

To enable DHCP snooping, perform this task:

	Command	Purpose
Step 1	Switch(config)# ip dhcp snooping	Enables DHCP snooping globally. You can use the no keyword to disable DHCP snooping.
Step 2	Switch(config)# ip dhcp snooping vlan <i>number</i> [<i>number</i>] vlan { <i>vlan range</i> }	Enables DHCP snooping on your VLAN or VLAN range.

	Command	Purpose
Step 3	Switch(config)# errdisable recovery {cause dhcp-rate-limit interval interval}	(Optional) Configures the amount of time required for recovery from a specified errdisable cause.
Step 4	Switch(config)# errdisable detect cause dhcp-rate-limit {action shutdown vlan}	(Optional) Enables per-VLAN errdisable detection. Note By default this command is enabled, and when a violation occurs the interface is shutdown.
Step 5	Switch(config-if)# ip dhcp snooping trust	Configures the interface as trusted or untrusted. You can use the no keyword to configure an interface to receive messages from an untrusted client.
Step 6	Switch(config-if)# ip dhcp snooping limit rate rate	Configures the number of DHCP packets per second (pps) that an interface can receive. ¹
Step 7	Switch(config)# end	Exits configuration mode.
Step 8	Switch# show ip dhcp snooping	Verifies the configuration.

1. We recommend not configuring the untrusted interface rate limit to more than 100 packets per second. The recommended rate limit for each untrusted client is 15 packets per second. Normally, the rate limit applies to untrusted interfaces. If you want to set up rate limiting for trusted interfaces, keep in mind that trusted interfaces aggregate all DHCP traffic in the switch, and you will need to adjust the rate limit to a higher value. You should fine tune this threshold depending on the network configuration. The CPU should not receive DHCP packets at a sustained rate of more than 1,000 packets per second.

You can configure DHCP snooping for a single VLAN or a range of VLANs. To configure a single VLAN, enter a single VLAN number. To configure a range of VLANs, enter a beginning and an ending VLAN number or a dash and range of VLANs.

The number of incoming DHCP packets is rate-limited to prevent a denial-of-service attack. When the rate of incoming DHCP packets exceeds the configured limit, the switch places the port in the errdisabled state. To prevent the port from shutting down, you can use the **errdisable detect cause dhcp-rate-limit action shutdown vlan** global configuration command to shut down just the offending VLAN on the port where the violation occurred.

When a secure port is in the errdisabled state, you can bring it out of this state automatically by configuring the **errdisable recovery cause dhcp-rate-limit** global configuration command or you can manually reenabel it by entering the **shutdown** and **no shutdown** interface configuration commands. If a port is in per-VLAN errdisable mode, you can also use **clear errdisable interface name vlan range** command to reenabel the VLAN on the port.

This example shows how to enable DHCP snooping on VLAN 500 through 555:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# ip dhcp snooping
Switch(config)# ip dhcp snooping vlan 500 555
Switch(config)# ip dhcp snooping information option format remote-id string switch123
Switch(config)# interface GigabitEthernet 5/1
Switch(config-if)# ip dhcp snooping trust
Switch(config-if)# ip dhcp snooping limit rate 100
Switch(config-if)# ip dhcp snooping vlan 555 information option format-type circuit-id
string customer-555
Switch(config-if)# interface FastEthernet 2/1
Switch(config-if)# ip dhcp snooping vlan 555 information option format-type circuit-id
string customer-500
Switch(config)# end
Switch# show ip dhcp snooping
Switch DHCP snooping is enabled
DHCP snooping is configured on following VLANs:
500,555
```


DHCP snooping is operational on following VLANs:
500,555

DHCP snooping is configured on the following L3 Interfaces:

Insertion of option 82 is enabled

circuit-id default format: vlan-mod-port

remote-id: switch123 (string)

Option 82 on untrusted port is not allowed Verification of hwaddr field is enabled DHCP snooping trust/rate is configured on the following Interfaces:

Interface	Trusted	Rate limit (pps)
-----	-----	-----
FastEthernet5/1	yes	100
Custom circuit-ids:		
VLAN 555: customer-555		
FastEthernet2/1	no	unlimited
Custom circuit-ids:		
VLAN 500: customer-500		

Switch#

The following configuration describes the DHCP snooping configuration steps if routing is defined on another Catalyst switch (for example, a Catalyst 6500 series switch):

```
// Trust the uplink gigabit Ethernet trunk port
```

```
interface range GigabitEthernet 1/1 - 2
switchport mode trunk
switchport trunk encapsulation dot1q
ip dhcp snooping trust
```

```
!
```

```
interface VLAN 14
ip address 10.33.234.1 255.255.254.0
ip helper-address 10.5.1.2
```



Note

If you are enabling trunking on uplink gigabit interfaces, and the above routing configuration is defined on a Catalyst 6500 series switch, you must configure the “trust” relationship with downstream DHCP snooping (on a Catalyst 4500 series switch) which adds Option 82. On a Catalyst 6500 series switch, this task is accomplished with the **ip dhcp relay information trusted** VLAN configuration command.

Enabling DHCP Snooping on the Aggregation Switch

To enable DHCP snooping on an aggregation switch, configure the interface connecting to a downstream switch as a snooping untrusted port. If the downstream switch (or a device such as a DSLAM in the path between the aggregation switch and the DHCP clients) adds DHCP information Option 82 to the DHCP packets, the DHCP packets would be dropped on arriving on a snooping untrusted port. If you configure the **ip dhcp snooping information option allow-untrusted** global configuration command on the aggregation switch, the aggregation switch can accept DHCP requests with Option 82 information from any snooping untrusted port.

Enabling DHCP Snooping and Option 82

To enable DHCP snooping and Option 82 on the switch, perform the following steps:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# ip dhcp snooping	Enables DHCP snooping globally.
Step 3	Switch(config)# ip dhcp snooping vlan <i>vlan-range</i>	Enables DHCP snooping on a VLAN or range of VLANs. The range is 1 to 4094. You can enter a single VLAN ID identified by VLAN ID number, a series of VLAN IDs separated by commas, a range of VLAN IDs separated by hyphens, or a range of VLAN IDs separated by entering the starting and ending VLAN IDs separated by a space.
Step 4	Switch(config)# ip dhcp snooping information option	Enables the switch to insert and remove DHCP relay information (Option 82 field) in forwarded DHCP request messages to the DHCP server. It is the default setting.
Step 5	Switch(config)# ip dhcp snooping information option format remote-id [string <i>ASCII-string</i> <i>hostname</i>]	(Optional) Configures the remote ID suboption. You can configure the remote ID to be: <ul style="list-style-type: none"> String of up to 63 ASCII characters (no spaces) Configured hostname for the switch If the hostname is longer than 63 characters, it is truncated to 63 characters in the remote ID configuration. The default remote ID is the switch MAC address.
Step 6	Switch(config)# ip dhcp snooping information option allow-untrusted	(Optional) If the switch is an aggregation switch connected to an edge switch, enables the switch to accept incoming DHCP snooping packets with Option 82 information from the edge switch. The default setting is disabled. Note Enter this command only on aggregation switches that are connected to trusted devices.
Step 7	Switch(config)# interface <i>interface-id</i>	Specifies the interface to be configured, and enter interface configuration mode.
Step 8	Switch(config-if)# ip dhcp snooping vlan <i>vlan</i> information option format-type circuit-id [override] string <i>ASCII-string</i>	(Optional) Configures the circuit ID suboption for the specified interface. Specify the VLAN and port identifier, using a VLAN ID in the range of 1 to 4094. The default circuit ID is the port identifier, in the format vlan-mod-port . You can configure the circuit ID to be a string of 3 to 63 ASCII characters (no spaces). Optional) Use the override keyword when you do not want the circuit-ID suboption inserted in TLV format to define subscriber information.
Step 9	Switch(config-if)# ip dhcp snooping trust	(Optional) Configures the interface as trusted or untrusted. You can use the no keyword to configure an interface to receive messages from an untrusted client. The default setting is untrusted.

	Command	Purpose
Step 10	Switch(config-if)# ip dhcp snooping limit rate rate	(Optional) Configures the number of DHCP packets per second that an interface can receive. The range is 1 to 2048. By default, no rate limit is configured. Note We recommend an untrusted rate limit of not more than 100 packets per second. If you configure rate limiting for trusted interfaces, you might need to increase the rate limit if the port is a trunk port assigned to more than one VLAN on which DHCP snooping is enabled.
Step 11	Switch(config-if)# exit	Returns to global configuration mode.
Step 12	Switch(config)# ip dhcp snooping verify mac-address	(Optional) Configures the switch to verify that the source MAC address in a DHCP packet that is received on untrusted ports matches the client hardware address in the packet. The default is to verify that the source MAC address matches the client hardware address in the packet.
Step 13	Switch(config)# end	Returns to privileged EXEC mode.
Step 14	Switch# show running-config	Verifies your entries.
Step 15	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To disable DHCP snooping, use the **no ip dhcp snooping** global configuration command. To disable DHCP snooping on a VLAN or range of VLANs, use the **no ip dhcp snooping vlan vlan-range** global configuration command. To disable the insertion and removal of the Option 82 field, use the **no ip dhcp snooping information option** global configuration command. To configure an aggregation switch to drop incoming DHCP snooping packets with Option 82 information from an edge switch, use the **no ip dhcp snooping information option allow-untrusted** global configuration command.

This example shows how to enable DHCP snooping globally and on VLAN 10 and to configure a rate limit of 100 packets per second on a port:

```
Switch(config)# ip dhcp snooping
Switch(config)# ip dhcp snooping vlan 10
Switch(config)# ip dhcp snooping information option
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# ip dhcp snooping limit rate 100
```

The following example shows how to enable DHCP snooping on VLAN 500 through 555 and option 82 circuit-id:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# ip dhcp snooping
Switch(config)# ip dhcp snooping vlan 500 555
Switch(config)# ip dhcp snooping information option format remote-id string switch123
Switch(config)# interface GigabitEthernet 5/1
Switch(config-if)# ip dhcp snooping trust
Switch(config-if)# ip dhcp snooping limit rate 100
Switch(config-if)# ip dhcp snooping vlan 555 information option format-type circuit-id
string customer-555
Switch(config-if)# interface FastEthernet 2/1
Switch(config-if)# ip dhcp snooping vlan 555 information option format-type circuit-id
string customer-500
Switch(config)# end
```

This example shows how to configure the Option 82 circuit-ID override suboption:

```
Switch(config-if)# ip dhcp snooping vlan 250 information option format-type circuit-id
```

```
override string testcustomer
```

Enabling DHCP Snooping on Private VLAN

DHCP snooping can be enabled on private VLANs, which provide isolation between Layer 2 ports within the same VLAN. If DHCP snooping is enabled (or disabled), the configuration is propagated to both the primary VLAN and its associated secondary VLANs. You cannot enable (or disable) DHCP snooping on a primary VLAN without reflecting this configuration change on the secondary VLANs.

Configuring DHCP snooping on a secondary VLAN is still allowed, but it does not take effect if the associated primary VLAN is already configured. If the associated primary VLAN is configured, the effective DHCP snooping mode on the secondary VLAN is derived from the corresponding primary VLAN. Manually configuring DHCP snooping on a secondary VLAN causes the switch to issue this warning message:

```
DHCP Snooping configuration may not take effect on secondary vlan XXX
```

The **show ip dhcp snooping** command displays all VLANs (both primary and secondary) that have DHCP snooping enabled.

Configuring DHCP Snooping on Private VLAN

DHCP snooping, IPSG, and DAI are Layer 2-based security features that can be enabled and disabled on an individual VLAN, including auxiliary or voice VLAN. You need to enable DHCP snooping on a voice VLAN for a Cisco IP phone to function properly.

Configuring DHCP Snooping with an Ethernet Channel Group

When you configure DHCP snooping, you need to configure trunk interfaces that transmit DHCP packets as trusted interfaces by adding **ip dhcp snooping trust** to the physical interface configuration. However, if DHCP packets will be transmitted over an Ethernet channel group, you must configure **ip dhcp snooping trust** on the logical port channel interface, for example:

```
Switch# show run int port-channel150
Building configuration...

Current configuration : 150 bytes
!
interface Port-channel150
  switchport
  switchport trunk native vlan 4092
  switchport mode trunk
  switchport nonegotiate
  ip dhcp snooping trust
end

Switch#
```

Enabling the DHCP Snooping Database Agent

To configure the database agent, perform one or more of the following tasks:

Command	Purpose
Switch(config)# ip dhcp snooping database {url write-delay seconds timeout seconds}	(Required) Configures a URL for the database agent (or file) and the related timeout values.
Switch(config)# no ip dhcp snooping database [write-delay timeout]	
Switch# show ip dhcp snooping database [detail]	(Optional) Displays the current operating state of the database agent and statistics associated with the transfers.
Switch# clear ip dhcp snooping database statistics	(Optional) Clears the statistics associated with the database agent.
Switch# renew ip dhcp snooping database [validation none] [url]	(Optional) Requests the read entries from a file at the given URL.
Switch# ip dhcp snooping binding mac-addr vlan vlan ipaddr interface ifname expiry lease-in-seconds	(Optional) Adds or deletes bindings to the snooping database.
Switch# no ip dhcp snooping binding mac-addr vlan vlan ipaddr interface ifname	



Note

Because both NVRAM and bootflash have limited storage capacity, you should use TFTP or network-based files. If you use flash to store the database file, new updates (by the agent) result in the creation of new files (flash fills quickly). Moreover, because of the nature of the file system used on flash, a large number of files can cause slow access. When a file is stored in a remote location accessible through TFTP, an RPR or SSO standby supervisor engine can take over the binding list when a switchover occurs.



Note

Network-based URLs (such as TFTP and FTP) require that you create an empty file at the configured URL before the switch can write the set of bindings for the first time.

Limiting the Rate of Incoming DHCP Packets

The switch CPU performs DHCP validation checks; therefore, the number of incoming DHCP packets is rate-limited to prevent a denial-of-service attack.

When the rate of incoming DHCP packets exceeds the configured limit, the switch places the port in the errdisabled state. The port remains in that state until you intervene or you enable errdisable recovery so that ports automatically emerge from this state after a specified timeout period.



Note

Unless you explicitly configure a rate limit on an interface, changing the trust state of the interface also changes its rate limit to the default value for that trust state. After you configure the rate limit, the interface retains the rate limit even when its trust state is changed. If you enter the **no ip dhcp snooping limit rate** interface configuration command, the interface reverts to its default rate limit.

To prevent the port from shutting down, you can use the **errdisable detect cause dhcp-rate-limit action shutdown vlan** global configuration command to shut down just the offending VLAN on the port where the violation occurred.

To limit the rate of incoming DHCP packets, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# errdisable detect cause dhcp-rate-limit [action shutdown vlan]	Enables per-VLAN errdisable detection.
Step 3	Switch(config)# interface interface-id	Specifies the interface to be rate-limited, and enter interface configuration mode.
Step 4	Switch(config-if)# [no] ip dhcp snooping limit rate	Limits the rate of incoming DHCP requests and responses on the interface. The default rate is disabled.
Step 5	Switch(config-if)# exit	Returns to global configuration mode.
Step 6	Switch(config)# errdisable recovery {cause dhcp-rate-limit interval interval}	(Optional) Enables error recovery from the DHCP errdisable state. By default, recovery is disabled, and the recovery interval is 300 seconds. For interval interval , specify the time in seconds to recover from the errdisable state. The range is 30 to 86400.
Step 7	Switch(config)# exit	Returns to privileged EXEC mode.
Step 8	Switch# show interfaces status	Verifies your settings.
Step 9	Switch# show errdisable recovery	Verifies your settings.
Step 10	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To return to the default rate-limit configuration, use the **no ip dhcp-rate-limit** interface configuration command. To disable error recovery for DHCP inspection, use the **no errdisable recovery cause dhcp-rate-limit** global configuration command.

This example shows how to set an upper limit for the number of incoming packets (100 pps) and to specify a burst interval (1 second):

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface g3/31
Switch(config-if)# ip dhcp-rate-limit rate 100 burst interval 1
Switch(config-if)# exit
Switch(config)# errdisable recovery cause dhcp-rate-limit
Switch(config)# exit
Switch# show interfaces status
```

Port	Name	Status	Vlan	Duplex	Speed	Type
Tel1/1		connected	1	full	10G	10GBase-LR
Tel1/2		connected	vl-err-dis	full	10G	10GBase-LR

```

SwitchB# show errdisable recovery
ErrDisable Reason    Timer Status
-----
udld                  Disabled
bpduguard             Disabled
security-violatio    Disabled
channel-misconfig    Disabled
vmmps                 Disabled
pagp-flap             Disabled
dtp-flap              Disabled
link-flap             Disabled
l2ptguard             Disabled
psecure-violation    Disabled
gbic-invalid          Disabled
dhcp-rate-limit       Disabled
unicast-flood         Disabled
storm-control         Disabled
arp-inspection        Enabled

Timer interval: 300 seconds

Interfaces that will be enabled at the next timeout:

SwitchB#
1w2d: %SW_DAI-4-PACKET_RATE_EXCEEDED: 101 packets received in 739 milliseconds on Gi3/31.
1w2d: %PM-4-ERR_DISABLE: arp-inspection error detected on Gi3/31, putting Gi3/31 in
err-disable state
SwitchB# show clock
*02:21:43.556 UTC Fri Feb 4 2005
SwitchB#
SwitchB# show interface g3/31 status

Port      Name                Status      Vlan      Duplex  Speed Type
Gi3/31                    err-disabled 100        auto     auto 10/100/1000-TX
SwitchB#
SwitchB#
1w2d: %PM-4-ERR_RECOVER: Attempting to recover from arp-inspection err-disable state on
Gi3/31
SwitchB# show interface g3/31 status

Port      Name                Status      Vlan      Duplex  Speed Type
Gi3/31                    connected   100        a-full   a-100 10/100/1000-TX
SwitchB# show clock
*02:27:40.336 UTC Fri Feb 4 2005
SwitchB#

```

Configuration Examples for the Database Agent

The following examples show how to configuration commands in the previous procedure:

Example 1: Enabling the Database Agent

The following example shows how to configure the DHCP snooping database agent to store the bindings at a given location and to view the configuration and operating state:

```

Switch# configure terminal
Switch(config)# ip dhcp snooping database tftp://10.1.1.1/directory/file
Switch(config)# end

```

```

Switch# show ip dhcp snooping database detail
Agent URL : tftp://10.1.1.1/directory/file
Write delay Timer : 300 seconds
Abort Timer : 300 seconds

Agent Running : No
Delay Timer Expiry : 7 (00:00:07)
Abort Timer Expiry : Not Running

Last Succeeded Time : None
Last Failed Time : 17:14:25 UTC Sat Jul 7 2001
Last Failed Reason : Unable to access URL.

Total Attempts      :      21   Startup Failures :      0
Successful Transfers :      0   Failed Transfers :     21
Successful Reads    :      0   Failed Reads   :      0
Successful Writes   :      0   Failed Writes  :     21
Media Failures      :      0

First successful access: Read

Last ignored bindings counters :
Binding Collisions   :      0   Expired leases   :      0
Invalid interfaces   :      0   Unsupported vlans :      0
Parse failures       :      0
Last Ignored Time : None

Total ignored bindings counters:
Binding Collisions   :      0   Expired leases   :      0
Invalid interfaces   :      0   Unsupported vlans :      0
Parse failures       :      0

Switch#

```

The first three lines of output show the configured URL and related timer configuration values. The next three lines show the operating state and the amount of time left for expiry of write delay and abort timers.

Among the statistics shown in the output, startup failures indicate the number of attempts the read or create of the file has failed upon bootup.



Note

Because the location is based off in the network, you must create a temporary file on the TFTP server. You can create a temporary file on a typical UNIX workstation by creating a 0 byte file “file” in the directory “directory” that can be referenced by the TFTP server daemon. With some server implementations on UNIX workstations, the file should be provided with full (777) permissions for write access to the file.

DHCP snooping bindings are keyed on the MAC address and VLAN combination. If an entry in the remote file has an entry for a given MAC address and VLAN set, for which the switch already has a binding, the entry from the remote file is ignored when the file is read. This condition is referred to as the binding collision.

An entry in a file may no longer be valid because the lease indicated by the entry may have expired by the time it is read. The expired leases counter indicates the number of bindings ignored because of this condition. The Invalid interfaces counter refers to the number of bindings that have been ignored when the interface referred by the entry either does not exist on the system or is a router or DHCP snooping trusted interface if it exists, when the read happened. Unsupported VLANs refers to the number of entries that have been ignored because the indicated VLAN is not supported on the system. The Parse failures counter provides the number of entries that have been ignored when the switch is unable to interpret the meaning of the entries from the file.

The switch maintains two sets of counters for these ignored bindings. One provides the counters for a read that has at least one binding ignored by at least one of these conditions. These counters are shown as the “Last ignored bindings counters.” The total ignored bindings counters provides a sum of the number of bindings that have been ignored because of all the reads since the switch bootup. These two set of counters are cleared by the **clear** command. The total counter set may indicate the number of bindings that have been ignored since the last clear.

Example 2: Reading Binding Entries from a TFTP File

To manually read the entries from a TFTP file, perform this task:

	Command	Purpose
Step 1	Switch# show ip dhcp snooping database	Displays the DHCP snooping database agent statistics.
Step 2	Switch# renew ip dhcp snoop data url	Directs the switch to read the file from given URL.
Step 3	Switch# show ip dhcp snoop data	Displays the read status.
Step 4	Switch# show ip dhcp snoop bind	Verifies whether the bindings were read successfully.

it is an example of how to manually read entries from the tftp://10.1.1.1/directory/file:

```
Switch# show ip dhcp snooping database
Agent URL :
Write delay Timer : 300 seconds
Abort Timer : 300 seconds

Agent Running : No
Delay Timer Expiry : Not Running
Abort Timer Expiry : Not Running

Last Succeeded Time : None
Last Failed Time : None
Last Failed Reason : No failure recorded.

Total Attempts      :      0   Startup Failures :      0
Successful Transfers :      0   Failed Transfers :      0
Successful Reads     :      0   Failed Reads   :      0
Successful Writes    :      0   Failed Writes  :      0
Media Failures       :      0

Switch#
Switch# renew ip dhcp snoop data tftp://10.1.1.1/directory/file
Loading directory/file from 10.1.1.1 (via GigabitEthernet1/1): !
[OK - 457 bytes]
Database downloaded successfully.

Switch#
00:01:29: %DHCP_SNOOPING-6-AGENT_OPERATION_SUCCEEDED: DHCP snooping database Read
succeeded.
Switch#
Switch# show ip dhcp snoop data
Agent URL :
Write delay Timer : 300 seconds
Abort Timer : 300 seconds

Agent Running : No
Delay Timer Expiry : Not Running
Abort Timer Expiry : Not Running
```

■ Displaying DHCP Snooping Information

```

Last Succeeded Time : 15:24:34 UTC Sun Jul 8 2001
Last Failed Time : None
Last Failed Reason : No failure recorded.

```

```

Total Attempts      :          1   Startup Failures :          0
Successful Transfers :          1   Failed Transfers :          0
Successful Reads     :          1   Failed Reads    :          0
Successful Writes    :          0   Failed Writes   :          0
Media Failures       :          0

```

```
Switch#
```

```
Switch# show ip dhcp snoop bind
```

MacAddress	IpAddress	Lease(sec)	Type	VLAN	Interface
00:01:00:01:00:05	10.1.1.1	49810	dhcp-snooping	512	GigabitEthernet1/1
00:01:00:01:00:02	10.1.1.1	49810	dhcp-snooping	512	GigabitEthernet1/1
00:01:00:01:00:04	10.1.1.1	49810	dhcp-snooping	1536	GigabitEthernet1/1
00:01:00:01:00:03	10.1.1.1	49810	dhcp-snooping	1024	GigabitEthernet1/1
00:01:00:01:00:01	10.1.1.1	49810	dhcp-snooping	1	GigabitEthernet1/1

```
Switch#
```

```
Switch# clear ip dhcp snoop bind
```

```
Switch# show ip dhcp snoop bind
```

MacAddress	IpAddress	Lease(sec)	Type	VLAN	Interface
------------	-----------	------------	------	------	-----------

```
Switch#
```

Example 3: Adding Information to the DHCP Snooping Database

To manually add a binding to the DHCP snooping database, perform this task:

	Command	Purpose
Step 1	Switch# show ip dhcp snooping binding	Views the DHCP snooping database.
Step 2	Switch# ip dhcp snooping binding <i>binding-id</i> vlan <i>vlan-id</i> interface <i>interface</i> expiry <i>lease-time</i>	Adds the binding using the ip dhcp snooping EXEC command.
Step 3	Switch# show ip dhcp snooping binding	Checks the DHCP snooping database.

This example shows how to manually add a binding to the DHCP snooping database:

```
Switch# show ip dhcp snooping binding
```

MacAddress	IpAddress	Lease(sec)	Type	VLAN	Interface
------------	-----------	------------	------	------	-----------

```
Switch#
```

```
Switch# ip dhcp snooping binding 1.1.1 vlan 1 10.1.1.1 interface gi1/1 expiry 1000
```

```
Switch# show ip dhcp snooping binding
```

MacAddress	IpAddress	Lease(sec)	Type	VLAN	Interface
------------	-----------	------------	------	------	-----------

00:01:00:01:00:01	10.1.1.1	992	dhcp-snooping	1	GigabitEthernet1/1
-------------------	----------	-----	---------------	---	--------------------

```
Switch#
```

Displaying DHCP Snooping Information

You can display a DHCP snooping binding table and configuration information for all interfaces on a switch.

Displaying a Binding Table

The DHCP snooping binding table for each switch contains binding entries that correspond to untrusted ports. The table does not contain information about hosts interconnected with a trusted port because each interconnected switch has its own DHCP snooping binding table.

This example shows how to display the DHCP snooping binding information for a switch:

```
Switch# show ip dhcp snooping binding
-----
MacAddress      IpAddress      Lease(sec)    Type           VLAN    Interface
-----
00:02:B3:3F:3B:99  55.5.5.2      6943          dhcp-snooping  10      FastEthernet6/10
Switch#
```

Table 60-2 describes the fields in the `show ip dhcp snooping binding` command output.

Table 60-2 *show ip dhcp snooping binding Command Output*

Field	Description
MAC Address	Client hardware MAC address
IP Address	Client IP address assigned from the DHCP server
Lease (seconds)	IP address lease time
Type	Binding type; dynamic binding learned by DHCP snooping or statically-configured binding.
VLAN	VLAN number of the client interface
Interface	Interface that connects to the DHCP client host

Displaying the DHCP Snooping Configuration

This example shows how to display the DHCP snooping configuration for a switch:

```
Switch# show ip dhcp snooping
Switch DHCP snooping is enabled.
DHCP Snooping is configured on the following VLANs:
  10 30-40 100 200-220
Insertion of option 82 is enabled
Option82 on untrusted port is not allowed
Verification of hwaddr field is enabled
Interface      Trusted      Rate limit (pps)
-----
FastEthernet2/1  yes         10
FastEthernet3/1  yes         none
GigabitEthernet1/1 no          20
Switch#
```

About IP Source Guard

The IP source guard feature is enabled on a DHCP snooping untrusted Layer 2 port. Initially, all IP traffic on the port is blocked except for DHCP packets that are captured by the DHCP snooping process. When a client receives a valid IP address from the DHCP server, or when you configure a static IP source binding, a per-port and VLAN access control list (VACL) is installed on the port. This process restricts

the client IP traffic to those source IP addresses configured in the binding; any IP traffic with a source IP address other than that in the IP source binding is filtered out. This filtering limits the ability of a host to attack the network by claiming a neighbor host's IP address.

**Note**

If IP source guard is enabled on a trunk port with a large number of VLANs that have DHCP snooping enabled, you might exhaust ACL hardware resources, and some packets might be switched in software instead.

**Note**

When IP source guard is enabled, you might want to designate an alternative scheme for ACL hardware programming. For more information, see the “TCAM Programming and ACLs” section in [Chapter 62, “Configuring Network Security with ACLs”](#).

IP source guard supports the Layer 2 port only, including both access and trunk. For each untrusted Layer 2 port, there are two levels of IP traffic security filtering:

- Source IP address filter

IP traffic is filtered based on its source IP address. Only IP traffic with a source IP address that matches the IP source binding entry is permitted.

An IP source address filter is changed when a new IP source entry binding is created or deleted on the port. The port VACL is recalculated and reapplied in the hardware to reflect the IP source binding change. By default, if the IP filter is enabled without any IP source binding on the port, a default PVACL that denies all IP traffic is installed on the port. Similarly, when the IP filter is disabled, any IP source filter PVACL is removed from the interface.

- Source IP and MAC address filter

IP traffic is filtered based on its source IP address as well as its MAC address; only IP traffic with source IP and MAC addresses matching the IP source binding entry are permitted.

**Note**

When IP source guard is enabled in IP and MAC filtering mode, the DHCP snooping Option 82 must be enabled to ensure that the DHCP protocol works properly. Without Option 82 data, the switch cannot locate the client host port to forward the DHCP server reply. Instead, the DHCP server reply is dropped, and the client cannot obtain an IP address.

Configuring IP Source Guard

To enable IP source guard, perform this task:

	Command	Purpose
Step 1	Switch(config)# ip dhcp snooping	Enables DHCP snooping globally. You can use the no keyword to disable DHCP snooping.
Step 2	Switch(config)# ip dhcp snooping vlan <i>number</i> [<i>number</i>]	Enables DHCP snooping on your VLANs.
Step 3	Switch(config-if)# no ip dhcp snooping trust	Configures the interface as trusted or untrusted. You can use the no keyword of to configure an interface to receive only messages from within the network.

	Command	Purpose
Step 4	Switch(config-if)# ip verify source vlan dhcp-snooping port-security	Enables IP source guard, source IP, and source MAC address filtering on the port.
Step 5	Switch(config-if)# switchport port-security limit rate invalid-source-mac N	Enables security rate limiting for learned source MAC addresses on the port. Note This limit only applies to the port where IP source guard is enabled as filtering both IP and MAC addresses.
Step 6	Switch(config)# ip source binding mac-address Vlan vlan-id ip-address interface interface-name	Configures a static IP binding on the port.
Step 7	Switch(config)# end	Exits configuration mode.
Step 8	Switch# show ip verify source interface interface-name	Verifies the configuration.

If you want to stop IP source guard with static hosts on an interface, use the following commands in interface configuration submode:

```
Switch(config-if)# no ip verify source
Switch(config-if)# no ip device tracking max
```

If the **no ip device tracking** command is used in interface configuration submode, it actually runs in global configuration mode and causes IP device tracking to be disabled globally. Disabling IP device tracking globally causes IP source guard with static hosts to deny all IP traffic on interfaces using the **ip verify source tracking [port-security]** command.



Note

The static IP source binding can only be configured on switch port. If you enter the **ip source binding vlan interface** command on a Layer 3 port, you receive this error message:

```
Static IP source binding can only be configured on switch port.
```

This example shows how to enable per-Layer 2 port IP source guard on VLAN 10 through 20:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# ip dhcp snooping
Switch(config)# ip dhcp snooping vlan 10 20
Switch(config)# interface fa6/1
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk native vlan 10
Switch(config-if)# switchport trunk allowed vlan 11-20
Switch(config-if)# no ip dhcp snooping trust
Switch(config-if)# ip verify source vlan dhcp-snooping
Switch(config)# end
Switch# show ip verify source interface f6/1
```

Interface	Filter-type	Filter-mode	IP-address	Mac-address	Vlan
Fa6/1	ip-mac	active	10.0.0.1		10
Fa6/1	ip-mac	active	deny-all		11-20

```
Switch#
```

The output shows that there is one valid DHCP binding to VLAN 10.

Configuring IP Source Guard on Private VLANs

For IP source guard to be effective on PVLAN ports, you must enable DHCP snooping on primary VLANs. IP source guard on a primary VLAN is automatically propagated to a secondary VLAN. You can configure static IP source binding on a secondary VLAN, but it does not work. When manually configuring a static IP source binding on a secondary VLAN, you receive the following message:

IP source filter may not take effect on a secondary VLAN where IP source binding is configured. If the private VLAN feature is enabled, IP source filter on the primary VLAN will automatically propagate to all secondary VLAN.



Note

IP Source Guard is supported on private VLAN host ports only.

Displaying IP Source Guard Information

You can display IP source guard PVACL information for all interfaces on a switch using the **show ip verify source** command, as the following examples show:

- This example shows displayed PVACLs if DHCP snooping is enabled on VLAN 10 through 20, if interface fa6/1 is configured for IP filtering, and if there is an existing IP address binding 10.0.0.1 on VLAN 10:

Interface	Filter-type	Filter-mode	IP-address	Mac-address	Vlan
fa6/1	ip	active	10.0.0.1		10
fa6/1	ip	active	deny-all		11-20



Note

The second entry shows that a default PVACL (deny all IP traffic) is installed on the port for those snooping-enabled VLANs that do not have a valid IP source binding.

- This example shows displayed PVACL for a trusted port:

Interface	Filter-type	Filter-mode	IP-address	Mac-address	Vlan
fa6/2	ip	inactive-trust-port			

- This example shows displayed PVACL for a port in a VLAN not configured for DHCP snooping:

Interface	Filter-type	Filter-mode	IP-address	Mac-address	Vlan
fa6/3	ip	inactive-no-snooping-vlan			

- This example shows displayed PVACLs for a port with multiple bindings configured for an IP-to-MAC filtering:

Interface	Filter-type	Filter-mode	IP-address	Mac-address	Vlan
fa6/4	ip-mac	active	10.0.0.2	aaaa.bbbb.cccc	10
fa6/4	ip-mac	active	11.0.0.1	aaaa.bbbb.cccd	11
fa6/4	ip-mac	active	deny-all	deny-all	12-20

- This example shows displayed PVACLs for a port configured for IP-to-MAC filtering but not for port security:

Interface	Filter-type	Filter-mode	IP-address	Mac-address	Vlan
fa6/5	ip-mac	active	10.0.0.3	permit-all	10

```
fa6/5      ip-mac      active      deny-all      permit-all      11-20
```



Note The MAC filter shows permit-all because port security is not enabled, so the MAC filter cannot apply to the port or VLAN and is effectively disabled. Always enable port security first.

- This example shows displayed error message when entering the **show ip verify source** command on a port that does not have an IP source filter mode configured:

```
IP Source Guard is not configured on the interface fa6/6.
```

You can also use the **show ip verify source** command to display all interfaces on the switch that have IP source guard enabled, as follows:

```
Switch# show ip verify source
Interface  Filter-type  Filter-mode  IP-address      Mac-address      Vlan
-----
fa6/1      ip           active       10.0.0.1        10
fa6/1      ip           active       deny-all       11-20
fa6/2      ip           inactive-trust-port
fa6/3      ip           inactive-no-snooping-vlan
fa6/4      ip-mac       active       10.0.0.2        aaaa.bbbb.cccc  10
fa6/4      ip-mac       active       11.0.0.1        aaaa.bbbb.cccd  11
fa6/4      ip-mac       active       deny-all       deny-all        12-20
fa6/5      ip-mac       active       10.0.0.3        permit-all      10
fa6/5      ip-mac       active       deny-all       permit-all      11-20
```

Displaying IP Source Binding Information

You can display all IP source bindings configured on all interfaces on a switch using the **show ip source binding** command.

```
Switch# show ip source binding
MacAddress      IpAddress      Lease(sec)  Type           VLAN  Interface
-----
00:02:B3:3F:3B:99  55.5.5.2      6522        dhcp-snooping  10    FastEthernet6/10
00:00:00:0A:00:0B  11.0.0.1      infinite    static         10    FastEthernet6/10
Switch#
```

Table 60-3 describes the fields in the **show ip source binding** command output.

Table 60-3 *show ip source binding Command Output*

Field	Description
MAC Address	Client hardware MAC address
IP Address	Client IP address assigned from the DHCP server
Lease (seconds)	IP address lease time
Type	Binding type; static bindings configured from CLI to dynamic binding learned from DHCP snooping
VLAN	VLAN number of the client interface
Interface	Interface that connects to the DHCP client host

Configuring IP Source Guard for Static Hosts

**Note**

IPSG for static hosts should not be used on uplink ports.

IP source guard (IPSG) for static hosts extends the IPSG capability to non-DHCP and static environments.

This section includes these topics:

- [About IP Source Guard for Static Hosts, page 60-24](#)
- [Configuring IPSG for Static Hosts on a Layer 2 Access Port, page 60-24](#)
- [Configuring IPSG for Static Hosts on a PVLAN Host Port, page 60-28](#)

About IP Source Guard for Static Hosts

The prior feature, IPSG, uses the entries created by the DHCP snooping feature to validate the hosts connected to a switch. Any traffic received from a host without a valid DHCP binding entry is dropped. A DHCP environment is a prerequisite for IPSG to work. The IPSG for static hosts feature removes IPSG's dependency on DHCP. The switch creates static entries based on ARP requests or other IP packets and uses them to maintain the list of valid hosts for a given port. In addition, you can specify the number of hosts that would be allowed to send traffic to a given port. It is equivalent to port security at Layer 3.

**Note**

Some IP hosts with multiple network interfaces may inject some invalid packets into a network interface. Those invalid packets contain the IP-to-MAC address for another network interface of that host as the source address. It may cause IPSG for static hosts in the switch, which connects to the host, to learn the invalid IP-to-MAC address bindings and reject the valid bindings. You should consult the vendor of the corresponding operating system and the network device of that host to prevent it from injecting invalid packets.

IPSG for static hosts initially learns IP-to-MAC bindings dynamically through an ACL-based snooping method. IP-to-MAC bindings are learned from static hosts by using ARP and IP packets and are stored using the device tracking database. Once the number of IP addresses that have been dynamically learned or statically configured on a given port reaches a maximum limit, any packet with a new IP address is dropped in hardware. To handle hosts that have moved or gone away for any reason, the IPSG for static hosts feature uses the IP device tracking functionality to age out dynamically learned IP address bindings. This feature can be used in conjunction with DHCP snooping. Multiple bindings will be established on a port that is connected to both DHCP and static hosts (that is, bindings will be stored in both the device tracking database as well as the DHCP snooping binding database).

Configuring IPSG for Static Hosts on a Layer 2 Access Port

You can configure IPSG for static hosts on a Layer 2 access port.

To enable IPSG for static hosts with IP filters on a Layer 2 access port, perform this task:

	Command	Purpose
Step 1	Switch(config)# ip device tracking	Turns on the IP host table.
Step 2	Switch(config)# interface fastEthernet a/b	Enters IP configuration mode.
Step 3	Switch(config-if)# switchport mode access	Configures a port as access.
Step 4	Switch(config-if)# switchport access vlan n	Configures the VLAN for this port.
Step 5	Switch(config-if)# ip device tracking maximum n	Establishes a maximum limit for the bindings on this port. Upper bound for the maximum is 10. Note Starting from Cisco IOS XE Release 3.10.1E, the following IPDT commands are deprecated; there are no replacement commands: [no] ip device tracking probe count [no] ip device tracking probe delay . For more related information, see the <i>Configuring SISF-Based Device Tracking</i> chapter in this guide.
Step 6	Switch(config-if)# switchport port-security	(Optional) Activates port security for this port.
Step 7	Switch(config-if)# switchport port-security maximum n	(Optional) Establishes a maximum number of MAC addresses for this port.
Step 8	Switch(config-if)# ip verify source tracking [port-security]	Activates IPSG for static hosts on this port.
Step 9	Switch(config-if)# end	Exits configuration interface mode.
Step 10	Switch# show ip verify source interface-name	Verifies the configuration.
Step 11	Switch# show ip device track all [active inactive] count	Verifies the configuration by displaying the IP-to-MAC binding for a given host on the switch interface. <ul style="list-style-type: none"> • all active—Displays only the active IP-to-MAC binding entries. • all inactive—Displays only the inactive IP-to-MAC binding entries. • all—Displays the active and inactive IP-to-MAC binding entries.

To stop IPSG with static hosts on an interface, use the following commands in interface configuration submode:

```
Switch(config-if)# no ip verify source
Switch(config-if)# no ip device tracking max"
```

To enable IPSG with static hosts on a port, enter the following commands:

```
Switch(config)# ip device tracking ****enable IP device tracking globally
Switch(config)# ip device tracking max <n> ****set an IP device tracking maximum on int
Switch(config-if)# ip verify source tracking [port-security] ****activate IPSG on the port
```



Caution

If you only configure the **ip verify source tracking [port-security]** interface configuration command on a port without enabling IP device tracking globally or setting an IP device tracking maximum on that interface, IPSG with static hosts will reject all the IP traffic from that interface.

This issue also applies to IPSG with static hosts on a PVLAN host port.

This example shows how to enable IPSG for static hosts with IP filters on a Layer 2 access port and to verify the three valid IP bindings on the interface Fa4/3:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# ip device tracking
Switch(config)# interface fastEthernet 4/3
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 10
Switch(config-if)# ip device tracking maximum 5
Switch(config-if)# ip verify source tracking
Switch(config-if)# end

Switch# show ip verify source
```

Interface	Filter-type	Filter-mode	IP-address	Mac-address	Vlan
Fa4/3	ip trk	active	40.1.1.24		10
Fa4/3	ip trk	active	40.1.1.20		10
Fa4/3	ip trk	active	40.1.1.21		10

The following example shows how to enable IPSG for static hosts with IP MAC filters on a Layer 2 access port, to verify the five valid IP-MAC bindings on the interface Fa4/3, and to verify that the number of bindings on this interface has reached the maximum limit:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# ip device tracking
Switch(config)# interface fastEthernet 4/3
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 1
Switch(config-if)# ip device tracking maximum 5
Switch(config-if)# switchport port-security
Switch(config-if)# switchport port-security maximum 5
Switch(config-if)# ip verify source tracking port-security
Switch(config-if)# end

Switch# show ip verify source
```

Interface	Filter-type	Filter-mode	IP-address	Mac-address	Vlan
Fa4/3	ip-mac trk	active	40.1.1.24	00:00:00:00:03:04	1
Fa4/3	ip-mac trk	active	40.1.1.20	00:00:00:00:03:05	1
Fa4/3	ip-mac trk	active	40.1.1.21	00:00:00:00:03:06	1
Fa4/3	ip-mac trk	active	40.1.1.22	00:00:00:00:03:07	1
Fa4/3	ip-mac trk	active	40.1.1.23	00:00:00:00:03:08	1

The following example displays all IP-to-MAC binding entries for all interfaces. The CLI displays all active as well as inactive entries. When a host is learned on a interface, the new entry is marked as active. When the same host is disconnected from the current interface and connected to a different interface, a new IP-to-MAC binding entry is displayed as active as soon as the host is detected. The old entry for this host on the previous interface is now marked as inactive.

```
Switch# show ip device tracking all
Global IP Device Tracking for clients = Enabled
Global IP Device Tracking Probe Count = 3
Global IP Device Tracking Probe Interval = 300
Global IP Device Tracking Probe Delay Interval = 10
```

IP Address	MAC Address	Vlan	Interface	STATE
200.1.1.8	0001.0600.0000	8	GigabitEthernet3/1	INACTIVE
200.1.1.9	0001.0600.0000	8	GigabitEthernet3/1	INACTIVE
200.1.1.10	0001.0600.0000	8	GigabitEthernet3/1	INACTIVE
200.1.1.1	0001.0600.0000	9	GigabitEthernet4/1	ACTIVE
200.1.1.1	0001.0600.0000	8	GigabitEthernet3/1	INACTIVE
200.1.1.2	0001.0600.0000	9	GigabitEthernet4/1	ACTIVE
200.1.1.2	0001.0600.0000	8	GigabitEthernet3/1	INACTIVE
200.1.1.3	0001.0600.0000	9	GigabitEthernet4/1	ACTIVE
200.1.1.3	0001.0600.0000	8	GigabitEthernet3/1	INACTIVE
200.1.1.4	0001.0600.0000	9	GigabitEthernet4/1	ACTIVE
200.1.1.4	0001.0600.0000	8	GigabitEthernet3/1	INACTIVE
200.1.1.5	0001.0600.0000	9	GigabitEthernet4/1	ACTIVE
200.1.1.5	0001.0600.0000	8	GigabitEthernet3/1	INACTIVE
200.1.1.6	0001.0600.0000	8	GigabitEthernet3/1	INACTIVE
200.1.1.7	0001.0600.0000	8	GigabitEthernet3/1	INACTIVE

The following example displays all active IP-to-MAC binding entries for all interfaces:

```
Switch# show ip device tracking all active
IP Device Tracking = Enabled
IP Device Tracking Probe Count = 3
IP Device Tracking Probe Interval = 30
```

IP Address	MAC Address	Vlan	Interface	STATE
200.1.1.1	0001.0600.0000	9	GigabitEthernet4/1	ACTIVE
200.1.1.2	0001.0600.0000	9	GigabitEthernet4/1	ACTIVE
200.1.1.3	0001.0600.0000	9	GigabitEthernet4/1	ACTIVE
200.1.1.4	0001.0600.0000	9	GigabitEthernet4/1	ACTIVE
200.1.1.5	0001.0600.0000	9	GigabitEthernet4/1	ACTIVE

The following example displays all inactive IP-to-MAC binding entries for all interfaces. The host was first learned on GigabitEthernet 3/1 then moved to GigabitEthernet 4/1. The IP-to-MAC binding entries learned on GigabitEthernet 3/1 are marked as inactive.

```
Switch# show ip device tracking all inactive
IP Device Tracking = Enabled
IP Device Tracking Probe Count = 3
IP Device Tracking Probe Interval = 30
```

IP Address	MAC Address	Vlan	Interface	STATE
200.1.1.8	0001.0600.0000	8	GigabitEthernet3/1	INACTIVE
200.1.1.9	0001.0600.0000	8	GigabitEthernet3/1	INACTIVE
200.1.1.10	0001.0600.0000	8	GigabitEthernet3/1	INACTIVE
200.1.1.1	0001.0600.0000	8	GigabitEthernet3/1	INACTIVE
200.1.1.2	0001.0600.0000	8	GigabitEthernet3/1	INACTIVE
200.1.1.3	0001.0600.0000	8	GigabitEthernet3/1	INACTIVE

```

200.1.1.4      0001.0600.0000  8  GigabitEthernet3/1  INACTIVE
200.1.1.5      0001.0600.0000  8  GigabitEthernet3/1  INACTIVE
200.1.1.6      0001.0600.0000  8  GigabitEthernet3/1  INACTIVE
200.1.1.7      0001.0600.0000  8  GigabitEthernet3/1  INACTIVE

```

The following example displays the count of all IP device tracking host entries for all interfaces:

```

Switch# show ip device tracking all count
Total IP Device Tracking Host entries: 5
-----
Interface                Maximum Limit      Number of Entries
-----
Fa4/3                     5

```

Configuring IPSG for Static Hosts on a PVLAN Host Port

You can configure IPSG for static hosts on a PVLAN host port.

To enable IPSG for static hosts with IP filters on a PVLAN host port, perform this task:

	Command	Purpose
Step 1	Switch(config)# vlan <i>n1</i>	Enters configuration VLAN mode.
Step 2	Switch(config-vlan)# private-vlan primary	Establishes a primary VLAN on a PVLAN port.
Step 3	Switch(config-vlan)# exit	Exits VLAN configuration mode.
Step 4	Switch(config)# vlan <i>n2</i>	Enters configuration VLAN mode.
Step 5	Switch(config-vlan)# private-vlan isolated	Establishes an isolated VLAN on a PVLAN port.
Step 6	Switch(config-vlan)# exit	Exits VLAN configuration mode.
Step 7	Switch(config)# vlan <i>n1</i>	Enters configuration VLAN mode.
Step 8	Switch(config-vlan)# private-vlan association <i>201</i>	Associates the VLAN on an isolated PVLAN port.
Step 9	Switch(config-vlan)# exit	Exits VLAN configuration mode.
Step 10	Switch(config)# interface fastEthernet <i>a/b</i>	Enters interface configuration mode.
Step 11	Switch(config-if)# switchport mode private-vlan host	(Optional) Establishes a port as a PVLAN host.
Step 12	Switch(config-if)# switchport private-vlan host-association <i>a b</i>	(Optional) Associates this port with the corresponding PVLAN.
Step 13	Switch(config-if)# ip device tracking maximum <i>n</i>	Establishes a maximum limit for the bindings on this port.
Step 14	Switch(config-if)# ip verify source tracking [port-security]	Activates IPSG for static hosts on this port.
Step 15	Switch(config-if)# end	Exits configuration interface mode.
Step 16	Switch# show ip device tracking all	Verifies the configuration.
Step 17	Switch# show ip verify source <i>interface-name</i>	Verifies the configuration.

This example shows how to enable IPSG for static hosts with IP filters on a PVLAN host port:

```

Switch(config)# vlan 200
Switch(config-vlan)# private-vlan primary
Switch(config-vlan)# exit
Switch(config)# vlan 201

```

```

Switch(config-vlan)# private-vlan isolated
Switch(config-vlan)# exit
Switch(config)# vlan 200
Switch(config-vlan)# private-vlan association 201
Switch(config-vlan)# exit
Switch(config)# int fastEthernet 4/3
Switch(config-if)# switchport mode private-vlan host
Switch(config-if)# switchport private-vlan host-association 200 201
Switch(config-if)# ip device tracking maximum 8
Switch(config-if)# ip verify source tracking

```

```

Switch# show ip device tracking all
Global IP Device Tracking for clients = Enabled
Global IP Device Tracking Probe Count = 3
Global IP Device Tracking Probe Interval = 300
Global IP Device Tracking Probe Delay Interval = 10

```

IP Address	MAC Address	Vlan	Interface	STATE
40.1.1.24	0000.0000.0304	200	FastEthernet4/3	ACTIVE
40.1.1.20	0000.0000.0305	200	FastEthernet4/3	ACTIVE
40.1.1.21	0000.0000.0306	200	FastEthernet4/3	ACTIVE
40.1.1.22	0000.0000.0307	200	FastEthernet4/3	ACTIVE
40.1.1.23	0000.0000.0308	200	FastEthernet4/3	ACTIVE

The output shows the five valid IP-to-MAC bindings that have been learned on the interface Fa4/3. For the PVLAN, the bindings are associated with primary VLAN ID. In this example, the primary VLAN ID, 200, is shown in the table.

```

Switch# show ip verify source

```

Interface	Filter-type	Filter-mode	IP-address	Mac-address	Vlan
Fa4/3	ip trk	active	40.1.1.23		200
Fa4/3	ip trk	active	40.1.1.24		200
Fa4/3	ip trk	active	40.1.1.20		200
Fa4/3	ip trk	active	40.1.1.21		200
Fa4/3	ip trk	active	40.1.1.22		200
Fa4/3	ip trk	active	40.1.1.23		201
Fa4/3	ip trk	active	40.1.1.24		201
Fa4/3	ip trk	active	40.1.1.20		201
Fa4/3	ip trk	active	40.1.1.21		201
Fa4/3	ip trk	active	40.1.1.22		201

The output shows that the five valid IP-to-MAC bindings are on both the primary and secondary VLAN.



DHCPv6 Options Support

This module describes the Dynamic Host Control Protocol Version 6 (DHCPv6) Relay Agent, DHCPv6 Interface-ID, Lightweight DHCPv6 Relay Agent (LRDA), and CAPWAP Access Controller DHCP Option 52 features.

This module consists of these sections:

- [Restrictions for DHCPv6 Options Support, page 61-1](#)
- [Information About DHCPv6 Options Support, page 61-2](#)
- [How to Configure DHCPv6 Options Support, page 61-5](#)
- [Configuration Examples for DHCPv6 Options Support, page 61-9](#)
- [Additional References for DHCPv6 Options Support, page 61-10](#)
- [Feature Information for DHCPv6 Options Support, page 61-12](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see publications at this location:

[*Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch*](#)

If a command is not in the *Catalyst 4500 Series Switch Command Reference*, you can locate it in the Cisco IOS library, at this location:

[*Cisco IOS Master Command List, All Releases*](#)

Restrictions for DHCPv6 Options Support

The following restrictions apply to the Lightweight DHCPv6 Relay Agent (LDRA) feature:

- An interface or port cannot be configured as both client facing and server facing at the same time.
- Access nodes implementing LDRA do not support IPv6 control or routing.
- Unlike a DHCPv6 relay agent, an LDRA does not implement any IPv6 control functions (like Internet Control Message Protocol Version 6 [ICMPv6] functions), nor does it have any routing capability in the node.

Information About DHCPv6 Options Support

- [DHCPv6 Relay Agent Overview, page 61-2](#)
- [DHCPv6 Relay Options: Remote-ID, page 61-2](#)
- [DHCPv6 Interface-ID, page 61-3](#)
- [Lightweight DHCPv6 Relay Agent, page 61-3](#)
- [Interoperability between DHCPv6 Relay Agents and LDRA, page 61-3](#)
- [LDRA for VLANs and Interfaces, page 61-4](#)
- [CAPWAP Access Controller DHCPv6 Option, page 61-4](#)

DHCPv6 Relay Agent Overview

A DHCPv6 relay agent is any host that forwards DHCP packets between clients and servers. Relay agents are used to forward requests and replies between clients and servers when they are not on the same physical subnet.

A DHCPv6 relay agent, which may reside on the client's link, is used to relay messages between the client and the server. The DHCPv6 relay agent operation is transparent to the client. A DHCPv6 client locates a DHCPv6 server using a reserved, link-scoped multicast address. For direct communication between the DHCPv6 client and the DHCPv6 server, both of them must be attached to the same link. However, in some situations where ease of management, economy, or scalability is a concern, it is desirable to allow a DHCPv6 client to send a message to a DHCPv6 server that is not connected to the same link.

DHCPv6 Relay Options: Remote-ID

The DHCPv6 Remote ID Option feature adds the remote identification (remote-ID) option to relayed (RELAY-FORWARD) DHCPv6 packets.

The remote-ID option provides information to the DHCPv6 server, which includes port information, the system's DHCP Unique Identifier (DUID), and the virtual LAN (VLAN) ID. This information can be used to uniquely identify both the relay and the port on the relay through which the client packet arrived. The DHCPv6 server uses this information to select parameters specific to a particular user, host, or subscriber modem.

The addition of the remote-ID option to the RELAY-FORWARD packet occurs automatically and no user configuration is necessary.

The DHCPv6 server does not need to echo the remote-ID option in the RELAY-REPLY packet. The Internet Assigned Numbers Authority (IANA) has assigned the DHCPv6 option code 37 for the relay agent remote-ID option.

If the remote-ID option is included in the RELAY-REPLY packet, the option is removed from the packet before it is relayed to the client.

DHCPv6 Interface-ID

The interface-ID option is used by relay agents to decide which interface should be used when forwarding a RELAY-REPLY packet. If a relay agent receives a RELAY-REPLY message with an interface-ID option, the message is relayed to the client through the interface identified by the option.

The server must copy the interface-ID option from the RELAY-FORWARD message into the RELAY-REPLY message the server sends to the relay agent in response to the RELAY-FORWARD message. This option must not appear in any message except a RELAY-FORWARD or a RELAY-REPLY message.

Servers can use the interface-ID for parameter assignment policies. The interface-ID must be considered as an opaque value, with policies based on exact match only; that is, interface-ID must not be internally parsed by the server. The interface-ID value for an interface must be stable and remain unchanged, for example, after the relay agent is restarted; if the interface-ID changes, a server will not be able to use it reliably in parameter assignment policies.

Lightweight DHCPv6 Relay Agent

The Lightweight DHCPv6 Relay Agent feature allows relay agent information to be inserted by an access node that performs a link-layer bridging (non-routing) function. Lightweight DHCPv6 Relay Agent (LDRA) functionality can be implemented in existing access nodes, such as DSL access multiplexers (DSLAMs) and Ethernet switches, that do not support IPv6 control or routing functions. LDRA is used to insert relay-agent options in DHCPv6 message exchanges primarily to identify client-facing interfaces. LDRA functionality can be enabled on an interface and a VLAN.

An LDRA device or interface has the following features:

- Maintains interoperability with existing DHCPv6 relay agents and servers.
- Is functionally the equivalent of a Layer 2 relay agent, without routing capabilities.

**Note**

LDRA is a device or interface on which LDRA functionality is configured.

Background

A variety of different link-layer network topologies exist for the aggregation of IPv6 nodes into one or more devices. In Layer 2 aggregation networks (IEEE 802.1D bridging or similar) that have many nodes on a single link, a DHCPv6 server or DHCP relay agent normally does not recognize how a DHCP client is attached to a network. LDRA allows relay-agent information, including the Interface-ID option, to be inserted by the access node so that the information may be used by the DHCPv6 server for client identification.

Interoperability between DHCPv6 Relay Agents and LDRA

DHCPv6 relay agents are used to forward DHCPv6 messages between a client and a server when the client and server are not on the same IPv6 link. A DHCPv6 relay agent also adds an interface ID option in the upstream DHCPv6 message (from client-to-server) to identify the interface on which the client is connected. This information is used by the DHCPv6 relay agent while forwarding the downstream DHCPv6 message to the DHCPv6 client. The DHCPv6 relay agent is implemented alongside the routing functionality on the common node.

To maintain interoperability with existing DHCP relays and servers, LDRA implements the same message types (RELAY-FORWARD and RELAY-REPLY) as a DHCPv6 relay agent. LDRA allows relay-agent information to be inserted by an access node that performs a link-layer bridging (that is, non-routing) function. The LDRA resides on the same IPv6 link as the client and a DHCPv6 relay agent or server.

LDRA for VLANs and Interfaces

You can configure LDRA on VLANs and interfaces. LDRA is not enabled by default. You must enable it on the VLAN or interface first.

In a typical deployment, a majority of the interfaces or ports on a device are client facing. In such a scenario, you can configure LDRA functionality on the VLAN. When you configure LDRA on a VLAN, the functionality is configured on all ports or interfaces within the VLAN. Instead of configuring LDRA functionality individually on interfaces and ports within a VLAN, you can configure LDRA on the entire VLAN. As a result, all ports or interfaces associated with the VLAN will be configured as client facing.

You can also configure LDRA functionality on a specific interface or port. An interface or port can be configured as client-facing trusted, client-facing untrusted, or server facing.

The LDRA configuration on a VLAN has to be configured as trusted or untrusted. An LDRA must implement a configuration setting for all client-facing interfaces, marking them as trusted or as untrusted.

By default, any interface that is configured as client facing will be configured as an untrusted interface. When a client-facing interface is deemed untrusted, LDRA will discard any message of type RELAY-FORWARD received from the client-facing interface.

CAPWAP Access Controller DHCPv6 Option

The Control And Provisioning of Wireless Access Points (CAPWAP) protocol allows lightweight access points to use DHCPv6 to discover a Wireless Controller to which it can connect. CAPWAP is a standard, interoperable protocol that enables a controller to manage a collection of wireless access points.

Wireless access points use the DHCPv6 option 52 (RFC 5417) to supply the IPv6 management interface addresses of the primary, secondary, and tertiary Wireless Controllers.

Both stateless and stateful DHCPv6 addressing modes are supported. In stateless mode, access points obtain IPv6 address using the Stateless Address AutoConfiguration (SLAAC), while additional network information (not obtained from router advertisements) is obtained from a DHCPv6 server. In stateful mode, access points obtain both IPv6 addressing and additional network information exclusively from the DHCPv6 server. In both modes, a DHCPv6 server is required to provide option 52 if Wireless Controller discovery using DHCPv6 is required.


How to Configure DHCPv6 Options Support

- [Configuring the DHCPv6 Relay Agent, page 61-5](#)
- [Configuring LDRA Functionality on a VLAN, page 61-5](#)
- [Configuring LDRA Functionality on an Interface, page 61-6](#)
- [Verifying the LRDA Configuration, page 61-7](#)
- [Verifying the LRDA Configuration, page 61-7](#)

Configuring the DHCPv6 Relay Agent


	Command or Action	Purpose
Step 1	Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	Device# configure terminal	Enters global configuration mode.
Step 3	Device(config)# interface <i>type number</i>	Configures an interface and enters interface configuration mode.
Step 4	Device(config-if)# ipv6 dhcp relay destination <i>ipv6-address</i> [<i>interface-type</i> <i>interface-number</i>]	Specifies a destination address to which client packets are forwarded and enables the DHCPv6 relay service on the interface.
Step 5	Device(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring LDRA Functionality on a VLAN

	Command or Action	Purpose
Step 1	Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	Device# configure terminal	Enters global configuration mode.
Step 3	Device(config)# ipv6 dhcp-ldra { enable disable remote-id }	Enables LDRA functionality globally. <div>  <p>Note You must enable the LDRA functionality in global configuration mode before configuring it on an interface.</p> </div>
Step 4	Device(config)# vlan configuration <i>vlan-number</i>	Specifies a VLAN number and enters VLAN configuration mode.

	Command or Action	Purpose
Step 5	Device(config-vlan-config)# ipv6 dhcp ldra attach-policy { client-facing-trusted client-facing-untrusted }	Enables the LDRA functionality on a specified VLAN. <ul style="list-style-type: none"> The client-facing-trusted keyword configures all ports or interfaces associated with the VLAN as client facing, trusted ports. The client-facing-untrusted keyword configures all ports or interfaces associated with the VLAN as client facing, untrusted ports.
Step 6	Device(config-vlan-config)# exit	Exits VLAN configuration mode and returns to global configuration mode.
Step 7	Device(config)# interface <i>type number</i>	Configures an interface and enters interface configuration mode.
Step 8	Device(config-if)# switchport	Switches an interface that is in Layer 3 mode into Layer 2 mode for Layer 2 configuration.
Step 9	Device(config-if)# switchport access vlan <i>vlan-number</i>	Specifies that an interface operates in the specified VLAN instead of the default VLAN in interface configuration mode.
Step 10	Device(config-if)# ipv6 dhcp ldra attach-policy { client-facing-trusted client-facing-untrusted client-facing-disable server-facing }	Enables LDRA functionality on a specified interface or port. <ul style="list-style-type: none"> The server-facing keyword specifies an interface or port as server facing.
Step 11	Device(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring LDRA Functionality on an Interface

	Command or Action	Purpose
Step 1	Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	Device# configure terminal	Enters global configuration mode.
Step 3	Device(config)# ipv6 dhcp ldra { enable disable remote-id }	Enables LDRA functionality globally. <div>  <div> Note You must enable the LDRA functionality in global configuration mode before configuring it on an interface. </div> </div>
Step 4	Device(config)# interface <i>type number</i>	Configures an interface and enters interface configuration mode.
Step 5	Device(config-if)# switchport	Switches an interface that is in Layer 3 mode into Layer 2 mode for Layer 2 configuration.
Step 6	Device(config-if)# ipv6 dhcp ldra interface-id <i>interface-id</i>	Configures LDRA interface ID on a port or an interface.
Step 7	Device(config-if)# end	Exits VLAN configuration mode and returns to privileged EXEC mode.

Verifying the LRDA Configuration

Step 1 **show ipv6 dhcp interface**

Displays DHCPv6 interface information.

Example:

```
Device# show ipv6 dhcp interface

GigabitEthernet0/1 is in relay mode
Relay destinations:
  3FFB:C00:C18:6:A8BB:CCFF:FE03:2701
Serial3/0 is in relay mode
Relay destinations:
  3FFB:C00:C18:6:A8BB:CCFF:FE03:2600
  FE80::A8BB:CCFF:FE03:2801 on Serial3/0
  FF05::1:3
```

Step 2 **show ipv6 dhcp-ldra**

Displays LDRA configuration details. The fields in the example given below are self-explanatory.

Example:

```
Device# show ipv6 dhcp-ldra

DHCPv6 LDRA is Enabled.
DHCPv6 LDRA policy: client-facing-disable
Target: none
DHCPv6 LDRA policy: client-facing-trusted
Target: vlan 5
DHCPv6 LDRA policy: client-facing-untrusted
Target: none
DHCPv6 LDRA policy: server-facing
Target: Gi1/0/7
```

Step 3 **show ipv6 dhcp-ldra statistics**

Displays LDRA configuration statistics before and after initiating a DHCP session. The fields in the examples below are self-explanatory.

Example:

```
Device# show ipv6 dhcp-ldra statistics

DHCPv6 LDRA client facing statistics.
Messages received 2
Messages sent 2
Messages discarded 0
Messages Received
SOLICIT 1
REQUEST 1
Messages Sent
RELAY-FORWARD 2
DHCPv6 LDRA server facing statistics.
Messages received 2
Messages sent 2
Messages discarded 0
Messages Received
RELAY-REPLY 2
Messages Sent
```

```
ADVERTISE 1
REPLY 1
```

Step 4 **debug ipv6 dhcp-ldra all**

Enables all LDRA debugging flows. The fields in the example below are self-explanatory.

Example:

```
Device# debug ipv6 dhcp-ldra all

05:44:10: DHCPv6 LDRA API: Entered ipv6_dhcp_ldra_post_processor.
05:44:10: DHCPv6 LDRA EVENT: [Gi1/0/3 Vlan 5] Received SOLICIT from 2001:DB8:1::1 to
FF02::1:2.
05:44:10:
05:44:10:
05:44:10:
05:44:10:
05:44:10: 000300010015F906981B
05:44:10: option ORO(6), len 4
05:44:10: DNS-SERVERS,DOMAIN-LIST
05:44:10: option IA-NA(3), len 12
05:44:10: IAID 0x00040001, T1 0, T2 0
05:44:10: DHCPv6 LDRA API: Entered dhcpv6_ldra_client_facing_new_pak.
05:44:10: DHCPv6 LDRA EVENT: [Vlan 5] Sending RELAY-FORWARD from 2001:DB8:1::1
to FF02::1:2.
!
!
!
```

Configuring CAPWAP Access Points

	Command or Action	Purpose
Step 1	Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	Device# configure terminal	Enters global configuration mode.
Step 3	Device(config)# ipv6 dhcp pool <i>poolname</i>	Configures a DHCPv6 server configuration information pool and enters DHCPv6 pool configuration mode.
Step 4	Device(config-dhcpv6)# capwap-ac address <i>ipv6-address</i>	Configures CAPWAP access controller address.
Step 5	Device(config-dhcpv6)# end	Exits DHCPv6 pool mode and returns to privileged EXEC mode.

Configuration Examples for DHCPv6 Options Support

- [Example: Configuring the DHCPv6 Relay Agent, page 61-9](#)
- [Example: Configuring LDRA Functionality on a VLAN, page 61-9](#)
- [Example: Configuring LDRA Functionality on an Interface, page 61-9](#)
- [Example: Configuring CAPWAP Access Points, page 61-10](#)

Example: Configuring the DHCPv6 Relay Agent

```
Device> enable
Device# configure terminal
Device(config)# interface gigabitethernet 1/1
Device(config-if)# ipv6 dhcp relay destination FE80::250:A2FF:FEBF:A056 gigabitethernet
0/1
Device(config-if)# end
```

Example: Configuring LDRA Functionality on a VLAN

The following example shows how to configure Lightweight DHCPv6 Relay Agent (LDRA) on a VLAN numbered 5.

```
Device> enable
Device# configure terminal
Device(config)# ipv6 dhcp-ldra enable
Device(config)# vlan configuration 5
Device(config-vlan-config)# ipv6 dhcp ldra attach-policy client-facing-trusted
Device(config-vlan-config)# exit
Device(config)# interface gigabitethernet 0/0
Device(config-if)# switchport
Device(config-if)# switchport access vlan 5
Device(config-if)# ipv6 dhcp-ldra attach-policy client-facing-trusted
Device(config-if)# exit
Device(config)# interface gigabitethernet 1/0
Device(config-if)# switchport
Device(config-if)# switchport access vlan 5
Device(config-if)# ipv6 dhcp-ldra attach-policy server-facing
Device(config-if)# end
```

Example: Configuring LDRA Functionality on an Interface

In the following example, LDRA is configured on the interfaces GigabitEthernet 0/0:

```
Device> enable
Device# configure terminal
Device(config)# ipv6 dhcp-ldra enable
Device(config)# interface gigabitethernet 0/0
Device(config-if)# switchport
Device(config-if)# ipv6 dhcp-ldra interface-id 2
Device(config-if)# end
```

Example: Configuring CAPWAP Access Points

```
Device> enable
Device# configure terminal
Device(config)# ipv6 dhcp pool pool1
Device(config-dhcpv6)# capwap-ac address 2001:DB8::1
Device(config-dhcpv6)# end
Device#
```

Additional References for DHCPv6 Options Support

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Catalyst 4500 commands	Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch

Standards and RFCs

Standard/RFC	Title
RFC 3315	<i>Dynamic Host Configuration Protocol for IPv6</i>
RFC 4649	<i>Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Relay Agent Remote-ID Option</i>
RFC 5417	<i>Control And Provisioning of Wireless Access Points (CAPWAP) Access Controller DHCP Option</i>
RFC 6221	<i>Lightweight DHCPv6 Relay Agent</i>

MIBs

MIB	MIBs Link
•	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
<p>The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.</p>	<p>http://www.cisco.com/cisco/web/support/index.html</p>

Feature Information for DHCPv6 Options Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1 Feature Information for DHCPv6 Options Support

Feature Name	Releases	Feature Information
CAPWAP Access Controller DHCP Option 52	Cisco IOS Release 15.2(5)E2	The Control And Provisioning of Wireless Access Points (CAPWAP) protocol allows Lightweight Access Points to use DHCPv6 to discover a Wireless Controller to which it can connect.
DHCPv6 Interface-ID	Cisco IOS Release 15.2(5)E2	The interface-ID option is used by relay agents to decide which interface should be used when forwarding a RELAY-REPLY packet.
DHCPv6 Relay Agent	Cisco IOS Release 15.2(5)E2	A DHCP relay agent, which may reside on the client's link, is used to relay messages between the client and server.



Configuring Network Security with ACLs

This chapter describes how to use access control lists (ACLs) to configure network security on the Catalyst 4500 series switches.



Note

Catalyst 4500 series switches supports time-based ACLs.

This chapter consists of the following major sections:

- [About ACLs, page 62-2](#)
- [Hardware and Software ACL Support, page 62-6](#)
- [Troubleshooting High CPU Due to ACLs, page 62-7](#)
- [TCAM Programming and ACLs, page 62-10](#)
- [Layer 4 Operators in ACLs, page 62-11](#)
- [Configuring Unicast MAC Address Filtering, page 62-16](#)
- [Configuring Named MAC Extended ACLs, page 62-16](#)
- [Configuring EtherType Matching, page 62-17](#)
- [Configuring Named IPv6 ACLs, page 62-18](#)
- [Applying IPv6 ACLs to Layer 2 and 3 Interface, page 62-20](#)
- [Configuring VLAN Maps, page 62-21](#)
- [Displaying VLAN Access Map Information, page 62-28](#)
- [Using VLAN Maps with Router ACLs, page 62-28](#)
- [Configuring PACLS, page 62-31](#)
- [Using PACL with VLAN Maps and Router ACLs, page 62-36](#)
- [Configuring Object Group ACLs, page 62-39](#)
- [Configuring RA Guard, page 62-50](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About ACLs

This section includes these topics:

- [Overview, page 62-2](#)
- [Supported Features That Use ACLs, page 62-3](#)
- [Router ACLs, page 62-3](#)
- [Port ACLs, page 62-4](#)
- [Dynamic ACLs, page 62-5](#)
- [VLAN Maps, page 62-5](#)

Overview

An ACL is a collection of sequential permit and deny conditions that applies to packets. When a packet is received on an interface, the switch compares the fields in the packet against any applied ACLs to verify that the packet has the permissions required to be forwarded, based on the conditions specified in the access lists. It tests the packets against the conditions in an access list one-by-one. The first match determines whether the switch accepts or rejects the packets. Because the switch stops testing conditions after the first match, the order of conditions in the list is critical. If no conditions match, the switch drops the packet. If no restrictions exist, the switch forwards the packet; otherwise, the switch drops the packet.

Switches traditionally operate at Layer 2, switching traffic within a VLAN. Routers route traffic between VLANs at Layer 3. The Catalyst 4500 series switch can accelerate packet routing between VLANs by using Layer 3 switching. The Layer 3 switch bridges the packet, and then routes the packet internally without going to an external router. The packet is then bridged again and sent to its destination. During this process, the switch can control all packets, including packets bridged within a VLAN.

You configure access lists on a router or switch to filter traffic and provide basic security for your network. If you do not configure ACLs, all packets passing using the switch could be allowed on all parts of the network. You can use ACLs to control which hosts can access different parts of a network or to decide which types of traffic are forwarded or blocked at router interfaces. For example, you can allow e-mail traffic to be forwarded but not Telnet traffic. ACLs can be configured to block inbound traffic, outbound traffic, or both. However, on Layer 2 interfaces, you can apply ACLs only in the inbound direction.

An ACL contains an ordered list of access control entries (ACEs). Each ACE specifies permit or deny and a set of conditions the packet must satisfy in order to match the ACE. The meaning of permit or deny depends on the context in which the ACL is used. Negative TCP flags such as -syn, -psh or -fin in ACEs are not considered when you apply IP ACLs. We recommend that you use positive TCP flags in ACEs.

**Note**

The Catalyst 4500 series switch does not support non-contiguous ports on the same ACE or on a download able ACE.

The Catalyst 4500 series switch supports three types of ACLs:

- IP ACLs, which filter IP traffic, including TCP, the User Datagram Protocol (UDP), Internet Group Management Protocol (IGMP), and Internet Control Message Protocol (ICMP)
- IPv6 ACLs
- MAC ACLs which match based on Ethernet addresses and Ether Type

Supported Features That Use ACLs

The switch supports three applications of ACLs to filter traffic:

- Router ACLs are applied to Layer 3 interfaces. They control the access of routed traffic between VLANs. All Catalyst 4500 series switches can create router ACLs, but you must have a Cisco IOS software image on your switch to apply an ACL to a Layer 3 interface and filter packets routed between VLANs.
- Port ACLs perform access control on traffic entering a Layer 2 interface. If insufficient hardware CAM entries exist, the output port ACL is not applied to the port and a warning message is given to user. (This restriction applies to all access group modes for output port ACLs.) When sufficient CAM entries exist, the output port ACL may be reapplied.

If there is any output port ACL configured on a Layer 2 port, then no VACL or router ACL can be configured on the VLANs that the Layer 2 port belongs to. Also, the reverse is true: port ACLs and VLAN-based ACLs (VACLs and router ACLs) are mutually exclusive on a Layer 2 port. This restriction applies to all access group modes. On the input direction, port ACLs, VLAN-based ACLs, and router ACLs can co-exist.

You can apply one IPv4 access list, one IPv6 access list and one MAC access list for a Layer 2 interface.

- You can use VLAN maps to filter traffic between devices in the same VLAN. You do not need the enhanced image to create or apply VLAN maps. VLAN maps are configured to control access based on Layer 3 addresses for IP. MAC addresses using Ethernet ACEs control the access of unsupported protocols. After you apply a VLAN map to a VLAN, all packets (routed or bridged) entering the VLAN are checked against that map. Packets can either enter the VLAN through a switch port or through a routed port after being routed.

You can use both router ACLs and VLAN maps on the same switch.

Router ACLs

You can apply one access list of each supported type to an interface.



Note

Catalyst 4500 series switches running Cisco IOS Release 12.2(40)SG do *not* support IPv6 port ACLs (PACLs).

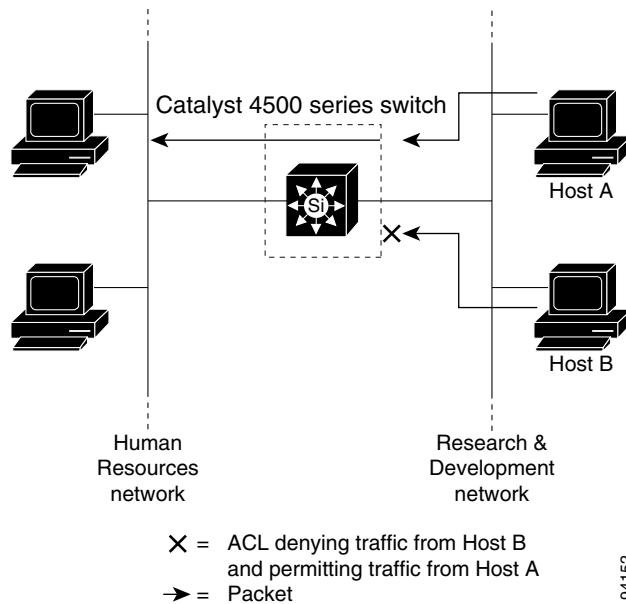
Multiple features can use one ACL for a given interface, and one feature can use multiple ACLs. When a single router ACL is used by multiple features, it is examined multiple times. The access list type determines the input to the matching operation:

- Standard IP access lists use source addresses for matching operations.
- Extended IP access lists use source and destination addresses and optional protocol type information for matching operations.

The switch examines ACLs associated with features configured on a given interface and a direction. As packets enter the switch on an interface, ACLs associated with all inbound features configured on that interface are examined. After packets are routed and before they are forwarded to the next hop, all ACLs associated with outbound features configured on the egress interface are examined.

ACLs permit or deny packet forwarding based on how the packet matches the entries in the ACL. For example, you can use access lists to allow one host to access a part of a network, but prevent another host from accessing the same part. In [Figure 62-1](#), ACLs applied at the router input allow Host A to access the Human Resources network, but prevent Host B from accessing the same network.

Figure 62-1 Using ACLs to Control Traffic to a Network



94152



Note

Starting IOS XE 3.11.0, Catalyst 4500 series switches do not support egress ACLs on a tunnel interface and on the source interface of the tunnel.

Port ACLs

You can also apply ACLs to Layer 2 interfaces on a switch. Port ACLs are supported on physical interfaces and EtherChannel interfaces. The following access lists are supported on Layer 2 interfaces:

- Standard IP access lists using source addresses
- Extended IP access lists using source and destination addresses and optional protocol type information
- IPv6 access lists using source and destination addresses and optional protocol type information
- MAC extended access lists using source and destination MAC addresses and optional protocol type information



Note

Negative TCP flags such as -syn, -psh or -fin in ACEs are not considered when you apply port ACLs. We recommend that you use positive TCP flags in ACEs.

As with router ACLs, the switch examines ACLs associated with features configured on a given interface and permits or denies packet forwarding based on how the packet matches the entries in the ACL. In the example in [Figure 62-1](#), if all workstations were in the same VLAN, ACLs applied at the Layer 2 input would allow Host A to access the Human Resources network, but prevent Host B from accessing the same network.

When you apply a port ACL to a trunk port, the ACL filters traffic on all VLANs present on the trunk port. When you apply a port ACL to a port with voice VLAN, the ACL filters traffic on both data and voice VLANs.

With port ACLs, you can filter IP traffic by using IP access lists and non-IP traffic by using MAC addresses. You can filter both IP and non-IP traffic on the same Layer 2 interface by applying both an IP access list and a MAC access list to the interface.

With port ACLs, you can filter IPv4 traffic with IPv4 access lists, IPv6 traffic with IPv6 access lists, and non-IP traffic with MAC access lists. You can filter multiple types of traffic simultaneously by applying ACLs of the appropriate type to the Layer 2 interface simultaneously.

**Note**

You cannot simultaneously apply more than one access list of a given type to a Layer 2 interface. If an IPv4, IPv6, or MAC access list is already configured on a Layer 2 interface, and you apply a new IPv4, IPv6 or MAC access list to the interface, the new ACL replaces the previously configured ACL of the same type.

Dynamic ACLs

Various security features, such as 802.1X, NAC and Web Authentication, are capable of downloading ACLs from a central server and applying them to interfaces. Prior to Cisco IOS Release 12.2(54)SG, these features required the explicit configuration of a standard port ACL

Starting with Cisco IOS Release 12.2(54)SG, a port ACL does not require configuration. For more details refer to the [“Removing the Requirement for a Port ACL”](#) section on [page 62-32](#).

VLAN Maps

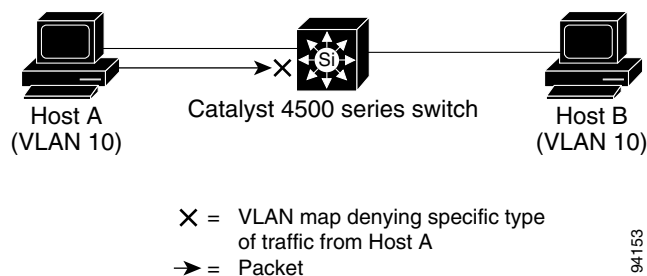
VLAN maps can control the access of all traffic in a VLAN. You can apply VLAN maps on the switch to all packets that are routed into or out of a VLAN or are bridged within a VLAN. VLAN maps are not defined by direction (input or output).

**Note**

Negative TCP flags such as -syn, -psh or -fin in ACEs are not considered when you apply VLAN ACLs. We recommend that you use positive TCP flags in ACEs.

You can configure VLAN maps to match Layer 3 addresses for IP traffic. Access of all non-IP protocols is controlled with a MAC address and an Ethertype using MAC ACLs in VLAN maps. (IP traffic is not controlled by MAC ACLs in VLAN maps.) You can enforce VLAN maps only on packets heading to the switch; you cannot enforce VLAN maps on traffic between hosts on a hub or on another switch connected to this switch.

With VLAN maps, forwarding packets is permitted or denied, based on the action specified in the map. [Figure 62-2](#) illustrates how a VLAN map is applied to deny a specific type of traffic from Host A in VLAN 10 from being forwarded.

Figure 62-2 Using VLAN Maps to Control Traffic

Hardware and Software ACL Support

This section describes how to determine whether ACLs are processed in hardware or in software:

- Flows that match a *deny* statement in standard and extended ACLs are dropped in hardware if ICMP unreachable messages are disabled.
- Flows that match a *permit* statement in standard ACLs are processed in hardware.
- The following ACL types are not supported in software:
 - Standard Xerox Network Systems (XNS) Protocol access list
 - Extended XNS access list
 - DECnet access list
 - Protocol type-code access list
 - Standard Internet Packet Exchange (IPX) access list
 - Extended IPX access list



Note

Packets that require logging are processed in software. A copy of the packets is sent to the CPU for logging while the actual packets are forwarded in hardware so that non-logged packet processing is not impacted.

By default, the Catalyst 4500 series switch sends ICMP unreachable messages when a packet is denied by an access list; these packets are not dropped in hardware but are forwarded to the switch so that it can generate the ICMP unreachable message.

To drop access list denied packets in hardware on the input interface, you must disable ICMP unreachable messages using the **no ip unreachables** interface configuration command. The **ip unreachables** command is enabled by default.



Note

Cisco IOS Release 12.2(40)SG does not support disabling IP unreachables on interfaces routing IPv6 traffic.



Note

If you set the **no ip unreachable** command on all Layer 3 interfaces, output ACL denied packets do not come to the CPU.

Troubleshooting High CPU Due to ACLs

Packets that match entries in fully programmed ACLs are processed in hardware.

Packets that match entries in partially programmed ACLs are processed in software using the CPU. This may cause high CPU utilization and packets to be dropped.

CPU spikes and connectivity loss may be observed when an ACL applied to a VLAN interface blocks HSRP management multicast traffic. In this scenario where both HSRP member devices may become Active, the resulting high number of IPv6 Neighbor Discovery packets being lifted to the CPU may cause a spike. To avoid this, ensure that the active and the standby devices in HSRP can communicate. Additionally, do not configure the IPv6 HSRP multicast address in the ACL.

To determine whether packets are being dropped due to high CPU utilization, reference the following:

http://www.cisco.com/en/US/products/hw/switches/ps663/products_tech_note09186a00804cef15.shtml

If the ACL and/or IPSG configuration is partially programmed in hardware, upgrading to Cisco IOS Release 12.2(31)SGA or later and resizing the TCAM regions may enable the ACLs to be fully programmed.

**Note**

Removal of obsolete TCAM entries can take several CPU process review cycles to complete. This process may cause some packets to be switched in software if the TCAM entry or mask utilization is at or near 100 percent.

Selecting Mode of Capturing Control Packets

In some deployments, you might want to bridge control packets in hardware rather than globally capture and forward them in software (at the expense of the CPU). The per-VLAN capture mode feature allows a Catalyst 4500 series switch to capture control packets only on selected VLANs and bridge traffic in hardware on all other VLANs.

When you use per-VLAN capture mode on your switch, it partially disables the global TCAM capture entries internally and attaches feature-specific capture ACLs on those VLANs that are enabled for snooping features. (All IP capture entries, and other non-IP entries are still captured through global TCAM.)

Because this feature controls specific control packets, they are captured only on the VLANs on which the internal ACLs are installed. On all other VLANs, the control traffic is bridged in hardware rather than forwarded to CPU.

The per-VLAN capture mode allows you to apply user-defined ACLs and QoS policers (in hardware) on control packets. You can also subject the aggregate control traffic ingressing the CPU to control plane policing.

When you use per-VLAN capture mode, the following four protocol groups are selectable per-VLAN. The breakdown of protocols intercepted by each group is as follows:

- IGMP Snooping—Cgmp, Ospf, Igmp, RipV2, Pim, 224.0.0.1, 224.0.0.2, 224.0.0.*
- DHCP Snooping—Client to Server, Server to Client, Server to Server

Because some of the groups have multiple overlapping ACEs (for example, 224.0.0.* is present in all the groups except for DHCP Snooping), turning on a certain group will also trigger the interception of some protocols from other groups.

Following are the programming triggers for the four protocol groups per-VLAN:

- IGMP Snooping should be enabled globally on a given VLAN.
- DHCP Snooping should be enabled globally on a given VLAN.

Guidelines and Restrictions



Note

Before configuring per-VLAN capture mode, you should examine your configuration to ensure that only the necessary features are enabled on the desired VLANs.

The following guidelines and restrictions apply to per-VLAN capture mode:

- Starting with Cisco IOS Release 15.0(2)SG on Supervisor Engine 6-E, 6L-E
Starting with Cisco IOS XE Release 3.2.0 on Supervisor Engine 7-E
Starting with Cisco IOS XE Release 3.2.0XO on Supervisor Engine 7L-E
Starting with Cisco IOS XE Release 3.6.0 on Supervisor Engine 8-E
Starting with Cisco IOS XE Release 3.10.0E on Supervisor Engine 9-E
globally reserved static ACL entries in the TCAM region for Layer 3 control packets are removed. The per-VLAN CTI command is not needed and does not apply for Layer 3 control packets because these packets are captured in per-VLAN fashion by default.

The following still function:

- Global static capture and CTI commands for IGMP or PIM packets (both use MAC addresses 224.0.0.1 and 224.0.0.2)
- Global and per-VLAN CTI for DHCP packets

With Cisco IOS Release 15.0(2)SG, per-VLAN capture of Layer 3 control packets is driven by SVI configuration. Except for IGMP, PIM, or DHCP, no special configuration is required.

Enabling per-VLAN capture mode consumes additional entries in the ACL/feature TCAM. The number of available TCAM entries depends on the type of supervisor engine. The entry and mask count further limits the utilization of the ACL/feature TCAM.

- On Supervisor Engines IV, V and V-10 GE a maximum of 32 action entries are supported at ingress and 64 entries are supported at egress. To avoid high CPU utilization, move ACEs with a 'log' action towards end of the ACL so that the available action index can be used optimally to process other ACE actions.
- Certain configurations can exhaust TCAM resource earlier in per-VLAN capture mode than in global capture mode (such as, when IP Source Guard is enabled on several interfaces or on a user-configured PACL).

You can resize TCAM regions to make more entries available to the PortAndVlan or PortOrVlan region based on the configuration. This allows more entries to be programmed in hardware before reaching the limit. When TCAM resources are exhausted, the packets are forwarded in software.

- In per-VLAN capture mode, you can configure ACLs to permit or deny control traffic on a VLAN or port.

Because security ACLs are terminated by an *implicit deny*, you must ensure that the ACLs are configured to permit the control packets necessary for the feature (protocol) to operate. However, this rule does not differ from the default behavior.

- CPU spikes and connectivity loss may be observed when an ACL applied to a VLAN interface blocks HSRP management multicast traffic. In this scenario where both HSRP member devices may become Active, the resulting high number of IPv6 Neighbor Discovery packets being lifted to the CPU may cause a spike. To avoid this, ensure that the active and the standby devices in HSRP can communicate. Additionally, do not configure the IPv6 HSRP multicast address in the ACL.

Selecting Control Packet Capture

To select the mode of capturing control packets, perform this task:

	Command	Purpose
Step 1	Switch# conf terminal	Enters configuration mode.
Step 2	Switch(config)# [no] access-list hardware capture mode [vlan global]	Selects mode of capturing control packets. The no form of the access-list hardware capture mode command restores the capture mode to the default, which is global.
Step 3	Switch(config)# end	Returns to enable mode.

This example shows how to configure a Catalyst 4500 series switch to capture control packets only on VLANs where features are enabled:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# access-list hardware capture mode vlan
Switch(config)# end
Switch#
```

This example shows how to configure a Catalyst 4500 series switch to capture control packets globally across all VLANs (using static ACL, the default mode):

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# access-list hardware capture mode global
Switch(config)# end
Switch#
```

When the capture mode changes from global to VLAN, the static CAM entries are invalidated. This creates a window during which control packets may pass through a Catalyst 4500 series switch without being intercepted to the CPU. This temporary situation is restored when the new per-VLAN capture entries are programmed in the hardware.

When you configure per-VLAN capture mode, you should examine the **show** commands for individual features to verify the appropriate behavior. In per-VLAN capture mode, the invalidated static CAM entries will appear as inactive in the output of the **show platform hardware acl input entries static** command. For example, the hit count for inactive entries will remain frozen because those entries are invalidated and applied per-VLAN where the feature is enabled. The following table lists the CamIndex entry types and the Cam regions.

CamIndex Entry Type	Active	Hit Count	CamRegion
50 PermitSharedStp	Y	3344	ControlPktsTwo
51 PermitLoopbackTest	Y	0	ControlPktsTwo

CamIndex Entry Type	Active	Hit Count	CamRegion
52 PermitProtTunnel	Y	0	ControlPktsTwo
53 CaptureCgmp	N	440	ControlPktsTwo
55 CaptureIgmpp	N	0	ControlPktsTwo
0 IgmppImv1ToCpu	N	N/A	0 (estimate)
0 IgmppGeneralQueryToCpu	N	N/A	0 (estimate)
2 IgmppToCpu	N	N/A	0 (estimate)
3 IgmppImv2ToCpu	N	N/A	0 (estimate)
2048 Ipv6MldGeneralQueryCopyToCpu	N	N/A	0 (estimate)
2050 Ipv6MldGeneralQueryCopyToCpu	N	N/A	0 (estimate)
2052 Ipv6MldQueryOrReportV1ToCpu	N	N/A	0 (estimate)
2054 Ipv6MldQueryOrReportV1ToCpu	N	N/A	0 (estimate)
2056 Ipv6MldReportV2ToCpu	N	N/A	0 (estimate)
2058 Ipv6MldReportV2ToCpu	N	N/A	0 (estimate)
2060 Ipv6MldDoneToCpu	N	N/A	0 (estimate)
2064 Ipv6MldImv2ToCpu	N	N/A	0 (estimate)

TCAM Programming and ACLs

You apply three types of hardware resources when you program ACLs and ACL-based features: mapping table entries (MTEs), profiles, and TCAM value/mask entries. If any of these resources are exhausted, packets are sent to the CPU for software-based processing.



Note

Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, 7-E, 6L-E, and 6-E automatically manage the available resources. Because masks are not shared on the supervisor engines, only one programming algorithm exists. No regions exist so region resizing is not needed.

If you exhaust resources on the supervisor engine, you should consider reducing the complexity of your configuration.



Note

When an interface is in down state, TCAMs are not consumed for RACLs, but are for PACLs.



Note

TCAM resources are replicated or shared based on the feature combinations applied on the interfaces. For example, if the same ACL and flow monitor configurations are applied on two different interfaces, TCAM resources are shared between the two interfaces. But if multicast routing is added on any one of the interfaces, then TCAM resources are replicated and not shared.

Layer 4 Operators in ACLs

The following sections provide guidelines and restrictions for configuring ACLs that include Layer 4 port operations:

- [Restrictions for Layer 4 Operations, page 62-11](#)
- [Configuration Guidelines for Layer 4 Operations, page 62-12](#)
- [Using ACLs to Filter TCP Flags and How ACL Processing Impacts CPU, page 62-13](#)

Restrictions for Layer 4 Operations



Note

Cisco IOS XE Release 3.7.0E and Cisco IOS Release 15.2(3)E do not support the configuration of named ACLs for noncontiguous ports on an ACE.

You can specify these operator types, each of which uses one Layer 4 operation in the hardware:

- gt (greater than)
- lt (less than)
- neq (not equal)
- range (inclusive range)

The limits on the number of Layer 4 operations differ for each type of ACL, and can also vary based on other factors: whether an ACL is applied to incoming or outgoing traffic, whether the ACL is a security ACL or is used as a match condition for a QoS policy, and whether IPv6 ACLs are being programmed using the compressed flow label format.



Note

The IPv6 compressed flow label format uses the Layer 2 Address Table to compress a portion of the IPv6 source address of each ACE in the ACL. The extra space freed in the flow label can then be used to support more Layer 4 operations. For this compression to be used, the IPv6 ACL cannot contain any ACEs that mask in only a portion of the bottom 48 bits of the source IPv6 address.

Generally, you will receive at most the following number of Layer 4 operations on the same ACL:

Direction	Protocol	Type	Operations
Input	IPv4	Security	14
Input	IPv6	Compressed Security	16
Input	IPv6	Uncompressed Security	2
Input	IPv4	QoS	4
Input	IPv6	Compressed QoS	10
Input	IPv6	Uncompressed QoS	4
Output	IPv4	Security	16
Output	IPv6	Compressed Security	18
Output	IPv6	Uncompressed Security	4
Output	IPv4	QoS	4
Output	IPv6	Compressed QoS	10
Output	IPv6	Uncompressed QoS	4



Note

Where up to 16 operations are supported, the seventeenth will trigger an expansion.

If you exceed the number of available Layer 4 operations, each new operation might cause the affected ACE to be translated into multiple ACEs in the hardware. If this translation fails, packets are sent to the CPU for software processing.

When you globally enable the **ipv6 multicast-routing** and **ipv6 routing** global configuration commands, a reduced number of Layer 4 operations are available for use in IPv6 ACL or QoS. Additionally, the "eq" operator consumes a Layer 4 Operation if it is used to match a source port.

Configuration Guidelines for Layer 4 Operations

When using Layer 4 operators, consider these guidelines:

- Layer 4 operations are considered different if the operator or operand differ. For example, the following ACL contains three different Layer 4 operations because gt 10 and gt 11 are considered two different Layer 4 operations:

```
... gt 10 permit
... lt 9 deny
... gt 11 deny
```



Note The eq operator can be used an unlimited number of times because eq does not use a Layer 4 operation in hardware.

- Layer 4 operations are considered different if the same operator/operand couple applies once to a source port and once to a destination port, as in the following example:

```
... Src gt 10....
... Dst gt 10
```

A more detailed example follows:

```
access-list 101
... (dst port) gt 10 permit
... (dst port) lt 9 deny
... (dst port) gt 11 deny
... (dst port) neq 6 permit
... (src port) neq 6 deny
... (dst port) gt 10 deny

access-list 102
... (dst port) gt 20 deny
... (src port) lt 9 deny
... (src port) range 11 13 deny
... (dst port) neq 6 permit
```

Access lists 101 and 102 use the following Layer 4 operations:

- Access list 101 Layer 4 operations: 5
 - gt 10 permit and gt 10 deny both use the same operation because they are identical and both operate on the destination port.
- Access list 102 Layer 4 operations: 4
- Total Layer 4 operations: 8 (due to sharing between the two access lists)
 - neq6 permit is shared between the two ACLs because they are identical and both operate on the same destination port.
- A description of the Layer 4 operations usage is as follows:

- Layer 4 operation 1 stores gt 10 permit and gt 10 deny from ACL 101
- Layer 4 operation 2 stores lt 9 deny from ACL 101
- Layer 4 operation 3 stores gt 11 deny from ACL 101
- Layer 4 operation 4 stores neg 6 permit from ACL 101 and 102
- Layer 4 operation 5 stores neg 6 deny from ACL 101
- Layer 4 operation 6 stores gt 20 deny from ACL 102
- Layer 4 operation 7 stores lt 9 deny from ACL 102
- Layer 4 operation 8 stores range 11 13 deny from ACL 102

Using ACLs to Filter TCP Flags and How ACL Processing Impacts CPU

You can use IPv4 or IPv6 ACLs to filter TCP flags. You do this by configuring ACEs that make up an access list to allow matching on a flag that is set.

You use a combination of flags on which to filter; these combinations are processed in hardware. Only the following combinations are supported (applicable to IPv4 and IPv6 ACLs) and the flags must be used in the specified combination:

- **rst** and **ack**—equivalent to the keyword **established**.
 - **rst**—The reset flag indicates that the receiver should delete the connection without further interaction.
 - **ack**—The acknowledge flag indicates that the acknowledgment field of a segment specifies the next sequence number the sender of this segment is expecting to receive.
- **syn** and **fin** and **rst**
 - **syn**—The synchronize flag is used to establish connections.
 - **fin**—The finish flag is used to clear connections.
 - **rst**—See above
- **psh**—The push flag indicates the data in the call should be immediately pushed through to the receiving user.
- **urg**—The urgent flag indicates that the urgent field is meaningful and must be added to the segment sequence number



Note Match-all is not supported. Match-any is supported only when used in the following combinations of positive flags: "rst and ack" (must be combined), "syn and fin and rst" (must be combined), "psh" and "urg".

ACL processing can impact the CPU in two ways:

- For some packets, when the hardware runs out of resources, the software must perform the ACL matches:
 - The TCP flag combinations rst ack, syn fin rst, urg and psh are processed in hardware. Other TCP flag combinations are supported in software.
 - If the total number of Layer 4 operations in an ACL is less than six, you can distribute the operations in any way you choose.

To create an ACL (IPv4 or IPv6) to filter TCP tags, perform the following task:

	Command	Purpose
Step 1	configure terminal Example Switch# configure terminal	Blocks all traffic to or from the configured unicast MAC address in the specified VLAN. To clear MAC address-based blocking, use the no form of this command without the drop keyword.
Step 2	ip access-list extended <i>access-list-name</i> Example Switch(config)# ip access-list extended kmd1 Switch(config-ext-nacl)#	Specifies the IP access list by name and enters named access list configuration mode.
Step 3	[sequence-number] permit tcp <i>source source-wildcard</i> [<i>operator</i> [<i>port</i>]] <i>destination destination-wildcard</i> [<i>operator</i> [<i>port</i>]] [<i>established</i> { match-any match-all } {+ -} [<i>flag-name</i>] [<i>precedence precedence</i>] [<i>tos tos</i>] [log] [<i>time-range time-range-name</i>] [<i>fragments</i>] Example (IPv4-specific) Switch(config-ext-nacl)# permit tcp host 10.1.1.1 host 2.2.2.2 established Example (IPv6-specific) Switch(config-ext-nacl)# permit tcp host 2001:2:25:1::1 host 2001:2:25:1::10 established	Specifies a permit statement in named IP access list mode. This access list happens to use a permit statement first, but a deny statement could appear first, depending on the order of statements you need. Use the TCP command syntax of the permit command. Match-all is not supported. Match-any is supported only when used in the following combinations of positive flags: "rst and ack" (must be combined), "sync and fin and rst" (must be combined), "psh" and "urg"
Step 4	[sequence-number] deny tcp <i>source source-wildcard</i> [<i>operator</i> [<i>port</i>]] <i>destination destination-wildcard</i> [<i>operator</i> [<i>port</i>]] [<i>established</i> { match-any match-all } {+ -} [<i>flag-name</i>] [<i>precedence precedence</i>] [<i>tos tos</i>] [log] [<i>time-range time-range-name</i>] [<i>fragments</i>] Example (IPv4-specific) Switch(config-ext-nacl)# deny tcp host 3.3.3.3 host 4.4.4.4 fin rst syn Example (IPv6-specific) Switch(config-ext-nacl)# deny tcp host 2001:2:25:1::2 host 2001:2:25:1::20 fin rst syn	(Optional) Specifies a deny statement in named IP access list mode. This access list happens to use a permit statement first, but a deny statement could appear first, depending on the order of statements you need. Use the TCP command syntax of the deny command. See the deny (IP) command for additional command syntax to permit upper-layer protocols (ICMP, IGMP, TCP, and UDP). Match-all is not supported. Match-any is supported only when used in the following combinations of positive flags: "rst and ack" (must be combined), "sync and fin and rst" (must be combined), "psh" and "urg".
Step 5	Repeat Step 3 or Step 4 as necessary, adding statements by sequence number where you planned. Use the no sequence-number command to delete an entry.	Allows you to revise the access list.

	Command	Purpose
Step 6	end Example Switch(config-ext-nacl)# end	(Optional) Exits the configuration mode and returns to privileged EXEC mode.
Step 7	show ip access-lists <i>access-list-name</i> Example Switch# show ip access-lists kmd1	(Optional) Displays the contents of the IP access list. Review the output to confirm that the access list includes the new entry.

Examples

The following access lists are processed completely in hardware:

```
access-list 104 permit tcp any any established
access-list 105 permit tcp any any rst ack
access-list 107 permit tcp any syn fin rst
```

Access lists 104 and 105 are identical; established is shorthand for rst and ack.

Access list 101, is processed completely in software:

```
access-list 101 permit tcp any any syn
```

Because four source and two destination operations exist, access list 106 is processed in hardware:

```
access-list 106 permit tcp any range 100 120 any range 120 140
access-list 106 permit tcp any range 140 160 any range 180 200
access-list 106 permit tcp any range 200 220
access-list 106 deny tcp any range 220 240
```

In the following code, the Layer 4 operations for the third ACE trigger an attempt to translate dst lt 1023 into multiple ACEs in hardware, because three source and three destination operations exist. If the translation attempt fails, the third ACE is processed in software.

```
access-list 102 permit tcp any lt 80 any gt 100
access-list 102 permit tcp any range 100 120 any range 120 1024
access-list 102 permit tcp any gt 1024 any lt 1023
```

Similarly, for access list 103, the third ACE triggers an attempt to translate dst gt 1023 into multiple ACEs in hardware. If the attempt fails, the third ACE is processed in software. Although the operations for source and destination ports look similar, they are considered different Layer 4 operations.

```
access-list 103 permit tcp any lt 80 any lt 80
access-list 103 permit tcp any range 100 120 any range 100 120
access-list 103 permit tcp any gt 1024 any gt 1023
```



Note Remember that source port lt 80 and destination port lt 80 are considered different operations.

- Some packets must be sent to the CPU for accounting purposes, but the action is still performed by the hardware. For example, if a packet must be logged, a copy is sent to the CPU for logging, but the forwarding (or dropping) is performed in the hardware. Although logging slows the CPU, it does not affect the forwarding rate. This sequence of events would happen under the following conditions:
 - When a log keyword is used
 - When an output ACL denies a packet
 - When an input ACL denies a packet, and on the interface where the ACL is applied, **ip unreachable** is enabled (**ip unreachable** is enabled by default on all the interfaces)

Configuring Unicast MAC Address Filtering

To block all unicast traffic to or from a MAC address in a specified VLAN, perform this task:

Command	Purpose
Switch(config)# mac-address-table static <i>mac_address</i> vlan <i>vlan_ID</i> drop	Blocks all traffic to or from the configured unicast MAC address in the specified VLAN. To clear MAC address-based blocking, use the no form of this command without the drop keyword.

This example shows how to block all unicast traffic to or from MAC address 0050.3e8d.6400 in VLAN 12:

```
Switch# configure terminal
Switch(config)# mac-address-table static 0050.3e8d.6400 vlan 12 drop
```

Configuring Named MAC Extended ACLs

You can filter non-IPv4, non-IPv6 traffic on a VLAN and on a physical Layer 2 port by using MAC addresses and named MAC extended ACLs. The procedure is similar to that of configuring other extended named ACLs. You can use a number to name the access list, but MAC access list numbers from 700 to 799 are not supported.



Note

Named MAC extended ACLs cannot be applied to Layer 3 interfaces.

For more information about the supported non-IP protocols in the **mac access-list extended** command, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference*.

To create a named MAC extended ACL, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# [no] mac access-list extended name	Defines an extended MAC access list using a name. To delete the entire ACL, use the no mac access-list extended name global configuration command. You can also delete individual ACEs from named MAC extended ACLs.
Step 3	Switch(config-ext-macl)# {deny permit} {any host source MAC address source MAC address mask} {any host destination MAC address destination MAC address mask} [protocol-family {appletalk arp-non-ipv4 decnet ipx ipv6 (not supported on Sup 6-E and 6L-E) rarp-ipv4 rarp-non-ipv4 vines xns}]	In extended MAC access-list configuration mode, specify to permit or deny any source MAC address, a source MAC address with a mask, or a specific host source MAC address and any destination MAC address, destination MAC address with a mask, or a specific destination MAC address. Note IPv6 packets do <i>not</i> generate Layer 2 ACL lookup keys.
Step 4	Switch(config-ext-macl)# end	Returns to privileged EXEC mode.
Step 5	Switch# show access-lists [number name]	Shows the access list configuration.
Step 6	Switch(config)# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to create and display an access list named `mac1`, denying only EtherType DECnet Phase IV traffic, but permitting all other types of traffic:

```
Switch(config)# mac access-list extended mac1
Switch(config-ext-macl)# deny any any decnet-iv (old) protocol-family decnet (new)
Switch(config-ext-macl)# permit any any
Switch(config-ext-macl)# end
Switch # show access-lists
Extended MAC access list mac1
    deny any any decnet-iv (old) protocol-family decnet (new)
    permit any any
```

The following example shows how to enable or disable hardware statistics while configuring ACEs in the access list:

```
Switch# config t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# mac access-list extended mac1
Switch(config-ext-nacl)# hardware statistics
Switch(config-ext-nacl)# end
Switch# show access-lists
Extended MAC access list mac1
    deny any any decnet-iv (old) protocol-family decnet (new)
    permit any any
    hardware statistics
```

Configuring EtherType Matching

You can classify non-IP traffic based on the EtherType value using the existing MAC access list commands. When you classify non-IP traffic by EtherType, you can apply security ACLs and QoS policies to traffic that carry the same EtherType.

EtherType matching allows you to classify tagged and untagged IP packets based on the EtherType value. Tagged packets present a potential operation problem:

- While single-tagged packets are supported on the access and trunk ports, double-tagged packets are not.
- Single and double-tagged packets are not supported if the port mode is dot1qtunnel.

For more information about the **mac access-list extended** command, refer to the *Catalyst 4500 Series Switch Cisco IOS Command Reference*.

To create a named MAC extended ACL, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# [no] mac access-list extended name	Defines an extended MAC access list using a name. To delete the entire ACL, use the no mac access-list extended name global configuration command. You can also delete individual ACEs from named MAC extended ACLs.
Step 3	Switch(config-ext-macl)# {deny permit} {any host source MAC address source MAC address mask} {any host destination MAC address destination MAC address mask} [protocol-family {appletalk arp-non-ipv4 decnet ipx ipv6 (not supported on Sup 6-E and 6L-E) rarp-ipv4 rarp-non-ipv4 vines xns} ethertype]	In extended MAC access-list configuration mode, specify to permit or deny any based upon the EtherTypes value, valid values are 15636-65535. Note You can specify matching by either EtherType or protocol family but not both.
Step 4	Switch(config-ext-macl)# end	Returns to privileged EXEC mode.
Step 5	Switch# show access-lists [number name]	Shows the access list configuration.
Step 6	Switch(config)# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to create and display an access list named matching, permitting the 0x8863 and 0x8040 EtherType values:

```
Switch(config)# mac access-list extended matching
Switch(config-ext-macl)# permit any any 0x8863
Switch(config-ext-macl)# permit any any 0x8040
Switch(config-ext-macl)# end
Switch # show access-lists matching
Extended MAC access list matching
    permit any any 0x8863
    permit any any netbios
Switch #
```

Configuring Named IPv6 ACLs

Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, 7-E, 6L-E, and 6-E support hardware-based IPv6 ACLs to filter unicast, multicast and broadcast IPv6 traffic on Layer 2 and Layer 3 interfaces. You can only configure such access lists on Layer 3 interfaces that are configured with an IPv6 address.

Beginning with IOS XE 3.7.0, you can employ IPv6 wildcard masking when specifying the Layer 3 address of a IPv6 ACL entry. Scale and performance issues that might be introduced by this feature are captured in the following:

<http://www.cisco.com/c/en/us/products/switches/catalyst-4500-series-switches/datasheet-listing.html>

The following document covers all security related hardware TCAM resources: “Cisco Catalyst 4500E Supervisor Engine 8-E: Wired and Wireless Convergence Data Sheet”

http://www.cisco.com/c/en/us/products/collateral/switches/catalyst-4500-series-switches/data_sheet_c78-728191.html



Note

routing-type/mobility-type extension header options in an IPv6 ACL have never been supported, but were previously configurable. As of Release IOS XE 3.4.0SG and IOS 15.1(2)SG, configuration of these options has been removed.

To create a named IPv6 ACLs, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# ipv6 access-list name	Defines an IPv6 access list using a name and enters access-list configuration mode. To delete the IPv6 ACL, use the no form of the command. You can also delete individual ACEs from the IPv6 access list.
Step 3	Switch(config-ipv6-acl)# { deny permit } { any proto object-group service-object-group-name } { host ipv6-addr ipv6-prefix ipv6-addr ipv6-wildcard-bits object-group source-network-object-group-name } { host ipv6-addr ipv6-prefix ipv6-addr ipv6-wildcard-bits object-group dest-network-object-group-name }	Specifies one or more IPv6 ACEs. Repeat this step to define multiple ACEs. You can specify an IPv6 address or use a wildcard mask to specify addresses. If you use a wildcard, the address should be specified in hexadecimal format, using 16-bit values between colons. (See RFC 2373).
		 Note 0 is the care bit and must match; 1 is the <i>don't care</i> bit and the system does not care whether it matches.
Step 4	Switch(config-ipv6-acl)# hardware statistics	(Optional) Enables hardware statistics for the IPv6 ACL.
Step 5	Switch(config-ipv6-acl)# end	Returns to privileged EXEC mode.
Step 6	Switch# show ipv6 access-list	Display the IPv6 access list configuration.

The following example shows how to create and display an IPv6 access list named v6test, denying only one IPv6 traffic with one particular source and destination address, but permitting all other types of IPv6 traffic:

```
Switch(config)# ipv6 access-list v6test
Switch(config-ipv6-acl)# deny ipv6 host 2020::10 host 2040::10
Switch(config-ipv6-acl)# permit any any
Switch(config-ipv6-acl)# end
Switch# show ipv6 access-list
IPv6 access list v6test
  deny ipv6 host 2020::10 host 2040::10 sequence 10
  permit ipv6 any any sequence 20
```

The following example show various ways of configuring ACEs in IPv6 ACL:

```
Switch(config)#ipv6 access-list v6test
```

The permit entry specifies the source and destination IPv6 addresses using wildcard masks:
 Switch(config-ipv6-acl)#permit 1:2::3 FF:0:FFFF:AA:20:: 4:5::6 0:FFFF:2233::FFFF

Here the permit entry allows all packets that have a source UDP port, and specifies the permit conditions for a destination IPv6 addresses using prefix/ prefix-length:

```
Switch(config-ipv6-acl)#permit udp any 3:8::5/64
```

Here the permit entry allows all packets that have a source TCP port and the IPv6 addresses (that has been specified using a wildcard mask), and allows destination addresses that have IPv6 prefix ::/0.

```
Switch(config-ipv6-acl)#permit tcp 1:2::3 FFFF:FFFF:: any
```

Here the permit entry allows all packets (source and destination) that have IPv6 prefix ::/0. This is necessary because an implicit deny -all condition is at the end of each IPv6 access list.

```
Switch(config-ipv6-acl)#permit any any
```

To enable hardware statistics, enter the following commands while configuring ACEs in the access list:

```
Switch(config)# ipv6 access-list v6test
Switch(config-ipv6-acl)# hardware statistics
Switch(config-ipv6-acl)# end
```



Note

Hardware statistics is disabled by default.

Applying IPv6 ACLs to Layer 2 and 3 Interface

To apply an IPv6 ACL to a Layer 3 interface, perform the following task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-type slot/interface</i>	Specifies the interface to be configured. Note <i>interface-type</i> must be a Layer 3 interface.
Step 3	Switch(config-if)# ipv6 traffic-filter <i>ipv6-acl {in out}</i>	Applies the IPv6 ACL to a Layer 3 interface.



Note

IPv6 ACLs are supported on Layer 3 interfaces and on Layer 2 ports using the **ipv6 traffic-filter** command.

The following example applies the extended-named IPv6 ACL simple-ipv6-acl to SVI 300 routed ingress traffic:

```
Switch# configure terminal
Switch(config)# interface vlan 300
Switch(config-if)# ipv6 traffic-filter simple-ipv6-acl in
```

**Note**

Output IPv6 ACLs with ACE to match on the ICMP option fail on a switch.

The following conditions may cause a RACL to malfunction (no workaround):

- ACLs are applied on the output direction of the interface.
- IPv6 ACL contain Ace to match on the ICMP option fields (ICMP Type or ICMP Code).

The following examples of nonfunctioning RACLs:

```
IPv6 access list a1
  permit icmp any any nd-ns sequence 10
  deny ipv6 any any sequence 20

IPv6 access list a2
  permit icmp 2020::/96 any nd-ns sequence 10
  deny ipv6 any any sequence 20
```

Configuring VLAN Maps

This section includes these topics:

- [VLAN Map Configuration Guidelines, page 62-22](#)
- [Creating and Deleting VLAN Maps, page 62-22](#)
- [Applying a VLAN Map to a VLAN, page 62-25](#)
- [Using VLAN Maps in Your Network, page 62-25](#)

This section describes how to configure VLAN maps, which is the only way to control filtering within a VLAN. VLAN maps have no direction. To filter traffic in a specific direction by using a VLAN map, you need to include an ACL with specific source or destination addresses. If there is a match clause for that type of packet (IP or MAC) in the VLAN map, the default action is to drop the packet if the packet does not match any of the entries within the map. If there is no match clause for that type of packet, the default is to forward the packet.

To create a VLAN map and apply it to one or more VLANs, follow these steps:

Step 1 Create the standard or extended IP ACLs or named MAC extended ACLs that you want to apply to the VLAN.

Step 2 Enter the **vlan access-map** global configuration command to create a VLAN ACL map entry.

In access map configuration mode, you have the option to enter an **action** (**forward** [the default] or **drop**) and enter the **match** command to specify an IP packet or a non-IP packet and to match the packet against one or more ACLs (standard or extended). If a match clause is not specified, the action is applied to all packets. The match clause can be used to match against multiple ACLs. If a packet matches any of the specified ACLs, the action is applied.

**Note**

If the VLAN map has a match clause for the type of packet (IP or MAC) and the packet does not match the type, the default is to drop the packet. If there is no match clause in the VLAN map for that type of packet, and no action specified, the packet is forwarded.

Step 3 Use the **vlan filter** global configuration command to apply a VLAN map to one or more VLANs.

**Note**

You cannot apply a VLAN map to a VLAN on a switch that has ACLs applied to Layer 2 interfaces (port ACLs).

VLAN Map Configuration Guidelines

When configuring VLAN maps, consider these guidelines:

- VLAN maps do not filter IPv4 ARP packets.
- If there is no router ACL configured to deny traffic on a routed VLAN interface (input or output), and no VLAN map configured, all traffic is permitted.
- Each VLAN map consists of a series of entries. The order of entries in a VLAN map is important. A packet that comes into the switch is tested against the first entry in the VLAN map. If it matches, the action specified for that part of the VLAN map is taken. If there is no match, the packet is tested against the next entry in the map.
- If the VLAN map has at least one match clause for the type of packet (IP or MAC) and the packet does not match any of these match clauses, the default is to drop the packet. If there is no match clause for that type of packet in the VLAN map, the default is to forward the packet.
- The system might take longer to boot if you have configured a very large number of ACLs.

Creating and Deleting VLAN Maps

Each VLAN map consists of an ordered series of entries. To create, add to, or delete a VLAN map entry, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# vlan access-map <i>name</i> [<i>number</i>]	Creates a VLAN map, and give it a name and (optionally) a number. The number is the sequence number of the entry within the map. When you create VLAN maps with the same name, numbers are assigned sequentially in increments of 10. When modifying or deleting maps, you can enter the number of the map entry that you want to modify or delete. This command enables access-map configuration mode.
Step 3	Switch(config-access-map)# action { drop forward }	(Optional) Sets the action for the map entry. The default is to forward.

	Command	Purpose
Step 4	Switch(config-access-map)# match { ip ipv6 mac } address { <i>name</i> <i>number</i> } [<i>name</i> <i>number</i>]	Matches the packet (using either the IP, IPv6, or MAC address) against one or more standard or extended access lists. Note that packets are matched only against access lists of the correct protocol type. IP packets are compared with standard or extended IP access lists. Non-IP packets are only compared with named MAC extended access lists. If a match clause is not specified, the action is taken on all packets.
Step 5	Switch(config-access-map)# end	Returns to global configuration mode.
Step 6	Switch(config)# show running-config	Displays the access list configuration.
Step 7	Switch(config)# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

You can use the **no vlan access-map** *name* global configuration command to delete a map. You can use the **no vlan access-map** *name number* global configuration command to delete a single sequence entry from within the map. You can use the **no action** access-map configuration command to enforce the default action, which is to forward.

VLAN maps do not use the specific **permit** or **deny** keywords. To deny a packet by using VLAN maps, create an ACL that would match the packet, and then set the action to drop. A permit in the ACL is the same as a match. A deny in the ACL means no match.

Examples of ACLs and VLAN Maps

These examples show how to create ACLs and VLAN maps for specific purposes.

Example 1

This example shows how to create an ACL and a VLAN map to deny a packet. In the first map, any packets that match the ip1 ACL (TCP packets) would be dropped. You first create the ip1 ACL to permit any TCP packet and no other packets. Because there is a match clause for IP packets in the VLAN map, the default action is to drop any IP packet that does not match any of the match clauses.

```
Switch(config)# ip access-list extended ip1
Switch(config-ext-nacl)# permit tcp any any
Switch(config-ext-nacl)# exit

Switch(config)# vlan access-map map_1 10
Switch(config-access-map)# match ip address ip1
Switch(config-access-map)# action drop
```

This example shows how to create a VLAN map to permit a packet. ACL ip2 permits UDP packets; and any packets that match the ip2 ACL are forwarded.

```
Switch(config)# ip access-list extended ip2
Switch(config-ext-nacl)# permit udp any any
Switch(config-ext-nacl)# exit
Switch(config)# vlan access-map map_1 20
Switch(config-access-map)# match ip address ip2
Switch(config-access-map)# action forward
```

In this map, any IP packets that did not match any of the previous ACLs (that is, packets that are not TCP packets or UDP packets) would get dropped.

Example 2

In this example, the VLAN map is configured to drop IP packets and to forward MAC packets by default. By applying standard ACL 101 and the extended named access lists **igmp-match** and **tcp-match**, the VLAN map is configured to do the following:

- Forward all UDP packets
- Drop all IGMP packets
- Forward all TCP packets
- Drop all other IP packets
- Forward all non-IP packets

```
Switch(config)# access-list 101 permit udp any any
Switch(config)# ip access-list extended igmp-match
Switch(config-ext-nacl)# permit igmp any any
Switch(config)# ip access-list extended tcp-match
Switch(config-ext-nacl)# permit tcp any any
Switch(config-ext-nacl)# exit
Switch(config)# vlan access-map drop-ip-default 10
Switch(config-access-map)# match ip address 101
Switch(config-access-map)# action forward
Switch(config-access-map)# exit
Switch(config)# vlan access-map drop-ip-default 20
Switch(config-access-map)# match ip address igmp-match
Switch(config-access-map)# action drop
Switch(config-access-map)# exit
Switch(config)# vlan access-map drop-ip-default 30
Switch(config-access-map)# match ip address tcp-match
Switch(config-access-map)# action forward
```

Example 3

In this example, the VLAN map is configured to drop MAC packets and forward IP packets by default. By applying MAC extended access lists, **good-hosts** and **good-protocols**, the VLAN map is configured to do the following:

- Forward MAC packets from hosts 0000.0c00.0111 and 0000.0c00.0211
- Forward MAC packets of DECnet or VINES (Virtual Integrated Network Service) protocol-family
- Drop all other non-IP packets
- Forward all IP packets

```
Switch(config)# mac access-list extended good-hosts
Switch(config-ext-nacl)# permit host 000.0c00.0111 any
Switch(config-ext-nacl)# permit host 000.0c00.0211 any
Switch(config-ext-nacl)# exit
Switch(config)# mac access-list extended good-protocols
Switch(config-ext-nacl)# permit any any protocol-family decnet
Switch(config-ext-nacl)# permit any any protocol-family vines
Switch(config-ext-nacl)# exit
Switch(config)# vlan access-map drop-mac-default 10
Switch(config-access-map)# match mac address good-hosts
Switch(config-access-map)# action forward
Switch(config-access-map)# exit
Switch(config)# vlan access-map drop-mac-default 20
Switch(config-access-map)# match mac address good-protocols
Switch(config-access-map)# action forward
```

Example 4

In this example, the VLAN map is configured to drop all packets (IP and non-IP). By applying access lists **tcp-match** and **good-hosts**, the VLAN map is configured to do the following:

- Forward all TCP packets
- Forward MAC packets from hosts 0000.0c00.0111 and 0000.0c00.0211
- Drop all other IP packets
- Drop all other MAC packets

```
Switch(config)# vlan access-map drop-all-default 10
Switch(config-access-map)# match ip address tcp-match
Switch(config-access-map)# action forward
Switch(config-access-map)# exit
Switch(config)# vlan access-map drop-all-default 20
Switch(config-access-map)# match mac address good-hosts
Switch(config-access-map)# action forward
```

Applying a VLAN Map to a VLAN

To apply a VLAN map to one or more VLANs, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# vlan filter <i>mapname</i> vlan-list <i>list</i>	Applies the VLAN map to one or more VLAN IDs. The list can be a single VLAN ID (22), a consecutive list (10-22), or a string of VLAN IDs (12, 22, 30). Spaces around comma, and dash, are optional.
Step 3	Switch(config)# show running-config	Displays the access list configuration.
Step 4	Switch(config)# copy running-config startup-config	(Optional) Saves your entries in the configuration file.



Note

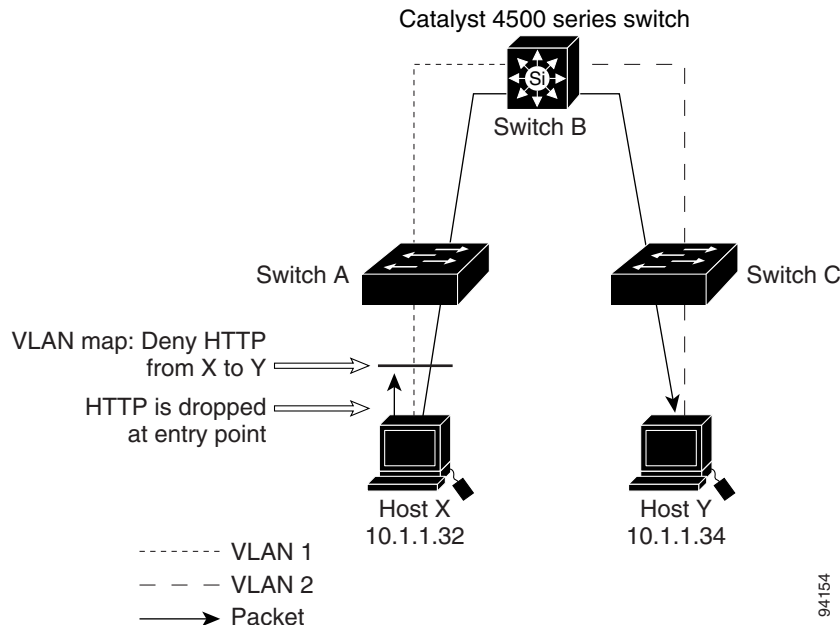
You cannot apply a VLAN map to a VLAN on a switch that has ACLs applied to Layer 2 interfaces (port ACLs).

This example shows how to apply VLAN map 1 to VLANs 20 through 22:

```
Switch(config)# vlan filter map 1 vlan-list 20-22
```

Using VLAN Maps in Your Network

Figure 62-3 shows a typical wiring closet configuration. Host X and Host Y are in different VLANs, connected to wiring closet switches A and C. Traffic moving from Host X to Host Y is routed by Switch B. Access to traffic moving from Host X to Host Y can be controlled at the entry point of Switch A. In the following configuration, the switch can support a VLAN map and a QoS classification ACL.

Figure 62-3 Wiring Closet Configuration

For example, if you do not want HTTP traffic to be switched from Host X to Host Y, you could apply a VLAN map on Switch A to drop all HTTP traffic moving from Host X (IP address 10.1.1.32) to Host Y (IP address 10.1.1.34) at Switch A and not bridge the traffic to Switch B. To configure this scenario, you would do the following.

First, define an IP access list HTTP to permit (match) any TCP traffic on the HTTP port, as follows:

```
Switch(config)# ip access-list extended http
Switch(config-ext-nacl)# permit tcp host 10.1.1.32 host 10.1.1.34 eq www
Switch(config-ext-nacl)# exit
```

Next, create a VLAN access map named map2 so that traffic that matches the HTTP access list is dropped and all other IP traffic is forwarded, as follows:

```
Switch(config)# vlan access-map map2 10
Switch(config-access-map)# match ip address http
Switch(config-access-map)# action drop
Switch(config-access-map)# exit

Switch(config)# ip access-list extended match_all
Switch(config-ext-nacl)# permit ip any any
Switch(config-ext-nacl)# exit
Switch(config)# vlan access-map map2 20
Switch(config-access-map)# match ip address match_all
Switch(config-access-map)# action forward
```

You then apply the VLAN access map named map2 to VLAN 1, as follows:

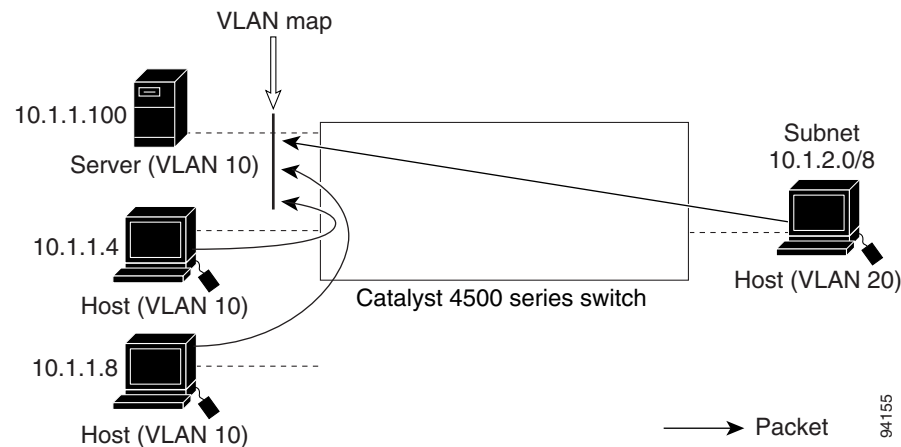
```
Switch(config)# vlan filter map2 vlan 1
```

Denying Access to a Server on Another VLAN

Figure 62-4 shows how to restrict access to a server on another VLAN. In this example, server 10.1.1.100 in VLAN 10 has the following access restrictions:

- Hosts in subnet 10.1.2.0/8 in VLAN 20 should not have access.
- Hosts 10.1.1.4 and 10.1.1.8 in VLAN 10 should not have access.

Figure 62-4 Deny Access to a Server on Another VLAN



This procedure configures ACLs with VLAN maps to deny access to a server on another VLAN. The VLAN map SERVER1_ACL denies access to hosts in subnet 10.1.2.0/8, host 10.1.1.4, and host 10.1.1.8. Then it permits all other IP traffic. In Step 3, VLAN map SERVER1 is applied to VLAN 10. To configure this scenario, follow these steps:

Step 1 Define the IP ACL to match and permit the correct packets.

```
Switch(config)# ip access-list extended SERVER1_ACL
Switch(config-ext-nacl)# permit ip 10.1.2.0 0.0.0.255 host 10.1.1.100
Switch(config-ext-nacl)# permit ip host 10.1.1.4 host 10.1.1.100
Switch(config-ext-nacl)# permit ip host 10.1.1.8 host 10.1.1.100
Switch(config-ext-nacl)# exit
```

Step 2 Define a VLAN map using the ACL to drop IP packets that match SERVER1_ACL and forward IP packets that do not match the ACL.

```
Switch(config)# vlan access-map SERVER1_MAP
Switch(config-access-map)# match ip address SERVER1_ACL
Switch(config-access-map)# action drop
Switch(config)# vlan access-map SERVER1_MAP 20
Switch(config-access-map)# action forward
Switch(config-access-map)# exit
```

Step 3 Apply the VLAN map to VLAN 10.

```
Switch(config)# vlan filter SERVER1_MAP vlan-list 10.
```

Displaying VLAN Access Map Information

To display information about VLAN access maps or VLAN filters, perform one of these commands:

Command	Purpose
Switch# show vlan access-map [<i>mapname</i>]	Shows information about all VLAN access maps or the specified access map.
Switch# show vlan filter [access-map <i>name</i> / vlan <i>vlan-id</i>]	Shows information about all VLAN filters or about a specified VLAN or VLAN access map.

it is a sample output of the **show vlan access-map** command:

```
Switch# show vlan access-map
Vlan access-map "map_1" 10
  Match clauses:
    ip address: ip1
  Action:
    drop
Vlan access-map "map_1" 20
  Match clauses:
    mac address: mac1
  Action:
    forward
Vlan access-map "map_1" 30
  Match clauses:
  Action:
    drop
```



Note

Sequence 30 does not have a match clause. All packets (IP as well as non-IP) are matched against it and dropped.

it is a sample output of the **show vlan filter** command:

```
Switch# show vlan filter
VLAN Map map_1 is filtering VLANs:
  20-22
```

Using VLAN Maps with Router ACLs

If the VLAN map has a match clause for a packet type (IP or MAC) and the packet does not match the type, the default is to drop the packet. If there is no match clause in the VLAN map, and no action is specified, the packet is forwarded if it does not match any VLAN map entry.



Note

You cannot combine VLAN maps or input router ACLs with port ACLs on a switch.

Topics include:

- [Guidelines for Using Router ACLs and VLAN Maps on the Same VLAN, page 62-29](#)
- [Examples of Router ACLs and VLAN Maps Applied to VLANs, page 62-29](#)

Guidelines for Using Router ACLs and VLAN Maps on the Same VLAN

Because the switch hardware performs one lookup for each direction (input and output), you must merge a router ACL and a VLAN map when they are configured on the same VLAN. Merging the router ACL with the VLAN map can significantly increase the number of ACEs.

When possible, try to write the ACL so that all entries have a single action except for the final, default action. You should write the ACL using one of these two forms:

```
permit...
permit...
permit...
deny ip any any
```

or

```
deny...
deny...
deny...
permit ip any any
```

To define multiple permit or deny actions in an ACL, group each action type together to reduce the number of entries.

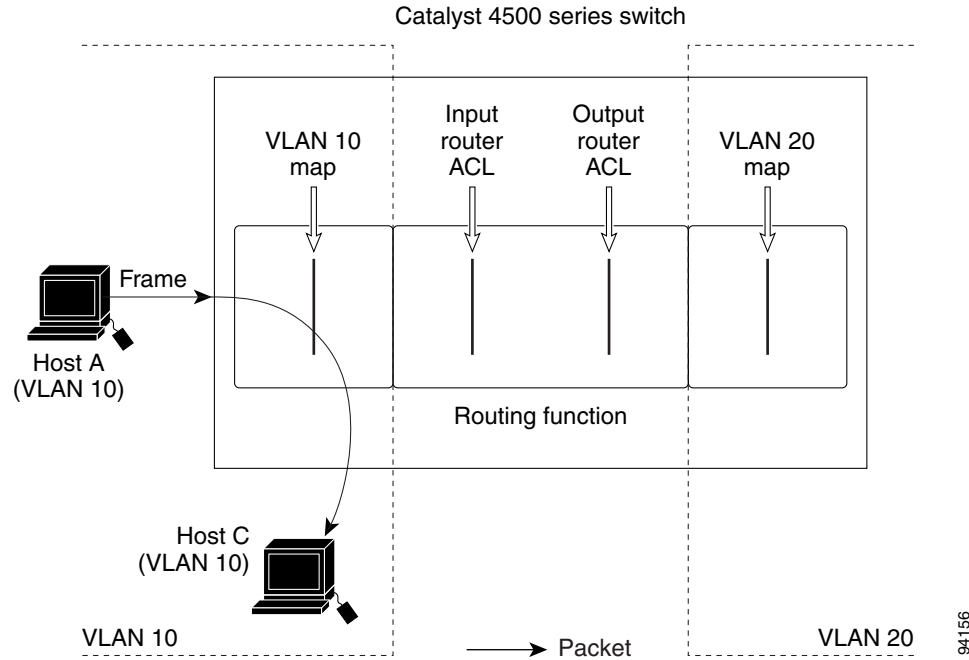
If you need to specify the full-flow mode and the ACL contains both IP ACEs and TCP/UDP/ICMP ACEs with Layer 4 information, put the Layer 4 ACEs at the end of the list. Doing this gives priority to the filtering of traffic based on IP addresses.

Examples of Router ACLs and VLAN Maps Applied to VLANs

These examples show how router ACLs and VLAN maps are applied on a VLAN to control the access of switched, bridged, routed, and multicast packets. Although the following illustrations show packets being forwarded to their destination, each time a packet crosses a line indicating a VLAN map or an ACL, the packet could be dropped rather than forwarded.

ACLs and Switched Packets

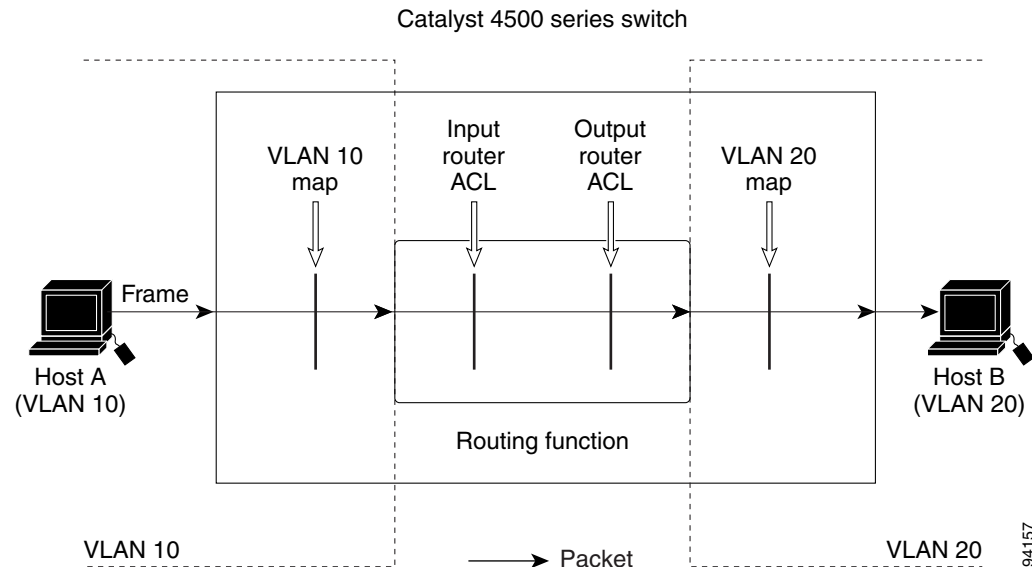
Figure 62-5 shows how an ACL processes packets that are switched within a VLAN. Packets switched within the VLAN are not processed by router ACLs.

Figure 62-5 Applying ACLs on Switched Packets

ACLs and Routed Packets

Figure 62-6 shows how ACLs are applied on routed packets. For routed packets, the ACLs are applied in this order:

1. VLAN map for input VLAN
2. Input router ACL
3. Output router ACL
4. VLAN map for output VLAN

Figure 62-6 Applying ACLs on Routed Packets

Configuring PACLS

This section describes how to configure PACLS, which are used to control filtering on Layer 2 interfaces. PACLS can filter traffic to or from Layer 2 interfaces based on Layer 3 information, Layer 4 head information or non-IP Layer 2 information.

This section includes these topics:

- [Creating a PACL, page 62-31](#)
- [PACL Configuration Guidelines, page 62-32](#)
- [Removing the Requirement for a Port ACL, page 62-32](#)
- [Webauth Fallback, page 62-33](#)
- [Configuring IPv4, IPv6, and MAC ACLs on a Layer 2 Interface, page 62-33](#)
- [Using PACL with Access-Group Mode, page 62-34](#)
- [Configuring Access-group Mode on Layer 2 Interface, page 62-35](#)
- [Applying ACLs to a Layer 2 Interface, page 62-35](#)
- [Displaying an ACL Configuration on a Layer 2 Interface, page 62-36](#)

Creating a PACL

To create a PACL and apply it to one or more interfaces, follow these steps:

- Step 1** Create the standard or extended IPv4 ACLs, IPv6 ACLs, or named MAC extended ACLs that you want to apply to the interface.

- Step 2** Use the `IP access-group`, `IPv6 traffic-filter`, or `mac access-group interface` command to apply IPv4, IPv6, or MAC ACLs to one or more Layer 2 interfaces.
-

PACL Configuration Guidelines

When configuring PACLS, consider these guidelines:

- There can be at most one IPv4, one IPv6, and one MAC access list applied to the same Layer 2 interface per direction.
- The IPv4 access list filters only IPv4 packets, the IPv6 access list filters only IPv6 packets, and the MAC access list filters only non-IP packets.
- The number of ACLs and ACEs that can be configured as part of a PACL are bounded by the hardware resources on the switch. Those hardware resources are shared by various ACL features (for example, RACL, VACL) that are configured on the system. If insufficient hardware resources to program PACL exist in hardware, the actions for input and output PACLS differ:
 - For input PACLS, some packets are sent to CPU for software forwarding.
 - For output PACLS, the PACL is disabled on the port.
- If insufficient hardware resources exist to program the PACL, the output PACL is not applied to the port, and you receive a warning message.
- The input ACL logging option is supported, although logging is not supported for output ACLs.
- The access group mode can change the way PACLS interact with other ACLs. To maintain consistent behavior across Cisco platforms, use the default access group mode.
- If a PACL is removed when there are active sessions on a port, a hole (permit ip any any) is installed on the port.

Removing the Requirement for a Port ACL

Prior to Cisco IOS Release 12.2(54)SG, a standard port ACL was necessary if you planned to download an ACL from a AAA server. This was because ACL infrastructure was insufficient to provide dynamic creation of access control entries without associating an ACL with the port.

Starting with Cisco IOS Release 12.2(54)SG, configuring a port ACL is not mandatory. If a port ACL is not configured on the port (by entering the `ip access-group number in` command), a default ACL (AUTH-DEFAULT-ACL) is attached automatically to the port when an ACL is downloaded. It allows only DHCP traffic and consists of the following ACEs:

```
permit udp any range bootps 65347 any range bootpc 65348
permit udp any any range bootps 65347
deny ip any any.
```

AUTH-DEFAULT-ACL is automatically created. To modify it, enter the following command:

```
ip access-list extended AUTH-DEFAULT-ACL
```

This ACL is not nvgened. AUTH-DEFAULT-ACL is attached provided there are sessions applying dynamic ACLs (Per-user/Filter-Id/DACL). AUTH-DEFAULT-ACL is removed when the last authenticated session with policies is cleared. It remains attached to the port provided at least one session is applying dynamic policies.

Configuration Restrictions

The following restrictions apply:

- Starting with Cisco IOS Release 12.2(54)SG, the port ACL does not require configuration; the default ACL is created automatically.
- Even if AUTH-DEFAULT-ACL is modified, it is not nvgened.

Debugging Considerations

Syslog messages appear when AUTH-DEFAULT-ACL is attached or detached from an interface provided you enter the **epm logging** command in configuration mode.

The following syslog displays when the default ACL is attached:

```
%EPM-6-AUTH_ACL: POLICY Auth-Default-ACL| EVENT CREATE-ATTACH-SUCCESS
```

The following syslog displays when the ACL is detached:

```
%EPM-6-AUTH_ACL: POLICY Auth-Default-ACL| EVENT DETACH-SUCCESS  
%EPM-6-AUTH_ACL: POLICY Auth-Default-ACL| EVENT DELETE-SUCCESS
```

Webauth Fallback

Many authentication methods require specific capabilities on the end-point device to respond to the network authenticating device with its identity or credentials. If the end-point lacks the required capability, the authenticator must fallback to alternative methods to gather host or user credentials. If the 802.1X/MAB authentication mechanism fails, a fallback to webauth might occur.

Prior to Cisco IOS Release 12.2(54)SG, webauth fallback implementation required a fallback profile configured on the authenticating device. As part of this profile, an admission rule must be configured along with the access policies (the fallback ACL).

Consider a situation where no port ACL is configured on a port. The first few hosts authenticated through 802.1X/MAB do not download any ACLs. All traffic from these hosts is allowed through. Now, suppose a host connects to the port, and there is a fallback to webauth to authenticate the host. The fallback ACL will be installed on the port, and traffic from previously authenticated hosts will also be restricted by this fallback ACL.

Starting with Cisco IOS Release 12.2(54)SG, Cisco uses a different approach to address this issue. When a host falls back to webauth for authentication, the ACE entries in the fallback ACL are converted into entries with Host IP insertion for a host that has fallen back and will be applied until the host authenticates. Once the host successfully authenticates, the fallback ACL is removed. The resultant host ACLs will be: dynamic ACLs and Port ACL/AUTH-DEFAULT-ACL. Refer to the previous section for an explanation of AUTH-DEFAULT -ACL.

Configuring IPv4, IPv6, and MAC ACLs on a Layer 2 Interface



Note

Only IPv4, IPv6 and MAC ACLs can be applied to Layer 2 physical interfaces.

Standard (numbered, named), Extended (numbered, named) IP ACLs, and Extended Named MAC ACLs are also supported.

To apply IPv4 or MAC ACLs on a Layer 2 interface, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface interface	Enters interface configuration mode.
Step 3	Switch(config-if)# [no] {ip mac} access-group {name number} {in out}	Applies numbered or named ACL to the Layer 2 interface. The no form deletes the IP or MAC ACL from the Layer 2 interface.
Step 4	Switch(config)# show running-config	Displays the access list configuration.

To apply IPv6 ACLs on a Layer 2 interface, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface interface	Enters interface configuration mode.
Step 3	Switch(config-if)# [no] ipv6 traffic-filter name {in out}	Applied the specified IPv6 ACL to the Layer 2 interface. The no form deletes the IPv6 ACL from the Layer 2 interface.
Step 4	Switch(config)# show running-config	Displays the access list configuration.

The following example shows how to configure the Extended Named IP ACL `simple-ip-acl` to permit all TCP traffic and implicitly deny all other IP traffic:

```
Switch(config)# interface Gi3/1
Switch(config-if)# ip access-list extended simple-ip-acl
Switch(config-ext-nacl)# permit tcp any any
Switch(config-ext-nacl)# end
```

The following example shows how to configure the Extended Named MACL `simple-mac-acl` to permit source host 000.000.011 to any destination host:

```
Switch(config)# interface Gi3/1
Switch(config-if)# mac access-list extended simple-mac-acl
Switch(config-ext-macl)# permit host 000.000.011 any
Switch(config-ext-macl)# end
```

Using PACL with Access-Group Mode

You can use the access group mode to change the way PACLS interact with other ACLs. For example, if a Layer 2 interface belongs to VLAN100, VACL (VLAN filter) V1 is applied on VLAN100, and PACL P1 is applied on the Layer 2 interface. In this situation, you must specify how P1 and V1 impact the traffic with the Layer 2 interface on VLAN100. In a per-interface method, you can use the **access-group mode** command to specify one of the following desired modes:

- **prefer port mode**—If PACL is configured on a Layer 2 interface, then PACL takes effect and overwrites the effect of other ACLs (Router ACL and VACL). If no PACL feature is configured on the Layer 2 interface, other features applicable to the interface are merged and applied on the interface. it is the default access group mode.

- prefer VLAN mode—VLAN-based ACL features take effect on the port if they have been applied on the port and no PACLs are in effect. If no VLAN-based ACL features are applicable to the Layer 2 interface, then the PACL feature already on the interface is applied.
- merge mode—Merges applicable ACL features before they are programmed into the hardware.

Configuring Access-group Mode on Layer 2 Interface

To configure an access mode on a Layer 2 interface, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface</i>	Enters interface configuration mode.
Step 3	Switch(config-if)# [no] access-group mode { prefer { port vlan } merge }	Applies numbered or named ACL to the Layer 2 interface. The no form deletes the IP or MAC ACL from the Layer 2 interface.
Step 4	Switch(config)# show running-config	Displays the access list configuration.

This example shows how to merge and apply features other than PACL on the interface:

```
Switch# configure terminal
Switch(config)# interface fast 6/1
Switch(config-if)# access-group mode prefer port
```

This example shows how to merge applicable ACL features before they are programmed into hardware:

```
Switch# configure terminal
Switch(config)# interface fast 6/1
Switch(config-if)# access-group mode merge
```

Applying ACLs to a Layer 2 Interface

To apply IPv4, IPv6, and MAC ACLs to a Layer 2 interface, perform one of these tasks:

Command	Purpose
Switch(config-if)# ip access-group <i>ip-acl</i> { in out }	Applies an IPv4 ACL to the Layer 2 interface.
Switch(config-if)# ipv6 traffic-filter <i>ipv6-acl</i> { in out }	Applies an IPv6 ACL to the Layer 2 interface.
Switch(config-if)# mac access-group <i>mac-acl</i> { in out }	Applies a MAC ACL to the Layer 2 interface.

This example applies the extended named IP ACL simple-ip-acl to interface FastEthernet 6/1 ingress traffic:

```
Switch# configure terminal
Switch(config)# interface fast 6/1
Switch(config-if)# ip access-group simple-ip-acl in
```

This example applies the IPv6 ACL simple-ipv6-acl to interface FastEthernet 6/1 ingress traffic:

```
Switch# configure terminal
Switch(config)# interface fast 6/1
Switch(config-if)# ipv6 traffic-filter simple-ipv6-acl in
```

This example applies the extended named MAC ACL `simple-mac-acl` to interface FastEthernet 6/1 egress traffic:

```
Switch# configure terminal
Switch(config)# interface fast 6/1
Switch(config-if)# mac access-group simple-mac-acl out
```

Displaying an ACL Configuration on a Layer 2 Interface

To display information about an ACL configuration on Layer 2 interfaces, perform one of these tasks:

Command	Purpose
Switch# show ip interface [<i>interface-name</i>]	Shows the IP access group configuration on the interface.
Switch# show mac access-group interface [<i>interface-name</i>]	Shows the MAC access group configuration on the interface.
Switch# show access-group mode interface [<i>interface-name</i>]	Shows the access group mode configuration on the interface.

This example shows that the IP access group `simple-ip-acl` is configured on the inbound direction of interface `fa6/1`:

```
Switch# show ip interface fast 6/1
FastEthernet6/1 is up, line protocol is up
  Inbound access list is simple-ip-acl
  Outgoing access list is not set
```

This example shows that MAC access group `simple-mac-acl` is configured on the inbound direction of interface `fa6/1`:

```
Switch# show mac access-group interface fast 6/1
Interface FastEthernet6/1:
  Inbound access-list is simple-mac-acl
  Outbound access-list is not set
```

This example shows that access group merge is configured on interface `fa6/1`:

```
Switch# show access-group mode interface fast 6/1
Interface FastEthernet6/1:
  Access group mode is: merge
```

Using PACL with VLAN Maps and Router ACLs

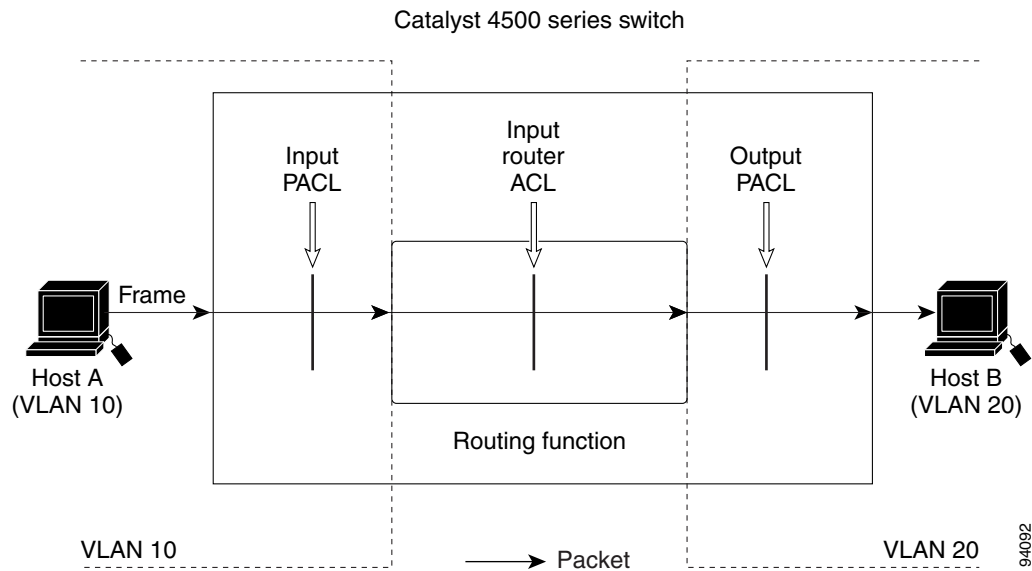
For PACLs, the interaction with Router ACLs and VACLs depends on the interface access group mode as shown in [Table 62-1](#).

Table 62-1 Interaction between PACLs, VACLs, and Router ACLs

ACL Type(s)	Input PACL		
	prefer port mode	prefer vlan mode	merge mode
1. Input Router ACL	PACL applied	Input Router ACL applied	PACL, Input Router ACL (merged) applied in order (ingress)
2. VACL	PACL applied	VACL applied	PACL, VACL (merged) applied in order (ingress)
3. VACL + Input Router ACL	PACL applied	VACL + Input Router ACL applied	PACL, VACL, Input Router ACL (merged) applied in order (ingress)

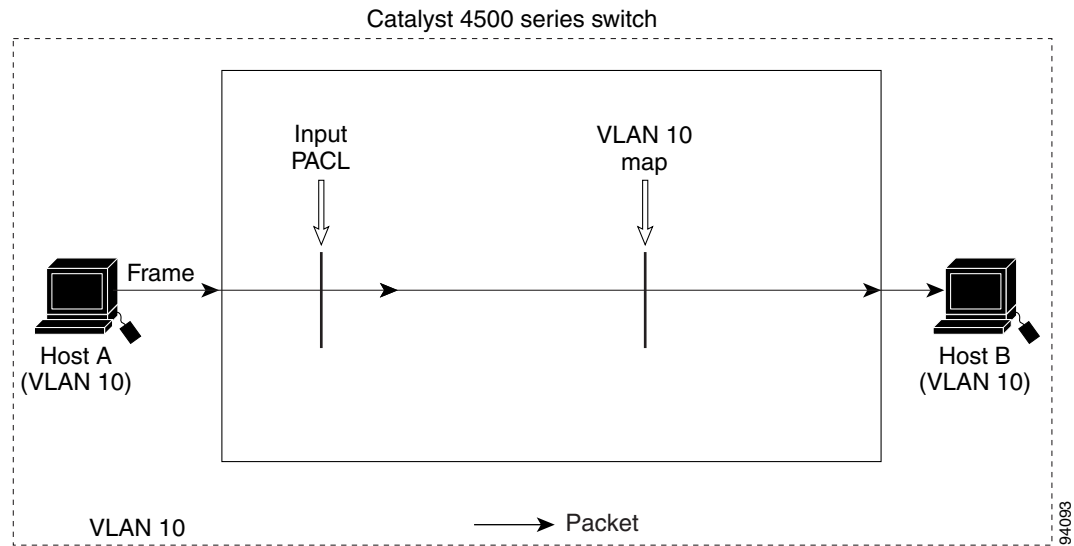
Each ACL type listed in [Table 62-1](#) corresponds with these scenarios:

Scenario 1: Host A is connected to an interface in VLAN 20, which has an SVI configured. The interface has input PACL configured, and the SVI has input Router ACL configured as shown in [Figure 62-7](#):

Figure 62-7 Scenario 1: PACL Interaction with an Input Router ACL

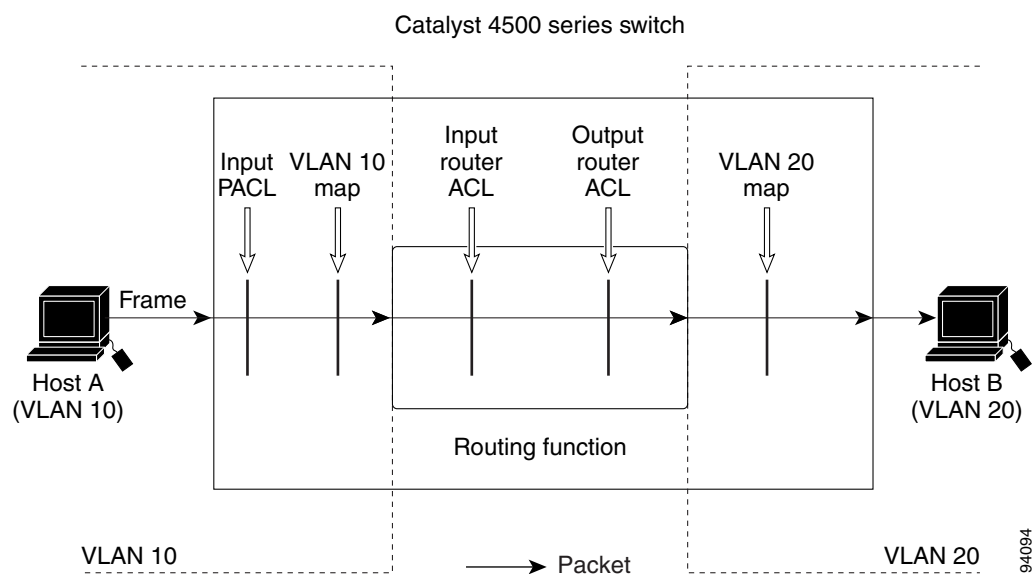
If the interface access group mode is prefer port, then only the input PACL is applied on the ingress traffic from Host A. If the mode is prefer VLAN, then only the input Router ACL is applied to ingress traffic from Host A that requires routing. If the mode is merge, then the input PACL is first applied to the ingress traffic from Host A, and the input Router ACL is applied on the traffic that requires routing.

Scenario 2: Host A is connected to an interface in VLAN 10, which has a VACL (VLAN Map) configured and an input PACL configured as shown in [Figure 62-8](#):

Figure 62-8 Scenario 2: PACL Interaction with a VACL

If the interface access group mode is prefer port, then only the input PACL is applied on the ingress traffic from Host A. If the mode is prefer VLAN, then only the VACL is applied to the ingress traffic from Host A. If the mode is merge, the input PACL is first applied to the ingress traffic from Host A, and the VACL is applied on the traffic.

Scenario 3: Host A is connected to an interface in VLAN 10, which has a VACL and an SVI configured. The SVI has an input Router ACL configured and the interface has an input PACL configured, as shown in Figure 62-9:

Figure 62-9 Scenario 3: VACL and Input Router ACL

If the interface access group mode is prefer port, then only the input PACL is applied on the ingress traffic from Host A. If the mode is prefer VLAN, then the merged results of the VACL and the input Router ACL are applied to the ingress traffic from Host A. If the mode is merge, the input PACL is first

applied to the ingress traffic from Host A, the VACL is applied on the traffic and finally, and the input Router ACL is applied to the traffic that needs routing. (that is, the merged results of the input PACL, VACL, and input Router ACL are applied to the traffic).

Configuring Object Group ACLs

Object groups provide an alternative way of dealing with ACLs.

Instead of allowing or disallowing individual IP addresses, protocols, and ports (which are used in conventional ACLs), you can use each ACE to allow or disallow an entire group of users to access a group of servers or services.

Object groups enable you to group ACE entries and add or remove entries while keeping your ACL structure more readable. Object group ACLs (OG ACLs) are especially suited to help you manage large ACLs that require frequent changing. Cisco IOS Firewall benefits from object groups, because they simplify policy creation (for example, group A has access to group A services).

Beginning with Cisco IOS XE Release 3.7.1E, object groups are supported for IPv4 ACLs (IPv4 OG ACLs), and with Cisco IOS XE Release 3.9.2E, for IPv6 ACLs (IPv6 OG ACLs).

The feature is supported only on Cisco Catalyst 4500E Series Switches with Supervisor Engine 9-E, 8-E, 7-LE, and 7-E, and Cisco Catalyst 4500-X Series Switches.

See the following sections for more information:

- [Overview, page 62-39](#)
- [Configuring IPv4 OG ACLs, page 62-40](#)
- [Configuring IPv6 OG ACLs, page 62-46](#)

Overview

All features that use or reference conventional ACLs are compatible with OG ACLs. This feature extends the conventional ACLs to support OG ACLs and also adds new keywords and the source and destination addresses and ports.

To configure OG ACLs, you first create one or more object groups. These can be any combination of network object groups or service object groups. You then create ACEs that apply a policy (such as permit or deny) to those object groups.

A network object group includes the following objects:

- Host IP addresses
- Network address of group members
- Nested object groups

A service object group includes the following objects:

- Source and destination protocol ports (such as Telnet or Simple Network Management Protocol [SNMP])
- Internet Control Message Protocol (ICMP) types (such as echo, echo-reply, or host-unreachable)
- Top-level protocols (such as Encapsulating Security Payload [ESP], TCP, or UDP)
- Other service object groups

You can configure an OG ACL multiple times with a source group only, a destination group only, or both source and destination groups.

You can add, delete, or change objects in an object group membership list dynamically (without deleting and redefining the object group), and without redefining the ACL ACE that uses the object group.

When you add a member to a group, delete a member from a group, or modify the policy statements in an ACE that uses an access group, the system updates the ACEs in the TCAM. An ACE that is defined using a group name, is equivalent to multiple ACEs (one applied to each entry in the object group). The system expands the object group ACL ACEs into multiple Cisco IOS ACEs (one ACE for each entry in the group) and populates the ACEs in the TCAM. Therefore, the object group ACL feature reduces the number of entries you need to configure but does not reduce TCAM usage.

You cannot delete an object group that is used within an ACL or a class-based policy language (CPL) policy.

Configuring IPv4 OG ACLs

- [Guidelines and Restrictions for Configuring IPv4 OG ACLs, page 62-40](#)
- [Creating a Network Object Group, page 62-40](#)
- [Creating a Service Object Group, page 62-42](#)
- [Configuring an IPv4 OG ACL, page 62-43](#)
- [Applying an IPv4 OG ACL to an Interface, page 62-44](#)
- [Verifying IPv4 OG ACLs, page 62-45](#)

Guidelines and Restrictions for Configuring IPv4 OG ACLs

- The object groups can be used only in extended named and numbered ACLs.
- IPv4 OG ACLs support only Layer 3 interfaces (such as routed interfaces and VLAN interfaces). They do not support Layer 2 features such as VLAN ACLs (VACLs) or port ACLs (PACLs).
- IPv4 OG ACLs are not supported with IPsec.
- IPv4 OG ACLs are not supported on management interfaces, such as FastEthernet1, and on GRE tunnels.
- The maximum number of object group-based ACEs supported in an ACL is 2048.
- IPv4 OG ACEs are used only while processing hardware-switched packets.

ACL statements using object groups are ignored on those packets that are sent to the Route Processor, and such ACL statements are not used for filtering. To match such packets, regular ACEs (without object groups) need to be created in the same ACL.

Creating a Network Object Group

To create a network object group, perform this task:

	Command or Action	Purpose
Step 1	configure terminal Example: Switch# configure terminal	Enters global configuration mode.
Step 2	object-group network <i>object-group-name</i> Example: Switch(config)# object-group network my-network-object-group	Defines the object group name and enters network object-group configuration mode.
Step 3	description <i>description-text</i> Example: Switch(config-network-group)# description test engineers	(Optional) Specifies a description of the object group. You can use up to 200 characters.
Step 4	host {<i>host-address</i> <i>host-name</i>} Example: Switch(config-network-group)# host 209.165.200.237	(Optional) Specifies the IP address or name of a host. If you specify a host address, you must use an IPv4 address.
Step 5	network-address {/nn <i>network-mask</i>} Example: Switch(config-network-group)# 209.165.200.241 255.255.255.224	(Optional) Specifies a subnet object. You must specify an IPv4 address for the network address. The default network mask is 255.255.255.255.
Step 6	group-object <i>nested-object-group-name</i> Example: Switch(config-network-group)# group-object my-nested-object-group	(Optional) Specifies a nested (child) object group to be included in the current (parent) object group. The child object group type must match that of the parent (for example, if you are creating a network object group, you must specify another network object group as the child). You can use duplicated objects in an object group only by nesting group objects. For example, if object 1 is in both group A and group B, you can define a group C that includes both A and B. However, you cannot include a group object that causes the group hierarchy to become circular (for example, you cannot include group A in group B and then also include group B in group A). While you can have an unlimited number of nested levels we recommend a maximum of two levels.
Step 7	Repeat the steps until you have specified objects on which you want to base your object group.	—
Step 8	end Example: Switch(config-network-group)# end	Exits network object-group configuration mode and returns to privileged EXEC mode.

Creating a Service Object Group

	Command or Action	Purpose
Step 1	configure terminal Example: Switch# configure terminal	Enters global configuration mode.
Step 2	object-group service <i>object-group-name</i> Example: Switch(config)# object-group service my-service-object-group	Defines the object group name and enters network object-group configuration mode.
Step 3	description <i>description-text</i> Example: Switch(config-service-group)# description test engineers	(Optional) Specifies a description of the object group. You can use up to 200 characters.
Step 4	<i>protocol</i> Example: Switch(config-service-group)# ahp	(Optional) Specifies an IP protocol number or name.
Step 5	{tcp udp tcp-udp} [source {[eq] lt gt} port1 range port1 port2}] [[eq] lt gt] port1 range port1 port2] Example: Switch(config-service-group)# tcp-udp range 2000 2005	(Optional) Specifies TCP, UDP, or both.
Step 6	icmp <i>icmp-type</i> Example: Switch(config-service-group)# icmp conversion-error	(Optional) Specifies the decimal number or name of an Internet Control Message Protocol (ICMP) type.

	Command or Action	Purpose
Step 7	group-object <i>nested-object-group-name</i> Example: Switch(config-service-group)# group-object my-nested-object-group	(Optional) Specifies a nested (child) object group to be included in the current (parent) object group. The child object group type must match that of the parent (for example, if you are creating a network object group, you must specify another network object group as the child). You can use duplicate objects in an object group only by nesting group objects. For example, if object 1 is in both group A and group B, you can define a group C that includes both A and B. However, you cannot include a group object that causes the group hierarchy to become circular (for example, you cannot include group A in group B and then also include group B in group A). While you can have an unlimited number of nested levels we recommend a maximum of two levels.
Step 8	Repeat the steps until you have specified objects on which you want to base your object group.	—
Step 9	end Example: Switch(config-network-group)# end	Exits network object-group configuration mode and returns to privileged EXEC mode.

Configuring an IPv4 OG ACL

When creating an object group ACL, configure an ACL that references one or more object groups. As with conventional ACLs, you can associate the same access policy with one or more interfaces.

You can define multiple ACEs that reference object groups within the same object group ACL. You can also reuse a specific object group in multiple ACEs. To create an object group ACL, perform the following task:

	Command or Action	Purpose
Step 1	configure terminal Example: Switch# configure terminal	Enters global configuration mode.
Step 2	ip access-list extended <i>access-list-name</i> Example: Switch(config)# ip access-list extended my-ogacl-policy	Defines an extended IP access list using a name and enters extended access-list configuration mode.

	Command or Action	Purpose
Step 3	remark <i>remark</i> Example: Switch(config-ext-nacl)# remark my-ogacl-policy is to provide the marketing network access to the server	(Optional) Adds a comment about the configured access list entry. A remark can precede or follow an access list entry. In this example, the remark reminds the network administrator that the subsequent entry denies the Marketing network access to the interface.
Step 4	permit <i>protocol source [source-wildcard] destination [destination-wildcard] [option option-name] [precedence precedence] [tos tos] [established] [log log-input] [time-range time-range-name] [fragments]</i> Example: Switch(config-ext-nacl)# permit object-group my-service-object-group object-group my-network-object-group any	Permits any packet that matches all conditions specified in the statement. Every access list needs at least one permit statement. Optionally use the object-group <i>service-object-group-name</i> keyword and argument as a substitute for the protocol. Optionally use the object-group <i>source-network-object-group-name</i> keyword and argument as a substitute for the source <i>source-wildcard</i> . Optionally use the object-group <i>destination-network-object-group-name</i> keyword and argument as a substitute for the destination <i>destination-wildcard</i> . If <i>source-wildcard</i> or <i>destination-wildcard</i> is omitted, a wildcard mask of 0.0.0.0 is assumed, which matches on all bits of the source or destination address, respectively. Optionally use the any keyword as a substitute for the source <i>source-wildcard</i> or destination <i>destination-wildcard</i> to specify the address and wildcard of 0.0.0.0 255.255.255.255. Use the log-input keyword to include input interface, source MAC address, or virtual circuit in the logging output.
Step 5	Repeat the steps to specify the fields and values on which you want to base your access list.	Remember that all sources not specifically permitted are denied by an implicit deny statement at the end of the access list.
Step 6	end Example: Device(config-ext-nacl)# end	Exits extended access-list configuration mode and returns to privileged EXEC mode.

Applying an IPv4 OG ACL to an Interface

An object group ACL can be used to control traffic on the interface it is applied to. To apply an object group ACL to an interface, perform the following task:

	Command or Action	Purpose
Step 1	configure terminal Example: Switch# configure terminal	Enters the global configuration mode.
Step 2	interface type number Example: Switch(config)# interface vlan 100	Specifies the interface and enters interface configuration mode.
Step 3	ip access-group {access-list-name access-list-number} {in out} Example: Switch(config-if)# ip access-group my-ogacl-policy in	Applies the ACL to the interface and specifies whether to filter inbound or outbound packets.
Step 4	end Example: Device(config-ext-nacl)# end	Exits interface configuration mode and returns to privileged EXEC mode.

Verifying IPv4 OG ACLs

Enter the **show object-group** *[object-group-name]* command, to display the configuration in the named or numbered object group (or in all object groups if no name is entered). For example:

```
Switch# show object-group
Network object group auth-proxy-acl-deny-dest
  host 209.165.200.235
Service object group auth-proxy-acl-deny-services
  tcp eq www
  tcp eq 443
Network object group auth-proxy-acl-permit-dest
  209.165.200.226 255.255.255.224
  209.165.200.227 255.255.255.224
  209.165.200.228 255.255.255.224
  209.165.200.229 255.255.255.224
  209.165.200.246 255.255.255.224
  209.165.200.230 255.255.255.224
  209.165.200.231 255.255.255.224
  209.165.200.232 255.255.255.224
  209.165.200.233 255.255.255.224
  209.165.200.234 255.255.255.224
Service object group auth-proxy-acl-permit-services
  tcp eq www
  tcp eq 443
```

Enter the **show ip access-list** *[access-list-name]* command, to display the contents of the named or numbered access list or object group ACL (or for all access lists and object group ACLs if no name is entered). For example:

```
Switch# show ip access-list my-ogacl-policy
Extended IP access list my-ogacl-policy
10 permit object-group my-service-object-group my-network-object-group any
```

Configuring IPv6 OG ACLs

- [Guidelines and Restrictions for Configuring IPv6 OG ACLs, page 62-46](#)
- [Creating a IPv6 Address Network Object Group, page 62-46](#)
- [Creating an IPv6 Service Object Group, page 62-47](#)
- [Configuring an IPv6 OG ACL, page 62-48](#)
- [Applying an IPv6 OG ACL to an Interface, page 62-49](#)
- [Verifying IPv6 OG ACLs, page 62-49](#)

Guidelines and Restrictions for Configuring IPv6 OG ACLs

- IPv6 OG ACLs are supported only on Layer 3 interfaces (such as routed interfaces and VLAN interfaces).
- Only Cisco IOS ACLs are supported. It is not supported with any other features. The **reflexive** and **evaluate** keywords are not supported.
- Only named extended Cisco IOS ACLs are supported. Numbered ACLs are not supported. As with regular ACEs, you can associate the same access policy with one or more interfaces.
- Feature interactions for IPv6 OG ACLs are the same as for Cisco IOS ACLs.
- The maximum number of object group-based ACEs supported in an IPv6 OG ACL is 2048.
- IPv6 OG ACEs are used only while processing hardware-switched packets.

ACL statements using object groups are ignored on those packets that are sent to the Route Processor, and such ACL statements are not used for filtering. To match such packets, regular ACEs (without object groups) need to be created in the same ACL.

Creating a IPv6 Address Network Object Group

To create an IPv6 address network object group, perform this task:

	Command or Action	Purpose
Step 1	configure terminal Example: Switch# configure terminal	Enters global configuration mode.
Step 2	object-group v6-network <i>object-group-name</i> Example: Switch(config)# object-group v6-network myOG	Defines the object group name and enters IPv6-address network object-group configuration mode.

	Command or Action	Purpose
Step 3	<p><code>{ ipv6-source-prefix ipv6-source-prefix description description-text exit group-object host { host-address host-name } no</code></p> <p>Example:</p> <pre>Switch(config-service-group)# description example of network object group Switch(config-v6network-group)# host 2001::1</pre>	<p>(Optional) Configures a member of the group and specifies a description of the object group.</p> <p>For the object group, you can configure a network address plus mask or a host (identified by host name or IPv6 address).</p>
Step 4	<p>Repeat the steps until you have specified all objects on which you want to base your object group.</p> <p>Example:</p> <pre>Switch(config-v6network-group)# 2002::1/64</pre>	—
Step 5	<p><code>{ end } { exit }</code></p> <p>Example:</p> <pre>Switch(config-network-group)# end</pre>	<p>To exit the configuration mode, enter the end command.</p> <p>To exit the IPv6-address object-group configuration mode, enter the exit command.</p>

Creating an IPv6 Service Object Group

To create an IPv6 service object group , perform this task:

	Command or Action	Purpose
Step 1	<p>configure terminal</p> <p>Example:</p> <pre>Switch# configure terminal</pre>	Enters global configuration mode.
Step 2	<p>object-group v6-service object-group-name</p> <p>Example:</p> <pre>Switch(config)# object-group v6-service mySG</pre>	Defines object group name and enters the service object-group configuration mode.
Step 3	<p><code>{ 0-255 ahp description description-text esp exit group-object hbh icmp ipv6 no pcp setp tcp tcp-udp udp }</code></p> <p>Example:</p> <pre>Switch(config-v6service-group)# description example of service object group Switch(config-v6service-group)# esp</pre>	<p>(Optional) Configures a member of the group and specifies a description of the object group.</p> <p>For the object group, you can configure a network address plus mask or a host (identified by host name or IPv6 address)</p>

	Command or Action	Purpose
Step 4	Repeat the steps until you have specified all objects on which you want to base your object group. Example: Switch(config-v6service-group)# ahp	—
Step 5	{ end } { exit } Example: Switch(config-v6service-group)# end	To exit the configuration mode, enter the end command. To exit the IPv6-address object-group configuration mode, enter the exit command.

Configuring an IPv6 OG ACL

	Command or Action	Purpose
Step 1	configure terminal Example: Switch# configure terminal	Enters global configuration mode.
Step 2	ipv6 access-list <i>access-list-name</i> Example: Switch(config)# ipv6 access-list example-ipv6-acl	Defines an OG ACL with the specified name and enters IPv6-ACL configuration mode.
Step 3	{ deny permit } { any proto object-group <i>service-object-group-name</i> } { host <i>ipv6-addr</i> <i>ipv6-prefix</i> <i>ipv6-addr</i> <i>ipv6-wildcard-bits</i> object-group <i>source-network-object-group-name</i> } { host <i>ipv6-addr</i> <i>ipv6-prefix</i> <i>ipv6-addr</i> <i>ipv6-wildcard-bits</i> object-group <i>dest-network-object-group-name</i> } Example: Switch(config-ext-nacl)# permit object-group mySG object-group myOG any sequence 10	(Optional) Permits any packet that matches all conditions specified in the statement. In this example, the service object group my SG, allows network object groups from myOG with any destination.
Step 4	Repeat the steps to specify the fields and values on which you want to base your access list.	Remember that all sources not specifically permitted are denied by an implicit deny statement at the end of the access list.
Step 5	{ end } { exit } Example: Switch(config-v6service-group)# end	To exit the configuration mode, enter the end command. To exit the IPv6-address object-group configuration mode, enter the exit command.

Applying an IPv6 OG ACL to an Interface

	Command or Action	Purpose
Step 1	configure terminal Example: Switch# configure terminal	Enters the global configuration mode.
Step 2	interface type / slot Example: Switch(config)# interface vlan 100	Specifies the interface and enters interface configuration mode. The interface-type must be a Layer 3 interface.
Step 3	ipv6 traffic-filter access-list-name {in out} Example: Switch(config-if)# ipv6 traffic-filter example-ipv6-acl in	Applies the ACL to the interface and specifies whether to filter inbound or outbound packets.
Step 4	end Example: Device(config-ext-nacl)# end	Exits interface configuration mode and returns to privileged EXEC mode.

Verifying IPv6 OG ACLs

Enter the **show ipv6 access-list [access-list-name]** command, to display the contents of the named access list or object group ACL (or for all access lists and object group ACLs if no name is entered). For example:

```
Switch# show ipv6 access-list example-ipv6-acl
IPv6 access list og-acl
permit object-group mySG object-group myOG any sequence 10
```

```
Switch# show ipv6 access-list example-ipv6-acl expanded
IPv6 access list og-acl
  permit tcp host 2001::1 any eq www sequence 10
  permit udp host 2001::1 any eq xdmcp sequence 10
  permit esp host 2001::1 any sequence 10
  permit hbh host 2001::1 any sequence 10
  permit tcp host 2002::1 any eq www sequence 10
  permit udp host 2002::1 any eq xdmcp sequence 10
  permit esp host 2002::1 any sequence 10
  permit hbh host 2002::1 any sequence 10
  permit tcp host 2003::1 any eq www sequence 10
  permit udp host 2003::1 any eq xdmcp sequence 10
  permit esp host 2003::1 any sequence 10
  permit hbh host 2003::1 any sequence 10
  permit tcp host 2001::255 any eq www sequence 10
  permit udp host 2001::255 any eq xdmcp sequence 10
  permit esp host 2001::255 any sequence 10
  permit hbh host 2001::255 any sequence 10
  permit tcp 2002::1/64 any eq www sequence 10
  permit udp 2002::1/64 any eq xdmcp sequence 10
  permit esp 2002::1/64 any sequence 10
  permit hbh 2002::1/64 any sequence 10
```

Configuring RA Guard

This section includes these topics:

- [Introduction, page 62-50](#)
- [Deployment, page 62-51](#)
- [Configuring RA Guard, page 62-51](#)
- [Examples, page 62-52](#)
- [Usage Guidelines, page 62-53](#)

Introduction

When deploying IPv6 networks, routers are configured to use IPv6 Router Advertisements to convey configuration information to hosts onlink. Router Advertisement is a critical part of the autoconfiguration process. The conveyed information includes the implied default router address obtained from the observed source address of the Router-Advertisement (RA) message. However, in some networks, invalid RAs are observed. This may happen because of misconfigurations or a malicious attacks on the network.

Devices acting as rogue routers may send illegitimate RAs. When using IPv6 within a single Layer 2 network segment, you can enable Layer 2 devices to drop rogue RAs before they reach end-nodes.

Beginning with Cisco IOS Release 54(SG)SG on Supervisor Engine 6-E (and 6L-E); Cisco IOS XE Release 3.3.0SG on Supervisor Engine 7-E; Cisco IOS XE Release 3.2.0XO on Supervisor Engine 7L-E, Cisco IOS XE Release 3.2.0XO on Supervisor Engine 8-E, and Cisco IOS XE Release 3.10.0E on Supervisor Engine 9-E, the Catalyst 4500 Series Switch supports RA Guard. This feature examines incoming Router-Advertisement and Router-Redirect packets and decides whether to switch or block them based solely on information found in the message and in the Layer 2 device configuration.

You can configure RA Guard in two modes (host and router) based on the device connected to the port.

- Host mode—All the Router-Advertisement and Router-Redirect messages are disallowed on the port.
- Router mode—All messages (RA/RS/Redirect) are allowed on the port; only host mode is supported.

You can configure Catalyst 4500 host ports to allow or disallow RA messages. Once a port is configured to disallow the Router-Advertisement and Router-Redirect packets, it filters the content of the received frames on that port and blocks Router-Advertisement or Router-Redirect frames.

When RA Guard is configured on a port, the following packets are dropped in hardware:

- Router-Advertisement packets —IPv6 ICMP packets with ICMP type = 134
- Router-Redirect packets—IPv6 ICMP packets with ICMP type = 137

Router Solicitation packets are sent out on the ports that are configured with RA Guard policy that defines the device role as a router.

Per port RA Guard ACL statistics are supported and displayed when you enter a **show ipv6 snooping counters interface** command. The statistics output displays the number of packets that have been dropped per port due to the RA Guard.

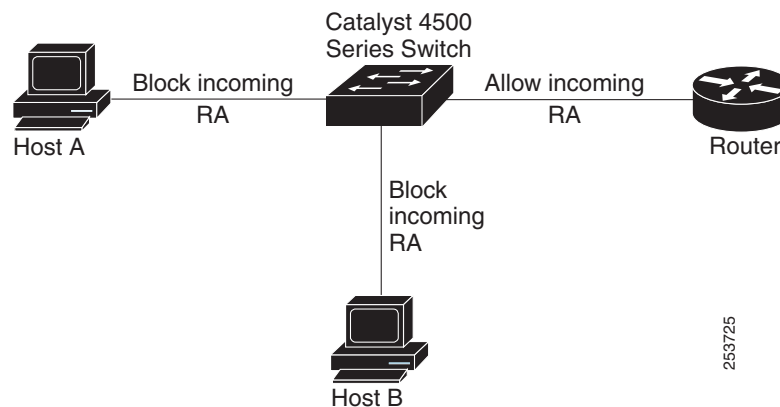
**Note**

Beginning with Cisco IOS Release 15.0(2)SG, per port RA Guard ACL statistics are supported and displayed when you enter a **show ipv6 snooping counters interface** command. (Previous to this release, you enter the **show ipv6 first-hop counters interface** command.)

Deployment

Figure 62-10 illustrates a deployment scenario for RA Guard. We drop RA packets from ports that are connected to hosts and permit RA packets from ports connected to the Router.

Figure 62-10 Typical RA Guard Deployment



Configuring RA Guard

To configure RA Guard, perform this step:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 1	Switch(config)# interface interface	Enters interface mode.
Step 2	Switch(config-if)# [no] ipv6 nd raguard	Enables RA Guard on the switch.
Step 3	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 4	Switch# show ipv6 nd raguard policy policy_name	Shows the policy on which RA Guard has been enabled.
		Note With Cisco Release IOS XE 3.4.0SG and IOS 15.1(2)SG, the show ipv6 nd raguard policy command replaces the show ipv6 first-hop policies command.

	Command	Purpose
Step 5	Switch# show ipv6 first-hop counters <i>interface</i>	Shows the number of packets dropped per port due to RA Guard. The counters can be displayed for a particular interface by using the <i>interface</i> option. Note If counters are not enabled for the port, the counter value is zero.
Step 6	Switch# clear ipv6 snooping counters <i>interface</i>	Clears RA Guard counters on a particular interface. The counters on all interfaces are cleared if the <i>interface</i> option is absent.

Examples

This examples shows how to enable RA Guard on the switch:

```
Switch(config)# int gi1/1
Switch(config-if)# ipv6 nd raguard
Switch(config-if)# end
Switch# show running-configuration interface gi1/1
```

Building configuration...

Current configuration : 53 bytes

!

```
interface GigabitEthernet1/1
```

```
    ipv6 nd raguard
```

```
end
```

The following example shows a sample output of the **show ipv6** commands:

```
Switch# show ipv6 snooping counters int gi 2/48
Received messages on gi 2/48 :
Protocol                Protocol message
NDP                      RS[9] RA[131] NS[7] NA[2]
DHCPv6                  SOL[24] ADV[2] REQ[1] REP[1]

Bridged messages from gi 2/48 :
Protocol                Protocol message
NDP                      RS[9] NS[7] NA[2]
DHCPv6                  SOL[24] ADV[1] REQ[1] REP[1]

Dropped messages on gi 2/48 :
Feature                Protocol Msg [Total dropped]
Snooping               NDP      RA  [131]
                        reason:  Packet not authorized on port [131]

                        NS   [2]
                        reason:  Packet accepted but not forwarded [2]

Switch#
```



Note

Beginning with Cisco IOS Release 15.0(2)SG, per port RA Guard ACL statistics are supported and displayed when you enter a **show ipv6 snooping counters** *interface* command. (Previous to this release, you enter the **show ipv6 first-hop counters** *interface* command.)

**Note**

Be aware that only RA (Router Advertisement) and REDIR (Router Redirected packets) counters are supported in 12.2(54)SG.

```
Switch# show ipv6 nd raguard policy RA_GUARD
Policy RA_GUARD configuration:
  device-role router
Policy RA_GUARD is applied on the following targets:
Target      Type  Policy      Feature      Target range
Gi 1/1      PORT RA_GUARD    RA guard     vlan all
```

Switch#

**Note**

With Cisco Release IOS XE 3.4.0SG and IOS 15.1(2)SG, the **show ipv6 nd raguard policy** command replaces the **show ipv6 first-hop policies** command.

Usage Guidelines

Observe the following restrictions:

- It is an ingress feature; only IPv6 Router-Advertisement and Router-Redirect packets entering through the port are filtered.
- RA Guard does not offer protection in environments where IPv6 traffic is tunneled.
- Starting with IOS XE 3.4.0SG/15.1(2)SG, RA Guard is supported in software. In prior releases, this Feature is supported only in hardware; packets are not punted to software except under resource exhaustion (for example, TCAM memory exhaustion).
- RA Guard is purely an Layer 2 port based feature and can be configured only on switchports. It works irrespective of whether IPv6 routing is enabled. It is supported on switchports and VLANs.
- RA Guard is supported on trunk ports and VLANs; filtering is performed on packets arriving from all the allowed VLANs.
- Starting with IOS XE 3.4.0SG/15.1(2)SG, RA Guard is not supported on EtherChannel. In prior releases, RA Guard is supported on EtherChannel; the RA Guard configuration (whether present or not) on the EtherChannel overrides the RA Guard configuration on the member ports.
- RA Guard is supported on ports that belong to PVLANS (for example, isolated secondary host ports, community secondary host ports, promiscuous primary host ports, (primary/secondary) trunk ports. Primary VLAN features are inherited and merged with port features.
- Starting with IOS XE 3.4.0SG/15.1(2)SG, RA Guard is supported on Supervisor Engine 8-E, 7-LE, and 7-E, 4500X-32, and 4500X-16 platforms. In prior releases, because of hardware limitations, it may not be possible for Supervisor Engine 6-E, Supervisor Engine 6L-E, Supervisor Engine 7-E and Supervisor Engine 7L-E to collect statistics for RA Guard in hardware. If so, an error message is displayed. Starting with Cisco IOS XE Release 3.10.0E, supported is extended to Supervisor Engine 9-E.

The **show ipv6 snooping counter** *interface* command displays the estimated counters.

**Note**

Beginning with Cisco IOS Release 15.0(2)SG, per port RA Guard ACL statistics are supported and displayed when you enter a **show ipv6 snooping counters** *interface* command. (Previous to this release, you enter the **show ipv6 first-hop counters** *interface* command.)



Configuring Authorization and Revocation of Certificates in a PKI

This module describes how to configure authorization and revocation of certificates in a public key infrastructure (PKI). It includes information on high-availability support for the certificate server.



Note

Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

Prerequisites for Authorization and Revocation of Certificates

Plan Your PKI Strategy



Tip

It is strongly recommended that you plan your entire PKI strategy before you begin to deploy actual certificates.

Authorization and revocation can occur only after you or a network administrator have completed the following tasks:

- Configured the certificate authority (CA).
- Enrolled peer devices with the CA.
- Identified and configured the protocol (such as IP Security [IPsec] or secure socket layer [SSL]) that is to be used for peer-to-peer communication.

You should decide which authorization and revocation strategy you are going to configure before enrolling peer devices because the peer device certificates might have to contain authorization and revocation-specific information.

“crypto ca” to “crypto pki” CLI Change

As of Cisco IOS Release 12.3(7)T, all commands that begin as “crypto ca” have been changed to begin as “crypto pki.” Although the router will still accept crypto ca commands, all output will be read back as crypto pki.

High Availability

For high availability, IPsec-secured Stream Control Transmission Protocol (SCTP) must be configured on both the active and the standby routers. For synchronization to work, the redundancy mode on the certificate servers must be set to ACTIVE/STANDBY after you configure SCTP.

Restrictions for Authorization and Revocation of Certificates

PKI High Availability (HA) support of intra-chassis stateful switchover (SSO) redundancy is currently not supported on all switches running the Cisco IOS Release 12.2 S software. See Cisco bug CSCtb59872 for more information.

?Depending on your Cisco IOS release, Lightweight Directory Access Protocol (LDAP) is supported.

Information About Authorization and Revocation of Certificates

PKI Authorization

PKI authentication does not provide authorization. Current solutions for authorization are specific to the router that is being configured, although a centrally managed solution is often required.

There is not a standard mechanism by which certificates are defined as authorized for some tasks and not for others. This authorization information can be captured in the certificate itself if the application is aware of the certificate-based authorization information. But this solution does not provide a simple mechanism for real-time updates to the authorization information and forces each application to be aware of the specific authorization information embedded in the certificate.

When the certificate-based ACL mechanism is configured as part of the trustpoint authentication, the application is no longer responsible for determining this authorization information, and it is no longer possible to specify for which application the certificate is authorized. In some cases, the certificate-based ACL on the router gets so large that it cannot be managed. Additionally, it is beneficial to retrieve certificate-based ACL indications from an external server.

Current solutions to the real-time authorization problem involve specifying a new protocol and building a new server (with associated tasks, such as management and data distribution).

PKI and AAA Server Integration for Certificate Status

Integrating your PKI with an authentication, authorization, and accounting (AAA) server provides an alternative online certificate status solution that leverages the existing AAA infrastructure. Certificates can be listed in the AAA database with appropriate levels of authorization. For components that do not explicitly support PKI-AAA, a default label of “all” from the AAA server provides authorization. Likewise, a label of “none” from the AAA database indicates that the specified certificate is not valid. (The absence of any application label is equivalent, but “none” is included for completeness and clarity). If the application component does support PKI-AAA, the component may be specified directly; for example, the application component could be “ipsec”, “ssl”, or “ocsp.” (ipsec=IP Security, ssl=Secure Sockets Layer, and osp=Open Settlement Protocol.)

**Note**

Currently, no application component supports specification of the application label.

- There may be a time delay when accessing the AAA server. If the AAA server is not available, the authorization fails.

RADIUS or TACACS+ Choosing a AAA Server Protocol

The AAA server can be configured to work with either the RADIUS or TACACS+ protocol. When you are configuring the AAA server for the PKI integration, you must set the RADIUS or TACACS attributes that are required for authorization.

If the RADIUS protocol is used, the password that is configured for the username in the AAA server should be set to Cisco, which is acceptable because the certificate validation provides authentication and the AAA database is only being used for authorization. When the TACACS protocol is used, the password that is configured for the username in the AAA server is irrelevant because TACACS supports authorization without requiring authentication (the password is used for authentication).

In addition, if you are using TACACS, you must add a PKI service to the AAA server. The custom attribute “cert-application=all” is added under the PKI service for the particular user or usergroup to authorize the specific username.

Attribute-Value Pairs for PKI and AAA Server Integration

The table below lists the attribute-value (AV) pairs that are to be used when setting up PKI integration with a AAA server. (Note the values shown in the table are possible values.) The AV pairs must match the client configuration. If they do not match, the peer certificate is not authorized.

**Note**

Users can sometimes have AV pairs that are different from those of every other user. As a result, a unique username is required for each user. The **all** parameter (within the **authorization username** command) specifies that the entire subject name of the certificate will be used as the authorization username.

Table 63-1 AV Pairs that Must Match




AV Pair	Value
cisco-avpair=pki:cert-application=all	Valid values are “all” and “none.”
cisco-avpair=pki:cert-trustpoint=msca	<p>The value is a Cisco IOS command-line interface (CLI) configuration trustpoint label.</p> <div>  <p>Note The cert-trustpoint AV pair is normally optional. If it is specified, the Cisco IOS router query must be coming from a certificate trustpoint that has a matching label, and the certificate that is authenticated must have the specified certificate serial number.</p> </div>

Table 63-1 AV Pairs that Must Match

AV Pair	Value
cisco-avpair=pki:cert-serial=16318DB7000100001671	<p>The value is a certificate serial number.</p> <div>  Note </div> <p>The cert-serial AV pair is normally optional. If it is specified, the Cisco IOS router query must be coming from a certificate trustpoint that has a matching label, and the certificate that is authenticated must have the specified certificate serial number.</p>
cisco-avpair=pki:cert-lifetime-end=1:00 jan 1, 2003	<p>The cert-lifetime-end AV pair is available to artificially extend a certificate lifetime beyond the time period that is indicated in the certificate itself. If the cert-lifetime-end AV pair is used, the cert-trustpoint and cert-serial AV pairs must also be specified. The value must match the following form: hours:minutes month day, year.</p> <div>  Note </div> <p>Only the first three characters of a month are used: Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec. If more than three characters are entered for the month, the remaining characters are ignored (for example Janxxxx).</p>

CRLs or OCSP Server Choosing a Certificate Revocation Mechanism

After a certificate is validated as a properly signed certificate, a certificate revocation method is performed to ensure that the certificate has not been revoked by the issuing CA. Cisco IOS software supports two revocation mechanisms--certificate revocation lists (CRLs) and Online Certificate Status Protocol (OCSP). Cisco IOS software also supports AAA integration for certificate checking; however, additional authorization functionality is included. For more information on PKI and AAA certificate authorization and status check, see the PKI and AAA Server Integration for Certificate Status section.

The following sections explain how each revocation mechanism works:

What Is a CRL

A certificate revocation list (CRL) is a list of revoked certificates. The CRL is created and digitally signed by the CA that originally issued the certificates. The CRL contains dates for when each certificate was issued and when it expires.

CAs publish new CRLs periodically or when a certificate for which the CA is responsible has been revoked. By default, a new CRL is downloaded after the currently cached CRL expires. An administrator may also configure the duration for which CRLs are cached in router memory or disable CRL caching completely. The CRL caching configuration applies to all CRLs associated with a trustpoint.

When the CRL expires, the router deletes it from its cache. A new CRL is downloaded when a certificate is presented for verification; however, if a newer version of the CRL that lists the certificate under examination is on the server but the router is still using the CRL in its cache, the router does not know that the certificate has been revoked. The certificate passes the revocation check even though it should have been denied.

When a CA issues a certificate, the CA can include in the certificate the CRL distribution point (CDP) for that certificate. Cisco IOS client devices use CDPs to locate and load the correct CRL. The Cisco IOS client supports multiple CDPs, but the Cisco IOS CA currently supports only one CDP; however, third-party vendor CAs may support multiple CDPs or different CDPs per certificate. If a CDP is not specified in the certificate, the client device uses the default Simple Certificate Enrollment Protocol (SCEP) method to retrieve the CRL. (The CDP location can be specified through the **cdp-url** command.)

When implementing CRLs, you should consider the following design considerations:

- CRL lifetimes and the security association (SA) and Internet Key Exchange (IKE) lifetimes.
- The CRL lifetime determines the length of time between CA-issued updates to the CRL. The default CRL lifetime value, which is 168 hours [1 week], can be changed through the **lifetime crl** command.
- The method of the CDP determines how the CRL is retrieved; some possible choices include HTTP, Lightweight Directory Access Protocol (LDAP), SCEP, or TFTP. HTTP, TFTP, and LDAP are the most commonly used methods. Although Cisco IOS software defaults to SCEP, an HTTP CDP is recommended for large installations using CRLs because HTTP can be made highly scalable.
- The location of the CDP determines from where the CRL is retrieved; for example, you can specify the server and file path from which to retrieve the CRL.

Querying All CDPs During Revocation Check

When a CDP server does not respond to a request, the Cisco IOS software reports an error, which may result in the peer's certificate being rejected. To prevent a possible certificate rejection and if there are multiple CDPs in a certificate, the Cisco IOS software will attempt to use the CDPs in the order in which they appear in the certificate. The router will attempt to retrieve a CRL using each CDP URL or directory specification. If an error occurs using a CDP, an attempt will be made using the next CDP.



Note

Prior to Cisco IOS Release 12.3(7)T, the Cisco IOS software makes only one attempt to retrieve the CRL, even when the certificate contains more than one CDP.



Tip

Although the Cisco IOS software will make every attempt to obtain the CRL from one of the indicated CDPs, it is recommended that you use an HTTP CDP server with high-speed redundant HTTP servers to avoid application timeouts because of slow CDP responses.

What Is OCSP

OCSP is an online mechanism that is used to determine certificate validity and provides the following flexibility as a revocation mechanism:

- OCSP can provide real-time certificate status checking.
- OCSP allows the network administrator to specify a central OCSP server, which can service all devices within a network.
- OCSP also allows the network administrator the flexibility to specify multiple OCSP servers, either per client certificate or per group of client certificates.
- OCSP server validation is usually based on the root CA certificate or a valid subordinate CA certificate, but may also be configured so that external CA certificates or self-signed certificates may be used. Using external CA certificates or self-signed certificates allows the OCSP servers certificate to be issued and validated from an alternative PKI hierarchy.

A network administrator can configure an OCSP server to collect and update CRLs from different CA servers. The devices within the network can rely on the OCSP server to check the certificate status without retrieving and caching each CRL for every peer. When peers have to check the revocation status of a certificate, they send a query to the OCSP server that includes the serial number of the certificate in question and an optional unique identifier for the OCSP request, or a nonce. The OCSP server holds a copy of the CRL to determine if the CA has listed the certificate as being revoked; the server then responds to the peer including the nonce. If the nonce in the response from the OCSP server does not match the original nonce sent by the peer, the response is considered invalid and certificate verification fails. The dialog between the OCSP server and the peer consumes less bandwidth than most CRL downloads.

If the OCSP server is using a CRL, CRL time limitations will be applicable; that is, a CRL that is still valid might be used by the OCSP server although a new CRL has been issued by the CRL containing additional certificate revocation information. Because fewer devices are downloading the CRL information on a regular basis, you can decrease the CRL lifetime value or configure the OCSP server not to cache the CRL. For more information, check your OCSP server documentation.

**Note**

OCSP multiple response handling: Support has been enabled for handling of multiple OCSP single responses from an OCSP responder in a response packet.

In addition to the debug log messages the following debug log message will be displayed:

CRYPTO_PKI: Number of single Responses in OCSP response:1 (this value can change depending upon the number of responses).

When to Use an OCSP Server

OCSP may be more appropriate than CRLs if your PKI has any of the following characteristics:

- Real-time certificate revocation status is necessary. CRLs are updated only periodically and the latest CRL may not always be cached by the client device. For example, if a client does not yet have the latest CRL cached and a newly revoked certificate is being checked, that revoked certificate will successfully pass the revocation check.
- There are a large number of revoked certificates or multiple CRLs. Caching a large CRL consumes large portions of Cisco IOS memory and may reduce resources available to other processes.
- CRLs expire frequently, causing the CDP to handle a larger load of CRLs.

**Note**

As of Cisco IOS Release 12.4(9)T or later, an administrator may configure CRL caching, either by disabling CRL caching completely or setting a maximum lifetime for a cached CRL per trustpoint.

When to Use Certificate-Based ACLs for Authorization or Revocation

Certificates contain several fields that are used to determine whether a device or user is authorized to perform a specified action.

Because certificate-based ACLs are configured on the device, they do not scale well for large numbers of ACLs; however, certificate-based ACLs do provide very granular control of specific device behavior. Certificate-based ACLs are also leveraged by additional features to help determine when PKI components such as revocation, authorization, or a trustpoint should be used. They provide a general mechanism allowing users to select a specific certificate or a group of certificates that are being validated for either authorization or additional processing.

Certificate-based ACLs specify one or more fields within the certificate and an acceptable value for each specified field. You can specify which fields within a certificate should be checked and which values those fields may or may not have.

There are six logical tests for comparing the field with the value--equal, not equal, contains, does not contain, less than, and greater than or equal. If more than one field is specified within a single certificate-based ACL, the tests of all of the fields within the ACL must succeed to match the ACL. The same field may be specified multiple times within the same ACL. More than one ACL may be specified, and ACL will be processed in turn until a match is found or all of the ACLs have been processed.

Ignore Revocation Checks Using a Certificate-Based ACL

Certificate-based ACLs can be configured to instruct your router to ignore the revocation check and expired certificates of a valid peer. Thus, a certificate that meets the specified criteria can be accepted regardless of the validity period of the certificate, or if the certificate meets the specified criteria, revocation checking does not have to be performed. You can also use a certificate-based ACL to ignore the revocation check when the communication with a AAA server is protected with a certificate.

Ignoring Revocation Lists

To allow a trustpoint to enforce CRLs except for specific certificates, enter the **match certificate** command with the **skip revocation-check** keyword. This type of enforcement is most useful in a hub-and-spoke configuration in which you also want to allow direct spoke-to-spoke connections. In pure hub-and-spoke configurations, all spokes connect only to the hub, so CRL checking is necessary only on the hub. For one spoke to communicate directly with another spoke, the **match certificate** command with the **skip revocation-check** keyword can be used for neighboring peer certificates instead of requiring a CRL on each spoke.

Ignoring Expired Certificates

To configure your router to ignore expired certificates, enter the **match certificate** command with the **allow expired-certificate** keyword. This command has the following purposes:

- If the certificate of a peer has expired, this command may be used to “allow” the expired certificate until the peer can obtain a new certificate.

- If your router clock has not yet been set to the correct time, the certificate of a peer will appear to be not yet valid until the clock is set. This command may be used to allow the certificate of the peer even though your router clock is not set.

**Note**

If Network Time Protocol (NTP) is available only via the IPsec connection (usually via the hub in a hub-and-spoke configuration), the router clock can never be set. The tunnel to the hub cannot be brought up because the certificate of the hub is not yet valid.

- "Expired" is a generic term for a certificate that is expired or that is not yet valid. The certificate has a start and end time. An expired certificate, for purposes of the ACL, is one for which the current time of the router is outside the start and end times specified in the certificate.

Skipping the AAA Check of the Certificate

If the communication with an AAA server is protected with a certificate, and you want to skip the AAA check of the certificate, use the **match certificate** command with the **skip authorization-check** keyword. For example, if a virtual private network (VPN) tunnel is configured so that all AAA traffic goes over that tunnel, and the tunnel is protected with a certificate, you can use the **match certificate** command with the **skip authorization-check** keyword to skip the certificate check so that the tunnel can be established.

The **match certificate** command and the **skip authorization-check** keyword should be configured after PKI integration with an AAA server is configured.

**Note**

If the AAA server is available only via an IPsec connection, the AAA server cannot be contacted until after the IPsec connection is established. The IPsec connection cannot be "brought up" because the certificate of the AAA server is not yet valid.

PKI Certificate Chain Validation

A certificate chain establishes a sequence of trusted certificates --from a peer certificate to the root CA certificate. Within a PKI hierarchy, all enrolled peers can validate the certificate of one another if the peers share a trusted root CA certificate or a common subordinate CA. Each CA corresponds to a trustpoint.

When a certificate chain is received from a peer, the default processing of a certificate chain path continues until the first trusted certificate, or trustpoint, is reached. In Cisco IOS Release 12.4(6)T and later releases, an administrator may configure the level to which a certificate chain is processed on all certificates including subordinate CA certificates.

Configuring the level to which a certificate chain is processed allows for the reauthentication of trusted certificates, the extension of a trusted certificate chain, and the completion of a certificate chain that contains a gap.

Reauthentication of Trusted Certificates

The default behavior is for the router to remove any trusted certificates from the certificate chain sent by the peer before the chain is validated. An administrator may configure certificate chain path processing so that the router does not remove CA certificates that are already trusted before chain validation, so that all certificates in the chain are re-authenticated for the current session.

Extending the Trusted Certificate Chain

The default behavior is for the router to use its trusted certificates to extend the certificate chain if there are any missing certificates in the certificate chain sent by the peer. The router will validate only certificates in the chain sent by the peer. An administrator may configure certificate chain path processing so that the certificates in the peer's certificate chain and the router's trusted certificates are validated to a specified point.

Completing Gaps in a Certificate Chain

An administrator may configure certificate chain processing so that if there is a gap in the configured Cisco IOS trustpoint hierarchy, certificates sent by the peer can be used to complete the set of certificates to be validated.

**Note**

If the trustpoint is configured to require parent validation and the peer does not provide the full certificate chain, the gap cannot be completed and the certificate chain is rejected and invalid.

**Note**

It is a configuration error if the trustpoint is configured to require parent validation and there is no parent trustpoint configured. The resulting certificate chain gap cannot be completed and the subordinate CA certificate cannot be validated. The certificate chain is invalid.

High-Availability Support

High-availability support for the certificate server is provided by:

- Synchronizing revoke commands with the standby certificate server.
- Sending serial-number commands when new certificates are issued.

This means that the standby certificate server is ready to issue certificates and certificate revocation lists (CRLs) if it becomes active.

Further high-availability support is provided by the following synchronizations with the standby:

- Certificate-server configuration.
- Pending requests.
- Grant and reject commands.
- For box-to-box high availability, which does not support configuration synchronization, a basic configuration synchronization mechanism is layered over a redundancy facility.
- Trustpoint configuration synchronization support.

How to Configure Authorization and Revocation of Certificates for Your PKI

Configuring PKI Integration with a AAA Server

Perform this task to generate a AAA username from the certificate presented by the peer and specify which fields within a certificate should be used to build the AAA database username.

The following restrictions should be considered when using the **all** keyword as the subject name for the **authorization username** command:

Some AAA servers limit the length of the username (for example, to 64 characters). As a result, the entire certificate subject name cannot be longer than the limitation of the server.

Some AAA servers limit the available character set that may be used for the username (for example, a space [] and an equal sign [=] may not be acceptable). You cannot use the **all** keyword for a AAA server having such a character-set limitation.

The **subject-name** command in the trustpoint configuration may not always be the final AAA subject name. If the fully qualified domain name (FQDN), serial number, or IP address of the router are included in a certificate request, the subject name field of the issued certificate will also have these components. To turn off the components, use the **fqdn**, **serial-number**, and **ip-address** commands with the **none** keyword.

CA servers sometimes change the requested subject name field when they issue a certificate. For example, CA servers of some vendors switch the relative distinguished names (RDNs) in the requested subject names to the following order: CN, OU, O, L, ST, and C. However, another CA server might append the configured LDAP directory root (for example, O=cisco.com) to the end of the requested subject name.

Depending on the tools you choose for displaying a certificate, the printed order of the RDNs in the subject name could be different. Cisco IOS software always displays the least significant RDN first, but other software, such as Open Source Secure Socket Layer (OpenSSL), does the opposite. Therefore, if you are configuring a AAA server with a full distinguished name (DN) (subject name) as the corresponding username, ensure that the Cisco IOS software style (that is, with the least significant RDN first) is used.

or


radius server host hostname [**key** string]



Note

Beginning with Cisco IOS XE Release 3.11.3aE, the legacy command **tacacs-server** is deprecated. Use the **tacacs server** command if the software running on your device is Cisco IOS XE Release 3.11.3aE or later releases.

	Command	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	aaa new-model Example: Router(config)# aaa new-model	Enables the AAA access control model.
Step 4	aaa authorization network <i>listname</i> [<i>method</i>] Example: Router (config)# aaa authorization network maxaaa group tacacs+	Sets the parameters that restrict user access to a network. ? <i>method</i> --Can be group radius , group tacacs+ , or group group-name .
Step 5	crypto pki trustpoint <i>name</i> Example: Route (config)# crypto pki trustpoint msca	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.

	Command	Purpose
Step 6	enrollment [mode] [retry period <i>minutes</i>] [retry count <i>number</i>] url <i>url</i> [pem] Example: <pre>Router (ca-trustpoint)# enrollment url http://caserver.myexample.com - or- Router (ca-trustpoint)# enrollment url http://[2001:DB8:1:1::1]:80</pre>	<p>Specifies the following enrollment parameters of the CA:</p> <p>(Optional) The mode keyword specifies the registration authority (RA) mode, if your CA system provides an RA. By default, RA mode is disabled.</p> <ul style="list-style-type: none"> • (Optional) The retry period keyword and minutes argument specifies the period, in minutes, in which the router waits before sending the CA another certificate request. Valid values are from 1 to 60. The default is 1. • (Optional) The retry count keyword and number argument specifies the number of times a router will resend a certificate request when it does not receive a response from the previous request. Valid values are from 1 to 100. The default is 10. • The url argument is the URL of the CA to which your router should send certificate requests. <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p> Note With the introduction of Cisco IOS Release 15.2(1)T, an IPv6 address can be added to the http: enrolment method. For example: http://[ipv6-address]:80. The IPv6 address must be enclosed in brackets in the URL. See the enrollment url (ca-trustpoint) command page for more information on the other enrollment methods that can be used.</p> </div> <ul style="list-style-type: none"> • (Optional) The pem keyword adds privacy-enhanced mail (PEM) boundaries to the certificate request.
Step 7	revocation-check <i>method</i> Example: <pre>Router (ca-trustpoint)# revocation-check crl</pre>	<p>(Optional) Checks the revocation status of a certificate.</p>
Step 8	exit Example: <pre>Router (ca-trustpoint)# exit</pre>	<p>Exits ca-trustpoint configuration mode and returns to global configuration mode.</p>

	Command	Purpose
Step 9	authorization username subjectname subjectname Example: Router (config)# authorization username subjectname serialnumber	Sets parameters for the different certificate fields that are used to build the AAA username. The <i>subjectname</i> argument can be any of the following: ? all --Entire distinguished name (subject name) of the certificate. ? commonname --Certification common name. ? country --Certificate country. • email --Certificate e-mail. ? ipaddress --Certificate IP address. ? locality --Certificate locality. ? organization --Certificate organization. ? organizationalunit --Certificate organizational unit. ? postalcode --Certificate postal code. ? serialnumber --Certificate serial number. ? state --Certificate state field. ? streetaddress --Certificate street address. ? title --Certificate title. ? unstructuredname --Certificate unstructured name.
Step 10	authorization list listname Example: Route (config)# authorization list maxaaa	Specifies the AAA authorization list.
Step 11	tacacs server servername Example: Router(config)# tacacs server newserver Router(config-tacacs-server)# key another_secret_key radius server host hostname [key string] Example: Router(config)# radius server host 192.0.2.1 key another_secret_key	Specifies a TACACS+ host. or Specifies a RADIUS host.

Troubleshooting Tips

To display debug messages for the trace of interaction (message type) between the CA and the router, use the debug crypto **pki transactions** command. (See the sample output, which shows a successful PKI integration with AAA server exchange and a failed PKI integration with AAA server exchange.)

Successful Exchange

```
Router# debug crypto pki transactions
Apr 22 23:15:03.695: CRYPTO_PKI: Found a issuer match
Apr 22 23:15:03.955: CRYPTO_PKI: cert revocation status unknown.
Apr 22 23:15:03.955: CRYPTO_PKI: Certificate validated without revocation check
Each line that shows "CRYPTO_PKI_AAA" indicates the state of the AAA authorization checks. Each
of the AAA AV pairs is indicated, and then the results of the authorization check are shown.

Apr 22 23:15:04.019: CRYPTO_PKI_AAA: checking AAA authorization (ipsecca_script_aalist,
PKIAAA-L, <all>)
Apr 22 23:15:04.503: CRYPTO_PKI_AAA: reply attribute ("cert-application" = "all")
Apr 22 23:15:04.503: CRYPTO_PKI_AAA: reply attribute ("cert-trustpoint" = "CA1")
Apr 22 23:15:04.503: CRYPTO_PKI_AAA: reply attribute ("cert-serial" = "15DE")
Apr 22 23:15:04.503: CRYPTO_PKI_AAA: authorization passed
Apr 22 23:12:30.327: CRYPTO_PKI: Found a issuer match
```

Failed Exchange

```
Router# debug crypto pki transactions
Apr 22 23:11:13.703: CRYPTO_PKI_AAA: checking AAA authorization =
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: reply attribute ("cert-application" = "all")
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: reply attribute ("cert-trustpoint" = "CA1")
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: reply attribute ("cert-serial" = "233D")
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: parsed cert-lifetime-end as: 21:30:00
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: timezone specific extended
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: cert-lifetime-end is expired
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: cert-lifetime-end check failed.
Apr 22 23:11:14.203: CRYPTO_PKI_AAA: authorization failed
```

In the above failed exchange, the certificate has expired.

Configuring a Revocation Mechanism for PKI Certificate Status Checking

Perform this task to set up a CRL as the certificate revocation mechanism--CRLs or OCSP--that is used to check the status of certificates in a PKI.

The revocation-check Command

Use the **revocation-check** command to specify at least one method (OCSP, CRL, or skip the revocation check) that is to be used to ensure that the certificate of a peer has not been revoked. For multiple methods, the order in which the methods are applied is determined by the order specified via this command.

If your router does not have the applicable CRL and is unable to obtain one or if the OCSP server returns an error, your router will reject the peer's certificate--unless you include the **none** keyword in your configuration. If the **none** keyword is configured, a revocation check will not be performed and the certificate will always be accepted.

Nonces and Peer Communications with OCSP Servers

When using OCSP, nonces, unique identifiers for OCSP requests, are sent by default during peer communications with your OCSP server. The use of nonces offers a more secure and reliable communication channel between the peer and OCSP server.

If your OCSP server does not support nonces, you may disable the sending of nonces. For more information, check your OCSP server documentation.

Before You Begin

- Before issuing any client certificates, the appropriate settings on the server (such as setting the CDP) should be configured.
- When configuring an OCSP server to return the revocation status for a CA server, the OCSP server must be configured with an OCSP response signing certificate that is issued by that CA server. Ensure that the signing certificate is in the correct format, or the router will not accept the OCSP response. See your OCSP manual for additional information.



Note

OCSP transports messages over HTTP, so there may be a time delay when you access the OCSP server.

If the OCSP server depends on normal CRL processing to check revocation status, the same time delay that affects CRLs will also apply to OCSP.

Configuring a Revocation Mechanism for PKI Certificate Status Checking

	Command	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	crypto pki trustpoint <i>name</i> Example: Router(config)# crypto pki trustpoint hazel	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.

	Command	Purpose
Step 4	ocsp url <i>url</i> Example: Router(ca-trustpoint)# ocsp url http://ocsp-server - or - Router(ca-trustpoint)# ocsp url http://10.10.10.1:80 - or - Router(ca-trustpoint)# ocsp url http://[2001DB8:1:1::2]:80	The <i>url</i> argument specifies the URL of an OCSP server so that the trustpoint can check the certificate status. This URL overrides the URL of the OCSP server (if one exists) in the Authority Info Access (AIA) extension of the certificate. All certificates associated with a configured trustpoint are checked by the OCSP server. The - or - URL can be a hostname, IPv4 address, or an IPv6 address.
Step 5	revocation-check <i>method1</i> [<i>method2</i> <i>method3</i>] Example: Router(ca-trustpoint)# revocation-check ocsp none	Checks the revocation status of a certificate. ? crl --Certificate checking is performed by a CRL. This is the default option. ? none --Certificate checking is ignored. ? ocsp --Certificate checking is performed by an OCSP server. If a second and third method are specified, each method will be used only if the previous method returns an error, such as a server being down.
Step 6	ocsp disable-nonce Example: Router(ca-trustpoint)# ocsp disable-nonce	(Optional) Specifies that a nonce, or an OCSP request unique identifier, will not be sent during peer communications with the OCSP server.
Step 7	exit Example: Router(ca-trustpoint)# exit	Returns to global configuration mode.
Step 8	exit Example: Router(config)# exit	Returns to privileged EXEC mode.
Step 9	show crypto pki certificates Example: Router# show crypto pki certificates	(Optional) Displays information about your certificates.
Step 10	show crypto pki trustpoints [<i>status</i> <i>label</i> [<i>status</i>]] Example: Router# show crypto pki trustpoints	Displays information about the trustpoint configured in router.

Configuring Certificate Authorization and Revocation Settings

Perform this task to specify a certificate-based ACL, to ignore revocation checks or expired certificates, to manually override the default CDP location, to manually override the OCSP server setting, to configure CRL caching, or to set session acceptance or rejection based on a certificate serial number, as appropriate.

Configuring Certificate-Based ACLs to Ignore Revocation Checks

To configure your router to use certificate-based ACLs to ignore revocation checks and expired certificates, perform the following steps:

- Identify an existing trustpoint or create a new trustpoint to be used when verifying the certificate of the peer. Authenticate the trustpoint if it has not already been authenticated. The router may enroll with this trustpoint if you want. Do not set optional CRLs for the trustpoint if you plan to use the **match certificate** command and **skip revocation-check** keyword.
- Determine the unique characteristics of the certificates that should not have their CRL checked and of the expired certificates that should be allowed.
- Define a certificate map to match the characteristics identified in the prior step.
- You can add the **match certificate** command and **skip revocation-check** keyword and the **match certificate** command and **allow expired-certificate** keyword to the trustpoint that was created or identified in the first step.



Note

Certificate maps are checked even if the peer's public key is cached. For example, when the public key is cached by the peer, and a certificate map is added to the trustpoint to ban a certificate, the certificate map is effective. This prevents a client with the banned certificate, which was once connected in the past, from reconnecting.

Manually Overriding CDPs in a Certificate

Users can override the CDPs in a certificate with a manually configured CDP. Manually overriding the CDPs in a certificate can be advantageous when a particular server is unavailable for an extended period of time. The certificate's CDPs can be replaced with a URL or directory specification without reissuing all of the certificates that contain the original CDP.

Manually Overriding the OCSP Server Setting in a Certificate

Administrators can override the OCSP server setting specified in the Authority Information Access (AIA) field of the client certificate or set by the issuing the **ocsp url** command. One or more OCSP servers may be manually specified, either per client certificate or per group of client certificates by the **match certificate override ocsp** command. The **match certificate override ocsp** command overrides the client certificate AIA field or the **ocsp url** command setting if a client certificate is successfully matched to a certificate map during the revocation check.



Note

Only one OCSP server can be specified per client certificate.

Configuring CRL Cache Control

By default, a new CRL will be downloaded after the currently cached CRL expires. Administrators can either configure the maximum amount of time in minutes a CRL remains in the cache by issuing the **crl cache delete-after** command or disable CRL caching by issuing the **crl cache none** command. Only the **crl-cache delete-after** command or the **crl-cache none** command may be specified. If both commands are entered for a trustpoint, the last command executed will take effect and a message will be displayed.

Neither the **crl-cache none** command nor the **crl-cache delete-after** command affects the currently cached CRL. If you configure the **crl-cache none** command, all CRLs downloaded after this command is issued will not be cached. If you configure the **crl-cache delete-after** command, the configured lifetime will only affect CRLs downloaded after this command is issued.

This functionality is useful is when a CA issues CRLs with no expiration date or with expiration dates days or weeks ahead.

Configuring Certificate Serial Number Session Control



A certificate serial number can be specified to allow a certificate validation request to be accepted or rejected by the trustpoint for a session. A session may be rejected, depending on certificate serial number session control, even if a certificate is still valid. Certificate serial number session control may be configured by using either a certificate map with the serial-number field or an AAA attribute, with the **cert-serial-not** command.

Using certificate maps for session control allows an administrator to specify a single certificate serial number. Using the AAA attribute allows an administrator to specify one or more certificate serial numbers for session control.


Before You Begin

- The trustpoint should be defined and authenticated before attaching certificate maps to the trustpoint.
- The certificate map must be configured before the CDP override feature can be enabled or the **serial-number** command is issued.
- The PKI and AAA server integration must be successfully completed to use AAA attributes as described in “PKI and AAA Server Integration for Certificate Status.”

	Command	Purpose
Step 1	enable	Enables privileged EXEC mode.
	Example: Router> enable	<ul style="list-style-type: none"> • Enter your password if prompted
Step 2	configure terminal	Enters global configuration mode.
	Example: Router# configure terminal	

Command	Purpose
Step 3 crypto pki certificate map label sequence-number Example: <pre>Router(config)# crypto pki certificate map Group 10</pre>	Defines values in a certificate that should be matched or not matched and enters ca-certificate-map configuration mode.
Step 4 <i>field-name match-criteria match-value</i> Example: <pre>Router(ca-certificate-map)# subject-name co MyExample</pre>	<p>Specifies one or more certificate fields together with their matching criteria and the value to match.</p> <p>The <i>field-name</i> is one of the following case-insensitive name strings or a date:</p> <ul style="list-style-type: none"> ? alt-subject-name ? expires-on ? issuer-name ? name ? serial-number ? subject-name ? unstructured-subject-name ? valid-start <p>Date field format is dd mm yyyy hh:mm:ss or mmm dd yyyy hh:mm:ss.</p> <div data-bbox="862 1066 911 1136">  Note </div> <div data-bbox="951 1108 1487 1598"> <p>The <i>match-criteria</i> is one of the following logical operators:</p> <ul style="list-style-type: none"> • co --contains (valid only for name fields and serial number field) • eq --equal (valid for name, serial number, and date fields) • ge --greater than or equal (valid only for date fields) • lt --less than (valid only for date fields) • nc --does not contain (valid only for name fields and serial number field) • ne --not equal (valid for name, serial number, and date fields) <p>The match-value is the name or date to test with the logical operator assigned by match-criteria.</p> </div> <div data-bbox="862 1629 911 1698">  Note </div> <div data-bbox="951 1671 1487 1812"> <p>Use this command only when setting up a certificate-based ACL--not when setting up a certificate-based ACL to ignore revocation checks or expired certificates.</p> </div>

	Command	Purpose
Step 5	exit Example: Router(ca-certificate-map)# exit	Returns to global configuration mode.
Step 6	crypto pki trustpoint <i>name</i> Example: Router(config)# crypto pki trustpoint Access2	Declares the trustpoint, given name and enters ca-trustpoint configuration mode.
Step 7	Do one of the following: <ul style="list-style-type: none"> crl-cache none crl-cache none <i>time</i> Example: Router(ca-trustpoint)# crl-cache none Example: Router(ca-trustpoint)# crl-cache delete-after 20	(Optional) Disables CRL caching completely for all CRLs associated with the trustpoint. The crl-cache none command does not affect any currently cached CRLs. All CRLs downloaded after this command is configured will not be cached. (Optional) Specifies the maximum time CRLs will remain in the cache for all CRLs associated with the trustpoint. ? <i>time</i> --The amount of time in minutes before the CRL is deleted. The crl-cache delete-after command does not affect any currently cached CRLs. The configured lifetime will only affect CRLs downloaded after this command is configured.
Step 8	match certificate <i>certificate-map-label</i> [allow expired-certificate skip revocation-check skip authorization-check Example: Router(ca-trustpoint)# match certificate Group skip revocation-check	(Optional) Associates the certificate-based ACL (that was defined via the crypto pki certificate map command) to a trustpoint. <i>certificate-map-label</i> --Must match the label argument specified via the crypto pki certificate map command. allow expired-certificate --Ignores expired certificates. skip revocation-check --Allows a trustpoint to enforce CRLs except for specific certificates. skip authorization-check --Skips the AAA check of a certificate when PKI integration with an AAA server is configured.

Command	Purpose
<p>Step 9</p> <p>match certificate <i>certificate-map-label</i> override cdp {<i>url</i> <i>directory</i>} <i>string</i></p> <p>Example:</p> <pre>Router(ca-trustpoint)# match certificate Group1 override cdp url http://server.cisco.com</pre>	<p>(Optional) Manually overrides the existing CDP entries for a certificate with a URL or directory specification.</p> <p><i>certificate-map-label</i> --A user-specified label that must match the label argument specified in a previously defined crypto pki certificate map command.</p> <ul style="list-style-type: none"> • url --Specifies that the certificate's CDPs will be overridden with an HTTP or LDAP URL. • directory --Specifies that the certificate's CDPs will be overridden with an LDAP directory specification. • <i>string</i> --The URL or directory specification. <div data-bbox="862 646 911 688"></div> <p>Note Some applications may time out before all CDPs have been tried and will report an error message. The error message will not affect the router, and the Cisco IOS software will continue attempting to retrieve a CRL until all CDPs have been tried.</p>
<p>Step 10</p> <p>match certificate <i>certificate-map-label</i> override oosp [<i>trustpoint trustpoint-label</i>] <i>sequence-number url oosp-url</i></p> <p>Example:</p> <pre>Router(ca-trustpoint)# match certificate mycertmapname override oosp trustpoint mytp 15 url http://192.0.2.2</pre>	<p>(Optional) Specifies an OOSP server, either per client certificate or per group of client certificates, and may be issued more than once to specify additional OOSP servers and client certificate settings including alternative PKI hierarchies.</p> <ul style="list-style-type: none"> • <i>certificate-map-label</i> --The name of an existing certificate map. • trustpoint --The trustpoint to be used when validating the OOSP server certificate. • <i>sequence-number</i> --The order the match certificate override oosp command statements apply to the certificate being verified. Matches are performed from the lowest sequence number to the highest sequence number. If more than one command is issued with the same sequence number, it overwrites the previous OOSP server override setting. • url --The URL of the OOSP server. <p>When the certificate matches a configured certificate map, the AIA field of the client certificate and any previously issued oosp url command settings are overwritten with the specified OOSP server.</p> <p>If no map-based match occurs, one of the following two cases will continue to apply to the client certificate.</p> <ul style="list-style-type: none"> • If OOSP is specified as the revocation method, the AIA field value will continue to apply to the client certificate. • If the oosp url configuration exists, the oosp url configuration settings will continue to apply to the client certificates.
<p>Step 11</p> <p>exit</p> <p>Example:</p> <pre>Router(ca-trustpoint)# exit</pre>	<p>Returns to global configuration mode.</p>

	Command	Purpose
Step 12	aaa new-model Example: Router(config)# aaa new-model	(Optional) Enables the AAA access control model.
Step 13	aaa attribute list list-name Example: Router(config)# aaa attribute list crl	(Optional) Defines an AAA attribute list locally on a router and enters config-attr-list configuration mode.
Step 14	attribute type {name} {value} Example: Router(config-attr-list)# attribute type cert-serial-not 6C4A	(Optional) Defines an AAA attribute type that is to be added to an AAA attribute list locally on a router. To configure certificate serial number session control, an administrator may specify a specific certificate in the value field to be accepted or rejected based on its serial number where name is set to cert-serial-not . If the serial number of the certificate matches the serial number specified by the attribute type setting, the certificate will be rejected. For a full list of available AAA attribute types, execute the show aaa attributes command.
Step 15	exit Example: Router(ca-trustpoint)# exit Example: Router(config-attr-list)# exit	Returns to global configuration mode.
Step 16	exit Example: Router(config)# exit	Returns to privileged EXEC mode.
Step 17	show crypto pki certificates Example: Router# show crypto pki certificates	(Optional) Displays the components of the certificates installed on the router if the CA certificate has been authenticated.

Example

The following is a sample certificate. The OCSP-related extensions are shown using exclamation points.

Certificate:

Data:

```

Version: v3
Serial Number:0x14
Signature Algorithm:SHAwithRSA - 1.2.840.113549.1.1.4
Issuer:CN=CA server,OU=PKI,O=Cisco Systems
Validity:
    Not Before:Thursday, August 8, 2002 4:38:05 PM PST
    Not After:Tuesday, August 7, 2003 4:38:05 PM PST
Subject:CN=OCSP server,OU=PKI,O=Cisco Systems

```

```

Subject Public Key Info:
  Algorithm:RSA - 1.2.840.113549.1.1.1
  Public Key:
    Exponent:65537
    Public Key Modulus:(2048 bits) :
      <snip>
Extensions:
  Identifier:Subject Key Identifier - 2.5.29.14
    Critical:no
    Key Identifier:
      <snip>
  Identifier:Authority Key Identifier - 2.5.29.35
    Critical:no
    Key Identifier:
      <snip>
!
  Identifier:OCSP NoCheck:- 1.3.6.1.5.5.7.48.1.5
    Critical:no
  Identifier:Extended Key Usage:- 2.5.29.37
    Critical:no
    Extended Key Usage:
      OCSPSigning
!

  Identifier:CRL Distribution Points - 2.5.29.31
    Critical:no
    Number of Points:1
    Point 0
      Distribution Point:
[URIName:ldap://CA-server/CN=CA server,OU=PKI,O=Cisco Systems]
  Signature:
    Algorithm:SHAwithRSA - 1.2.840.113549.1.1.4
  Signature:
    <snip>

```

The following example shows an excerpt of the running configuration output when adding a **match certificate override ocs** command to the beginning of an existing sequence:

```

match certificate map3 override ocs 5 url http://192.0.2.3/
show running-configuration
.
.
.
      match certificate map3 override ocs 5 url http://192.0.2.3/
      match certificate map1 override ocs 10 url http://192.0.2.1/
      match certificate map2 override ocs 15 url http://192.0.2.2/

```

The following example shows an excerpt of the running configuration output when an existing **match certificate override ocs** command is replaced and a trustpoint is specified to use an alternative PKI hierarchy:

```

match certificate map4 override ocs trustpoint tp4 10 url http://192.0.2.4/newvalue
show running-configuration
.
.
.
      match certificate map3 override ocs trustpoint tp3 5 url http://192.0.2.3/
      match certificate map1 override ocs trustpoint tp1 10 url http://192.0.2.1/
      match certificate map4 override ocs trustpoint tp4 10 url
http://192.0.2.4/newvalue
      match certificate map2 override ocs trustpoint tp2 15 url http://192.0.2.2/

```


Troubleshooting Tips

If you ignored revocation check or expired certificates, you should carefully check your configuration. Verify that the certificate map properly matches either the certificate or certificates that should be allowed or the AAA checks that should be skipped. In a controlled environment, try modifying the certificate map and determine what is not working as expected.

Configuring Certificate Chain Validation

Perform this task to configure the processing level for the certificate chain path of your peer certificates.

Before You Begin

- The device must be enrolled in your PKI hierarchy.
- The appropriate key pair must be associated with the certificate.



Note

A trustpoint associated with the root CA cannot be configured to be validated to the next level.

The **chain-validation** command is configured with the **continue** keyword for the trustpoint associated with the root CA, an error message will be displayed and the chain validation will revert to the default **chain-validation** command setting.

	Command	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	crypto pki trustpoint <i>name</i> Example: Router(config)# crypto pki trustpoint ca-sub1	Declares the trustpoint and a given name and enters ca-trustpoint configuration mode.

	Command	Purpose
Step 4	chain-validation [{ stop continue } [<i>parent-trustpoint</i>]] Example: Router(ca-trustpoint)# chain-validation continue ca-sub1	Configures the level to which a certificate chain is processed on all certificates including subordinate CA certificates. Use the stop keyword to specify that the certificate is already trusted. This is the default setting. Use the continue keyword to specify that the subordinate CA certificate associated with the trustpoint must be validated. The <i>parent-trustpoint</i> argument specifies the name of the parent trustpoint the certificate must be validated against.
Step 5	exit Example: Router(ca-trustpoint)# exit	Returns to global configuration mode

Configuring Certificate Servers for High Availability

You can configure certificate servers to synchronize revoke commands and send serial-number commands when new certificates are issued, preparing the standby certificate server to issue certificates and CRLs if it becomes active.

Prerequisites


The following conditions must be met for high availability on certificate servers:

- IPsec-secured SCTP must be configured on both the active and the standby routers.
- For synchronization to work, the redundancy mode on the certificate servers must be set to ACTIVE/STANDBY after you configure SCTP.

This section contains the following subsections:.

Setting Redundancy Mode on Certificate Servers to ACTIVE STANDBY

Perform this task on the active router to enable synchronization by setting the redundancy mode on the certificate servers to ACTIVE/STANDBY.

	Command	Purpose
Step 1	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 2	redundancy inter-device Example: Router(config)# redundancy inter-device	Configures redundancy and enters interdevice configuration mode.
Step 3	scheme standby <i>standby-group-name</i> Example: Router(config-red-interdevice)# scheme standby SB	Defines the redundancy scheme that is to be used. <ul style="list-style-type: none"> The only supported scheme is “standby.” <i>standby-group-name</i> --Must match the standby name specified in the standby name interface configuration command. Also, the standby name must be the same on both routers.
Step 4	exit Example: Router(config-red-interdevice)# exit	Exits interdevice configuration mode and returns to global configuration mode.
Step 5	interface <i>interface-name</i> Example: Router(config) # interface gigabitethernet0/1	Configures an interface type for the router and enters interface configuration mode.
Step 6	ip address <i>ip-address mask</i> Example: Router(config-if) ip address 10.0.0.1 255.255.255.0	Sets the local IP address for the interface.
Step 7	no ip route-cache cef Example: Router(config-if)# no ip route cache cef	Disables Cisco Express Forwarding operation on the interface.
Step 8	no ip route-cache Example: Router(config-if)# no ip route cache	Disables fast switching on the interface.
Step 9	standby ip <i>ip-address</i> Example: Router(config-if)# standby ip 10.0.0.3	Activates the Hot Standby Router Protocol (HSRP), <div>  Note </div> Configure the same address on the active and the standby routers.

	Command	Purpose
Step 10	standby priority <i>priority</i> Example: Router(config-if)# standby priority 50	Sets the HSRP priority to 50. The priority range is from 1 to 255, where 1 denotes the lowest priority and 255 the highest. The router in the HSRP group with the highest priority value becomes the active router.
Step 11	standby name <i>group-name</i> Example: Router(config-if)# standby name SB	Configures the name of the standby group. <ul style="list-style-type: none">The name specifies the HSRP group used. The HSRP group name must be unique on the router.
Step 12	standby delay minimum [<i>min-seconds</i>] reload [<i>reload-seconds</i>] Example: Router(config-if)# standby delay minimum 30 reload 60	Sets a delay for HSRP group initialization as follows: <ul style="list-style-type: none">The minimum delay after the interface comes up before initializing the HSRP groups is 30 seconds.The delay after the router has reloaded is 60 seconds.
Step 13	Repeat Steps 1-12 on the standby router, configuring the interface with a different IP address from that of the interface on the active router (Step 6).	--
Step 14	exit Example: Router(config-if)# exit	Returns to global configuration mode.
Step 15	exit Example: Router(config)# exit	Returns to privileged EXEC mode.
Step 16	show redundancy states Example: Router# show redundancy states	(Optional) Verifies the redundancy states: standby or active.

Configuring SCTP on the Active and Standby Certificate Servers


Perform this task on the active router to configure SCTP on both the active and the standby certificate server.

	Command	Purpose
Step 1	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 2	ipc zone default Example: Switch(config)# ipc zone default	Configures the interdevice communication protocol, Inter-Process Communication (IPC), and enters IPC zone configuration mode. Use this command to initiate the communication link between the active router and the standby router.
Step 3	association association-ID	Configures an association between the two devices and enters IPC association configuration mode.
Step 4	no shutdown Example: Router(config-ipczone-assoc)# no shutdown	Ensures that the server association is in the default (enabled) state.
Step 5	protocol sctp Example: Router(config-ipczone-assoc)# protocol sctp	Configures SCTP as the transport protocol and enters SCTP protocol configuration mode.
Step 6	local-port local-port-number Example: Router(config-ipc-protocol-sctp)# local-port 5000	Defines the local SCTP port number that is used to communicate with the redundant peer and enters IPC transport SCTP local configuration mode. local-port-number --There is not a default value. This argument must be configured for the local port to enable interdevice redundancy. Valid port values: 1 to 65535. The local port numbers should be the same as the remote port number on the peer router.
Step 7	local-ip device-real-ip-address [device-real-ip-address2] Example: Router(config-ipc-local-sctp)# local-ip 10.0.0.1	Defines at least one local IP address that is used to communicate with the redundant peer. The local IP addresses must match the remote IP addresses on the peer router. There can be either one or two IP addresses, which must be in global VPN routing and forwarding (VRF). A virtual IP address cannot be used.
Step 8	exit Example: Router(config-ipc-local-sctp)# exit	Exits IPC transport - SCTP local configuration mode.

	Command	Purpose
Step 9	remote-port <i>remote-port-number</i> Example: Router(config-ipc-protocol-sctp)# remote-port 5000	Defines the remote SCTP port number that is used to communicate with the redundant peer and enters IPC transport SCTP remote configuration mode. Note remote-port-number --There is not a default value. This argument must be configured for the remote port to enable interdevice redundancy. Valid port values: 1 to 65535. The remote port number should be the same as the local port number on the peer router.
Step 10	remote-ip <i>peer-real-ip-address</i> Example: Router(config-ipc-remote-sctp)# remote-ip 10.0.0.2	Defines a remote IP address of the redundant peer that is used to communicate with the local device. All remote IP addresses must refer to the same device. A virtual IP address cannot be used.
Step 11	Repeat Steps 1 through 10 on the standby router, reversing the IP addresses of the local and remote peers specified in Steps 7 and 10.	The virtual IP address (10.0.0.3) will be the same on both routers.

Synchronizing the Active and Standby Certificate Servers

Perform this task to synchronize the active and standby servers.

	Command	Purpose
Step 1	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 2	crypto key generate rsa general-keys redundancy label <i>key-label</i> modulus <i>modulus-size</i> Example: Router (config)# crypto key generate rsa general-keys redundancy label HA modulus 2048	Generates an RSA key pair named HA for the certificate server.  Note Specifying the redundancy keyword means that the keys will be non-exportable.
Step 3	exit Example: Router(config)# exit	Returns to privileged EXEC mode.
Step 4	show crypto key mypubkey rsa Example: Router# show crypto key mypubkey rsa	Verifies that redundancy is enabled.

	Command	Purpose
Step 5	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 6	ip http server Example: Router(config)# ip http server	Enables the HTTP server on your system
Step 7	crypto pki server cs-label Example: Router(config)# crypto pki server HA	Specifies the RSA key pair generated in Step 2 as the label for the certificate server and enters certificate server configuration mode.
Step 8	redundancy Example: Router (cs-server)# redundancy	Ensures that the server is synchronized to the standby server
Step 9	no shutdown Example: Router(cs-server)# no shutdown	Enables the certificate server.

**Note**

If the router interface with the SCTP traffic is not secure, you should ensure that the SCTP traffic between the high-availability devices is secured with IPsec.

Configuration Examples for Setting Up Authorization and Revocation of Certificates

Configuring and Verifying PKI AAA Authorization Examples

This section provides configuration examples of PKI AAA authorizations:

Router Configuration Example

The following **show running-config** command output shows the working configuration of a router that is set up to authorize VPN connections using the PKI Integration with AAA Server feature:

```
Router# show running-config
Building configuration...
!
version 12.3
!
hostname router7200router7200
!
aaa new-model
!
!
```

```

aaa authentication login default group tacacs+
aaa authentication login no_tacacs enable
aaa authentication ppp default group tacacs+
aaa authorization exec ACSLab group tacacs+
aaa authorization network ACSLab group tacacs+
aaa accounting exec ACSLab start-stop group tacacs+
aaa accounting network default start-stop group ACSLab
aaa session-id common
!
ip domain name example.com
!
crypto pki trustpoint EM-CERT-SERV
  enrollment url http://192.0.2.33:80
  serial-number
  crl optional
  rsakeypair STOREVPN 2048
  auto-enroll
  authorization list ACSLab
!
crypto pki certificate chain EM-CERT-SERV
  certificate 04
    30820214 3082017D A0030201 02020104 300D0609 2A864886 F70D0101 04050030
    17311530 13060355 0403130C 454D2D43 4552542D 53455256 301E170D 30343031
    31393232 30323535 5A170D30 35303131 38323230 3235355A 3030312E 300E0603
    55040513 07314437 45424434 301C0609 2A864886 F70D0109 02160F37 3230302D
    312E6772 696C2E63 6F6D3081 9F300D06 092A8648 86F70D01 01010500 03818D00
    30818902 818100BD F3B837AA D925F391 2B64DA14 9C2EA031 5A7203C4 92F8D6A8
    7D2357A6 BCC8596F A38A9B10 47435626 D59A8F2A 123195BB BE5A1E74 B1AA5AE0
    5CA162FF 8C3ACA4F B3EE9F27 8B031642 B618AE1B 40F2E3B4 F996BEFE 382C7283
    3792A369 236F8561 8748AA3F BC41F012 B859BD9C DB4F75EE 3CEE2829 704BD68F
    FD904043 0F555702 03010001 A3573055 30250603 551D1F04 1E301C30 1AA018A0
    16861468 7474703A 2F2F3633 2E323437 2E313037 2E393330 0B060355 1D0F0404
    030205A0 301F0603 551D2304 18301680 1420FC4B CF0B1C56 F5BD4C06 0AFD4E67
    341AE612 D1300D06 092A8648 86F70D01 01040500 03818100 79E97018 FB955108
    12F42A56 2A6384BC AC8E22FE F1D6187F DA5D6737 C0E241AC AAAEC75D 3C743F59
    08DEEFF2 0E813A73 D79E0FA9 D62DC20D 8E2798CD 2C1DC3EC 3B2505A1 3897330C
    15A60D5A 8A13F06D 51043D37 E56E45DF A65F43D7 4E836093 9689784D C45FD61D
    EC1F160C 1ABC8D03 49FB11B1 DA0BED6C 463E1090 F34C59E4
  quit
  certificate ca 01
    30820207 30820170 A0030201 02020101 300D0609 2A864886 F70D0101 04050030
    17311530 13060355 0403130C 454D2D43 4552542D 53455256 301E170D 30333132
    31363231 34373432 5A170D30 36313231 35323134 3734325A 30173115 30130603
    55040313 0C454D2D 43455254 2D534552 5630819F 300D0609 2A864886 F70D0101
    01050003 818D0030 81890281 8100C14D 833641CF D784F516 DA6B50C0 7B3CB3C9
    589223AB 99A7DC14 04F74EF2 AAEE8F5 E3BFAE97 F2F980F7 D889E6A1 2C726C69
    54A29870 7E7363FF 3CD1F991 F5A37CFF 3FFDD3D0 9E486C44 A2E34595 C2D078BB
    E9DE981E B733B868 AA8916C0 A8048607 D34B83C0 64BDC101 161FC103 13C06500
    22D6EE75 7D6CF133 7F1B515F 32830203 010001A3 63306130 0F060355 1D130101
    FF040530 030101FF 300E0603 551D0F01 01FF0404 03020186 301D0603 551D0E04
    16041420 FC4BCF0B 1C56F5BD 4C060AFD 4E67341A E612D130 1F060355 1D230418
    30168014 20FC4BCF 0B1C56F5 BD4C060A FD4E6734 1AE612D1 300D0609 2A864886
    F70D0101 04050003 81810085 D2E386F5 4107116B AD3AC990 CBE84063 5FB2A6B5
    BD572026 528E92ED 02F3A0AE 1803F2AE AA4C0ED2 0F59F18D 7B50264F 30442C41
    0AF19C4E 70BD3CB5 0ADD8DE8 8EF636BD 24410DF4 DB62DAFC 67DA6E58 3879AA3E
    12AFB1C3 2E27CB27 EC74E1FC AEE2F5CF AA80B439 615AA8D5 6D6DEDC3 7F9C2C79
    3963E363 F2989FB9 795BA8
  quit
!
!
crypto isakmp policy 10
  encr aes
  group 14
!

```



```

!
crypto ipsec transform-set ISC_TS_1 esp-aes esp-sha-hmac
!
crypto ipsec profile ISC_IPSEC_PROFILE_2
  set security-association lifetime kilobytes 530000000
  set security-association lifetime seconds 14400
  set transform-set ISC_TS_1
!
!
controller ISA 1/1
!
!
interface Tunnel0
  description MGRE Interface provisioned by ISC
  bandwidth 10000
  ip address 192.0.2.172 255.255.255.0
  no ip redirects
  ip mtu 1408
  ip nhrp map multicast dynamic
  ip nhrp network-id 101
  ip nhrp holdtime 500
  ip nhrp server-only
  no ip split-horizon eigrp 101
  tunnel source FastEthernet2/1
  tunnel mode gre multipoint
  tunnel key 101
  tunnel protection ipsec profile ISC_IPSEC_PROFILE_2
!
interface FastEthernet2/0
  ip address 192.0.2.1 255.255.255.0
  duplex auto
  speed auto
!
interface FastEthernet2/1
  ip address 192.0.2.2 255.255.255.0
  duplex auto
  speed auto
!
!
tacacs-server host 192.0.2.55 single-connection
tacacs-server directed-request
tacacs-server key company lab
!
ntp master 1
!
end

```

Debug of a Successful PKI AAA Authorization Example

The following **show debugging** command output shows a successful authorization using the PKI Integration with AAA Server feature:

```

Router# show debugging
General OS:
  TACACS access control debugging is on
  AAA Authentication debugging is on
  AAA Authorization debugging is on
Cryptographic Subsystem:
  Crypto PKI Trans debugging is on
Router#
May 28 19:36:11.117: CRYPTO_PKI: Trust-Point EM-CERT-SERV picked up

```

```

May 28 19:36:12.789: CRYPTO_PKI: Found a issuer match
May 28 19:36:12.805: CRYPTO_PKI: cert revocation status unknown.
May 28 19:36:12.805: CRYPTO_PKI: Certificate validated without revocation check
May 28 19:36:12.813: CRYPTO_PKI_AAA: checking AAA authorization (ACSLab, POD5.example.com,
<all>)
May 28 19:36:12.813: AAA/BIND(00000042): Bind i/f
May 28 19:36:12.813: AAA/AUTHOR (0x42): Pick method list 'ACSLab'
May 28 19:36:12.813: TPLUS: Queuing AAA Authorization request 66 for processing
May 28 19:36:12.813: TPLUS: processing authorization request id 66
May 28 19:36:12.813: TPLUS: Protocol set to None .....Skipping
May 28 19:36:12.813: TPLUS: Sending AV service=pki
May 28 19:36:12.813: TPLUS: Authorization request created for 66(POD5.example.com)
May 28 19:36:12.813: TPLUS: Using server 192.0.2.55
May 28 19:36:12.813: TPLUS(00000042)/0/NB_WAIT/203A4628: Started 5 sec timeout
May 28 19:36:12.813: TPLUS(00000042)/0/NB_WAIT: wrote entire 46 bytes request
May 28 19:36:12.813: TPLUS: Would block while reading pak header
May 28 19:36:12.817: TPLUS(00000042)/0/READ: read entire 12 header bytes (expect 27 bytes)
May 28 19:36:12.817: TPLUS(00000042)/0/READ: read entire 39 bytes response
May 28 19:36:12.817: TPLUS(00000042)/0/203A4628: Processing the reply packet
May 28 19:36:12.817: TPLUS: Processed AV cert-application=all
May 28 19:36:12.817: TPLUS: received authorization response for 66: PASS
May 28 19:36:12.817: CRYPTO_PKI_AAA: reply attribute ("cert-application" = "all")
May 28 19:36:12.817: CRYPTO_PKI_AAA: authorization passed
Router#
Router#
May 28 19:36:18.681: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 101: Neighbor 192.0.2.171 (Tunnel0) is
up: new adjacency
Router#
Router# show crypto isakmp sa

```

dst	src	state	conn-id	slot
192.0.2.22	192.0.2.102	QM_IDLE	84	0

Debugs of a Failed PKI AAA Authorization Example

The following **show debugging** command output shows that the router is not authorized to connect using VPN. The messages are typical of those that you might see in such a situation.

In this example, the peer username was configured as not authorized, by moving the username to a Cisco Secure ACS group called VPN_Router_Disabled in Cisco Secure ACS. The router, router7200.example.com, has been configured to check with a Cisco Secure ACS AAA server prior to establishing a VPN connection to any peer.

```

Router# show debugging
General OS:
  TACACS access control debugging is on
  AAA Authentication debugging is on
  AAA Authorization debugging is on
Cryptographic Subsystem:
  Crypto PKI Trans debugging is on

Router#
May 28 19:48:29.837: CRYPTO_PKI: Trust-Point EM-CERT-SERV picked up
May 28 19:48:31.509: CRYPTO_PKI: Found a issuer match
May 28 19:48:31.525: CRYPTO_PKI: cert revocation status unknown.
May 28 19:48:31.525: CRYPTO_PKI: Certificate validated without revocation check
May 28 19:48:31.533: CRYPTO_PKI_AAA: checking AAA authorization (ACSLab, POD5.example.com,
<all>)
May 28 19:48:31.533: AAA/BIND(00000044): Bind i/f
May 28 19:48:31.533: AAA/AUTHOR (0x44): Pick method list 'ACSLab'
May 28 19:48:31.533: TPLUS: Queuing AAA Authorization request 68 for processing

```

```

May 28 19:48:31.533: TPLUS: processing authorization request id 68
May 28 19:48:31.533: TPLUS: Protocol set to None .....Skipping
May 28 19:48:31.533: TPLUS: Sending AV service=pki
May 28 19:48:31.533: TPLUS: Authorization request created for 68(POD5.example.com)
May 28 19:48:31.533: TPLUS: Using server 192.0.2.55
May 28 19:48:31.533: TPLUS(00000044)/0/NB_WAIT/203A4C50: Started 5 sec timeout
May 28 19:48:31.533: TPLUS(00000044)/0/NB_WAIT: wrote entire 46 bytes request
May 28 19:48:31.533: TPLUS: Would block while reading pak header
May 28 19:48:31.537: TPLUS(00000044)/0/READ: read entire 12 header bytes (expect 6 bytes)
May 28 19:48:31.537: TPLUS(00000044)/0/READ: read entire 18 bytes response
May 28 19:48:31.537: TPLUS(00000044)/0/203A4C50: Processing the reply packet
May 28 19:48:31.537: TPLUS: received authorization response for 68: FAIL
May 28 19:48:31.537: CRYPTO_PKI_AAA: authorization declined by AAA, or AAA server not
found.
May 28 19:48:31.537: CRYPTO_PKI_AAA: No cert-application attribute found. Failing.
May 28 19:48:31.537: CRYPTO_PKI_AAA: authorization failed
May 28 19:48:31.537: CRYPTO_PKI: AAA authorization for list 'ACSLab', and user
'POD5.example.com' failed.
May 28 19:48:31.537: %CRYPTO-5-IKMP_INVALID_CERT: Certificate received from 192.0.2.162 is
bad: certificate invalid
May 28 19:48:39.821: CRYPTO_PKI: Trust-Point EM-CERT-SERV picked up
May 28 19:48:41.481: CRYPTO_PKI: Found a issuer match
May 28 19:48:41.501: CRYPTO_PKI: cert revocation status unknown.
May 28 19:48:41.501: CRYPTO_PKI: Certificate validated without revocation check
May 28 19:48:41.505: CRYPTO_PKI_AAA: checking AAA authorization (ACSLab, POD5.example.com,
<all>)
May 28 19:48:41.505: AAA/BIND(00000045): Bind i/f
May 28 19:48:41.505: AAA/AUTHOR (0x45): Pick method list 'ACSLab'
May 28 19:48:41.505: TPLUS: Queuing AAA Authorization request 69 for processing
May 28 19:48:41.505: TPLUS: processing authorization request id 69
May 28 19:48:41.505: TPLUS: Protocol set to None .....Skipping
May 28 19:48:41.505: TPLUS: Sending AV service=pki
May 28 19:48:41.505: TPLUS: Authorization request created for 69(POD5.example.com)
May 28 19:48:41.505: TPLUS: Using server 198.168.244.55
May 28 19:48:41.509: TPLUS(00000045)/0/IDLE/63B22834: got immediate connect on new 0
May 28 19:48:41.509: TPLUS(00000045)/0/WRITE/63B22834: Started 5 sec timeout
May 28 19:48:41.509: TPLUS(00000045)/0/WRITE: wrote entire 46 bytes request
May 28 19:48:41.509: TPLUS(00000045)/0/READ: read entire 12 header bytes (expect 6 bytes)
May 28 19:48:41.509: TPLUS(00000045)/0/READ: read entire 18 bytes response
May 28 19:48:41.509: TPLUS(00000045)/0/63B22834: Processing the reply packet
May 28 19:48:41.509: TPLUS: received authorization response for 69: FAIL
May 28 19:48:41.509: CRYPTO_PKI_AAA: authorization declined by AAA, or AAA server not
found.
May 28 19:48:41.509: CRYPTO_PKI_AAA: No cert-application attribute found. Failing.
May 28 19:48:41.509: CRYPTO_PKI_AAA: authorization failed
May 28 19:48:41.509: CRYPTO_PKI: AAA authorization for list 'ACSLab', and user
'POD5.example.com' failed.
May 28 19:48:41.509: %CRYPTO-5-IKMP_INVALID_CERT: Certificate received from 192.0.2.162 is
bad: certificate invalid
Router#
Router# show crypto ismp sa
dst          src          state          conn-id slot
192.0.2.2    192.0.2.102  MM_KEY_EXCH    95          0

```

Configuring a Revocation Mechanism Examples

This section contains the following configuration examples that can be used when specifying a revocation mechanism for your PKI:

Configuring an OCSP Server Example

The following example shows how to configure the router to use the OCSP server that is specified in the AIA extension of the certificate:

```
Router(config)# crypto pki trustpoint mytp
Router(ca-trustpoint)# revocation-check ocsp
```

Specifying a CRL and Then an OCSP Server Example

The following example shows how to configure the router to download the CRL from the CDP. If the CRL is unavailable, the OCSP server that is specified in the AIA extension of the certificate will be used. If both options fail, certificate verification will also fail.

```
Router(config)# crypto pki trustpoint mytp
Router(ca-trustpoint)# revocation-check crl ocsp
```

Specifying an OCSP Server Example

The following example shows how to configure your router to use the OCSP server at the HTTP URL “http://myocspserver:81.” If the server is down, the revocation check will be ignored.

```
Router(config)# crypto pki trustpoint mytp
Router(ca-trustpoint)# ocsp url http://myocspserver:81
Router(ca-trustpoint)# revocation-check ocsp none
```

Disabling Nonces in Communications with the OCSP Server Example

The following example shows communications when a nonce, or a unique identifier for the OCSP request, is disabled for communications with the OCSP server:

```
Router(config)# crypto pki trustpoint mytp
Router(ca-trustpoint)# ocsp url http://myocspserver:81
Router(ca-trustpoint)# revocation-check ocsp none
Router(ca-trustpoint)# ocsp disable-nonce
```

Configuring a Hub Router at a Central Site for Certificate Revocation Checks Example

The following example shows a hub router at a central site that is providing connectivity for several branch offices to the central site.

The branch offices are also able to communicate directly with each other using additional IPSec tunnels between the branch offices.

The CA publishes CRLs on an HTTP server at the central site. The central site checks CRLs for each peer when setting up an IPSec tunnel with that peer.

The example does not show the IPSec configuration--only the PKI-related configuration is shown.

Home Office Hub Configuration

```
crypto pki trustpoint VPN-GW
enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
serial-number none
fqdn none
ip-address none
subject-name o=Home Office Inc,cn=Central VPN Gateway
revocation-check crl
```

Central Site Hub Router

```
Router# show crypto ca certificate
Certificate
Status: Available
Certificate Serial Number: 2F62BE1400000000CA0
Certificate Usage: General Purpose
Issuer:
cn=Central Certificate Authority o=Home Office Inc
Subject:
Name: Central VPN Gateway cn=Central VPN Gateway o=Home Office Inc
CRL Distribution Points:
  http://ca.home-office.com/CertEnroll/home-office.crl Validity Date:
start date: 00:43:26 GMT Sep 26 2003
end date: 00:53:26 GMT Sep 26 2004
  renew date: 00:00:00 GMT Jan 1 1970 Associated Trustpoints: VPN-GW
CACertificate Status: Available
Certificate Serial Number: 1244325DE0369880465F977A18F61CA8 Certificate Usage: Signature
Issuer:
cn=Central Certificate Authority o=Home Office Inc
Subject:
cn=Central Certificate Authority o=Home Office Inc
CRL Distribution Points:
  http://ca.home-office.com/CertEnroll/home-office.crl Validity Date:
start date: 22:19:29 GMT Oct 31 2002
  enddate: 22:27:27 GMT Oct 31 2017 Associated Trustpoints: VPN-GW
```

Trustpoint on the Branch Office Router

```
crypto pki trustpoint home-office
enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll serial-number none
fqdn none

ip-address none
subject-name o=Home Office Inc,cn=Branch 1 revocation-check crl
```

A certificate map is entered on the branch office router.

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
branch1(config)# crypto pki certificate map central-site 10
branch1(ca-certificate-map)#
```

The output from the **show certificate** command on the central site hub router shows that the certificate was issued by the following:

```
cn=Central Certificate Authority
o=Home Office Inc
```

These two lines are combined into one line using a comma (,) to separate them, and the original lines are added as the first criteria for a match.

```
Router (ca-certificate-map)# issuer-name co cn=Central Certificate Authority, ou=Home Office Inc
```

!The above line wrapped but should be shown on one line with the line above it.

The same combination is done for the subject name from the certificate on the central site router (note that the line that begins with ?ame:? is not part of the subject name and must be ignored when creating the certificate map criteria). This is the subject name to be used in the certificate map.

cn=Central VPN Gateway

o=Home Office Inc

```
Router (ca-certificate-map)# subject-name eq cn=central vpn gateway, o=home office inc
```

Now the certificate map is added to the trustpoint that was configured earlier.

```
Router (ca-certificate-map)# crypto pki trustpoint home-office
Router (ca-trustpoint)# match certificate central-site skip revocation-check
Router (ca-trustpoint)# exit
Router (config)# exit
```

The configuration is checked (most of configuration is not shown).

```
Router# write term
!Many lines left out
.
.
.
crypto pki trustpoint home-office
```

```
enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll serial-number none
fqdn none
ip-address none
subject-name o=Home Office Inc,cn=Branch 1 revocation-check crl
match certificate central-site skip revocation-check
!
!
crypto pki certificate map central-site 10
issuer-name co cn = Central Certificate Authority, ou = Home Office Inc subject-name eq cn
= central vpn gateway, o = home office inc
!many lines left out
```

Note that the issuer-name and subject-name lines have been reformatted to make them consistent for later matching with the certificate of the peer.

If the branch office is checking the AAA, the trustpoint will have lines similar to the following:

```
crypto pki trustpoint home-office auth list allow_list
auth user subj commonname
```

After the certificate map has been defined as was done above, the following command is added to the trustpoint to skip AAA checking for the central site hub.

```
match certificate central-site skip authorization-check
```

In both cases, the branch site router has to establish an IPSec tunnel to the central site to check CRLs or to contact the AAA server. However, without the **match certificate** command and **central-site skip authorization-check (argument and keyword)**, the branch office cannot establish the tunnel until it has checked the CRL or the AAA server. (The tunnel will not be established unless the **match certificate** command and **central-site skip authorization-check** argument and keyword are used.)

The **match certificate** command and **allow expired-certificate** keyword would be used at the central site if the router at a branch site had an expired certificate and it had to establish a tunnel to the central site to renew its certificate.

Trustpoint on the Central Site Router

```
crypto pki trustpoint VPN-GW
enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll
serial-number none
fqdn none
ip-address none
subject-name o=Home Office Inc,cn=Central VPN Gateway
revocation-check crl
```

Trustpoint on the Branch 1 Site Router

```
Router# show crypto ca certificate
Certificate
Status: Available
Certificate Serial Number: 2F62BE1400000000CA0 Certificate Usage: General Purpose
Issuer:
cn=Central Certificate Authority o=Home Office Inc
Subject:
Name: Branch 1 Site cn=Branch 1 Site o=Home Office Inc
CRL Distribution Points:
  http://ca.home-office.com/CertEnroll/home-office.crl Validity Date:
start date: 00:43:26 GMT Sep 26 2003
end date: 00:53:26 GMT Oct 3 2003
renew date: 00:00:00 GMT Jan 1 1970
Associated Trustpoints: home-office
CA Certificate
Status: Available
Certificate Serial Number: 1244325DE0369880465F977A18F61CA8
Certificate Usage: Signature
Issuer:
cn=Central Certificate Authority o=Home Office Inc
Subject:
cn=Central Certificate Authority o=Home Office Inc
CRL Distribution Points:
  http://ca.home-office.com/CertEnroll/home-office.crl
Validity Date:
start date: 22:19:29 GMT Oct 31 2002
end date: 22:27:27 GMT Oct 31 2017
Associated Trustpoints: home-office
A certificate map is entered on the central site router.
```

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router (config)# crypto pki certificate map branch1 10
Router (ca-certificate-map)# issuer-name co cn=Central Certificate Authority, ou=Home
Office Inc
```

!The above line wrapped but should be part of the line above it.
 Router (ca-certificate-map)# **subject-name eq cn=Brahcn 1 Site,o=home office inc**
 The certificate map is added to the trustpoint.

```
Router (ca-certificate-map)# crypto pki trustpoint VPN-GW
Router (ca-trustpoint)# match certificate branch1 allow expired-certificate
Router (ca-trustpoint)# exit
Router (config)# exit
```

The configuration should be checked (most of the configuration is not shown).

```
Router# write term
!many lines left out
crypto pki trustpoint VPN-GW
enrollment url http://ca.home-office.com:80/certsrv/mscep/mscep.dll serial-number none
fqdn none
ip-address none
subject-name o=Home Office Inc,cn=Central VPN Gateway revocation-check crl
match certificate branch1 allow expired-certificate
!
!
crypto pki certificate map central-site 10
issuer-name co cn = Central Certificate Authority, ou = Home Office Inc
subject-name eq cn = central vpn gateway, o = home office inc
! many lines left out
```

The **match certificate** command and **branch1 allow expired-certificate** (argument and keyword) and the certificate map should be removed as soon as the branch router has a new certificate.

Configuring Certificate Authorization and Revocation Settings Examples

This section contains the following configuration examples that can be used when specifying a CRL cache control setting or certificate serial number session control:

Configuring Certificate Authorization and Revocation Settings Examples

This section contains the following configuration examples that can be used when specifying a CRL cache control setting or certificate serial number session control:

Configuring CRL Cache Control

The following example shows how to disable CRL caching for all CRLs associated with the CA1 trustpoint:

```
crypto pki trustpoint CA1 enrollment url http://CA1:80 ip-address FastEthernet0/0 crl
query ldap://ldap_CA1 revocation-check crl
crl-cache none
The current CRL is still cached immediately after executing the example configuration
shown above:
Router# show crypto pki crls
CRL Issuer Name:
cn=name Cert Manager,ou=pki,o=example.com,c=US
```



```
LastUpdate: 18:57:42 GMT Nov 26 2005
NextUpdate: 22:57:42 GMT Nov 26 2005
Retrieved from CRL Distribution Point:
ldap://ldap.example.com/CN=name Cert Manager,O=example.com
```

When the current CRL expires, a new CRL is then downloaded to the router at the next update. The **crl-cache none** command takes effect and all CRLs for the trustpoint are no longer cached; caching is disabled. You can verify that no CRL is cached by executing the **show crypto pki crls** command. No output will be shown because there are no CRLs cached.

The following example shows how to configure the maximum lifetime of 2 minutes for all CRLs associated with the CA1 trustpoint:

```
crypto pki trustpoint CA1
enrollment url http://CA1:80
ip-address FastEthernet0/0
crl query ldap://ldap_CA1
revocation-check crl
crl-cache delete-after 2
```

The current CRL is still cached immediately after executing the example configuration above for setting the maximum lifetime of a CRL:

```
Router# show crypto pki crls
CRL Issuer Name:
cn=name Cert Manager,ou=pki,o=example.com,c=US
LastUpdate: 18:57:42 GMT Nov 26 2005
NextUpdate: 22:57:42 GMT Nov 26 2005
Retrieved from CRL Distribution Point:
ldap://ldap.example.com/CN=name Cert Manager,O=example.com
When the current CRL expires, a new CRL is downloaded to the router at the next update and
the crl-cache delete-after command takes effect. This newly cached CRL and all subsequent
CRLs will be deleted after a maximum lifetime of 2 minutes.
You can verify that the CRL will be cached for 2 minutes by executing the show crypto pki
crls
command. Note that the NextUpdate time is 2 minutes after the LastUpdate time.
Router# show crypto pki crls
```

```
CRL Issuer Name:
cn=name Cert Manager,ou=pki,o=example.com,c=US
LastUpdate: 22:57:42 GMT Nov 26 2005
NextUpdate: 22:59:42 GMT Nov 26 2005 Retrieved from CRL Distribution Point:
ldap://ldap.example.com/CN=name Cert Manager,O=example.com
```

Configuring Certificate Serial Number Session Control

The following example shows the configuration of certificate serial number session control using a certificate map for the CA1 trustpoint:

```
crypto pki trustpoint CA1 enrollment url http://CA1 chain-validation stop
crl query ldap://ldap_server revocation-check crl
match certificate crl
!
crypto pki certificate map crl 10 serial-number co 279d
```



Note

If the match-criteria value is set to **eq** (equal) instead of **co** (contains), the serial number must match the certificate map serial number exactly, including any spaces.

The following example shows the configuration of certificate serial number session control using AAA attributes. In this case, all valid certificates will be accepted if the certificate does not have the serial number “4ACA.”

```
crypto pki trustpoint CA1
enrollment url http://CA1 ip-address FastEthernet0/0
crl query ldap://ldap_CA1
revocation-check crl
aaa new-model
!
aaa attribute list crl
attribute-type aaa-cert-serial-not 4ACA
```

The server log shows that the certificate with the serial number ?ACA?was rejected. The certificate rejection is shown using exclamation points.

```
...
Dec 3 04:24:39.051: CRYPTO_PKI: Trust-Point CA1 picked up
Dec 3 04:24:39.051: CRYPTO_PKI: locked trustpoint CA1, refcount is 1
Dec 3 04:24:39.051: CRYPTO_PKI: unlocked trustpoint CA1, refcount is 0
Dec 3 04:24:39.051: CRYPTO_PKI: locked trustpoint CA1, refcount is 1
Dec 3 04:24:39.135: CRYPTO_PKI: validation path has 1 certs
Dec 3 04:24:39.135: CRYPTO_PKI: Found a issuer match
Dec 3 04:24:39.135: CRYPTO_PKI: Using CA1 to validate certificate
Dec 3 04:24:39.135: CRYPTO_PKI: Certificate validated without revocation check
Dec 3 04:24:39.135: CRYPTO_PKI: Selected AAA username: 'PKIAAAA'
Dec 3 04:24:39.135: CRYPTO_PKI: Anticipate checking AAA list:'CRL'
Dec 3 04:24:39.135: CRYPTO_PKI_AAA: checking AAA authorization (CRL, PKIAAAA-L1, <all>)
Dec 3 04:24:39.135: CRYPTO_PKI_AAA: pre-authorization chain validation status (0x4)
Dec 3 04:24:39.135: AAA/BIND(00000021): Bind i/f
Dec 3 04:24:39.135: AAA/AUTHOR (0x21): Pick method list 'CRL'
...
Dec 3 04:24:39.175: CRYPTO_PKI_AAA: reply attribute ("cert-application" = "all")
Dec 3 04:24:39.175: CRYPTO_PKI_AAA: reply attribute ("cert-trustpoint" = "CA1")
!
Dec 3 04:24:39.175: CRYPTO_PKI_AAA: reply attribute ("cert-serial-not" = "4ACA")
Dec 3 04:24:39.175: CRYPTO_PKI_AAA: cert-serial doesn't match ("4ACA" != "4ACA")
!
Dec 3 04:24:39.175: CRYPTO_PKI_AAA: post-authorization chain validation status (0x7)
!
Dec 3 04:24:39.175: CRYPTO_PKI: AAA authorization for list 'CRL', and user 'PKIAAAA'
failed.
Dec 3 04:24:39.175: CRYPTO_PKI: chain cert was anchored to trustpoint CA1, and chain
validation result was: CRYPTO_PKI_CERT_NOT_AUTHORIZED
!
Dec 3 04:24:39.175: %CRYPTO-5-IKMP_INVALID_CERT: Certificate received from 192.0.2.43 is
bad:
certificate invalid
Dec 3 04:24:39.175: %CRYPTO-6-IKMP_MODE_FAILURE: Processing of Main mode failed with peer
at 192.0.2.43
```

Configuring Certificate Chain Validation Examples

This section contains the following configuration examples that can be used to specify the level of certificate chain processing for your device certificates:

Configuring Certificate Chain Validation from Peer to Root CA

In the following configuration example, all of the certificates will be validated--the peer, SubCA11, SubCA1, and RootCA certificates.

```
crypto pki trustpoint RootCA enrollment terminal
chain-validation stop revocation-check none rsakeypair RootCA
crypto pki trustpoint SubCA1 enrollment terminal
chain-validation continue RootCA revocation-check none
rsakeypair SubCA1
crypto pki trustpoint SubCA11 enrollment terminal
chain-validation continue SubCA1 revocation-check none
rsakeypair SubCA11
```

Configuring Certificate Chain Validation from Peer to Subordinate CA

In the following configuration example, the following certificates will be validated--the peer and SubCA1 certificates.

```
crypto pki trustpoint RootCA
enrollment terminal
chain-validation stop
revocation-check none
rsakeypair RootCA
crypto pki trustpoint SubCA1
enrollment terminal
chain-validation continue RootCA
revocation-check none
rsakeypair SubCA1
crypto pki trustpoint SubCA11
enrollment terminal
chain-validation continue SubCA1 r
evocation-check none
rsakeypair SubCA11
```

Configuring Certificate Chain Validation Through a Gap

In the following configuration example, SubCA1 is not in the configured Cisco IOS hierarchy but is expected to have been supplied in the certificate chain presented by the peer.

If the peer supplies the SubCA1 certificate in the presented certificate chain, the following certificates will be validated--the peer, SubCA11, and SubCA1 certificates.

If the peer does not supply the SubCA1 certificate in the presented certificate chain, the chain validation will fail.

```
crypto pki trustpoint RootCA
enrollment terminal
chain-validation stop
revocation-check none
rsakeypair RootCA
crypto pki trustpoint SubCA11
enrollment terminal
chain-validation continue RootCA
revocation-check none
rsakeypair SubCA11
```

Additional References

Related Documents

Related Topic	Document Title
PKI commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS Security Command Reference
Overview of PKI, including RSA keys, certificate enrollment, and CAs	“Cisco IOS PKI Overview: Understanding and Planning a PKI” module
RSA key generation and deployment	“Deploying RSA Keys Within a PKI” module
Certificate enrollment: supported methods, enrollment profiles, configuration tasks	“Configuring Certificate Enrollment for a PKI” module
Cisco IOS certificate server overview information and configuration tasks	“Configuring and Managing a Cisco IOS Certificate Server for PKI Deployment ” module
Recommended cryptographic algorithms	Next Generation Encryption



Support for IPv6

This chapter lists the IP version 6 (IPv6) features supported on the switch.

The IPv6 for Cisco IOS software feature documentation provides implementation and command reference information for IPv6 features supported in the Cisco IOS software. Not all IPv6 features are supported. We recommend that you read this entire chapter before reading the other IPv6 for Cisco IOS software feature documentation.

This chapter consists of these sections:

- [Finding Feature Information, page 64-1](#)
- [About IPv6, page 64-1](#)
- [IPv6 Default States, page 64-7](#)



Note

For Cisco IOS IPv6 Configuration Guides, see:

- [IPv6 Configuration Library, Cisco IOS Release 15E](#)
- [IPv6 Configuration Guide Library, Cisco IOS XE Release 3E](#)

For complete syntax and usage information for the switch commands used in this chapter, see:

- The [Cisco IOS IPv6 Command Reference](#)
 - The [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).
-

Finding Feature Information

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to <http://tools.cisco.com/ITDIT/CFN/>. An account on Cisco.com is not required.

About IPv6

IPv6 provides services such as end-to-end security, quality of service (QoS), and globally unique addresses. The IPv6 address space reduces the need for private addresses and Network Address Translation (NAT) processing by border routers at network edges.

For information about how Cisco Systems implements IPv6, go to this URL:

http://www.cisco.com/en/US/products/ps6553/products_ios_technology_home.html

This section describes the features that are supported for IPv6:

- [IPv6 Addressing and Basic Connectivity, page 65-1](#)
- [DHCP, page 64-2](#)
- [Security, page 64-2](#)
- [First-Hop Security, page 64-3](#)
- [QoS, page 64-3](#)
- [Management, page 64-3](#)
- [Multicast, page 64-4](#)
- [Static Routes, page 64-4](#)
- [First-Hop Redundancy Protocols, page 64-5](#)
- [Unicast Routing, page 64-5](#)
- [Tunneling, page 64-7](#)

DHCP

The following DHCP features are supported for IPv6 on the Catalyst 4500 series switch:

- Relay agent
- Relay agent notification for prefix delegation
- Reload persistent interface ID option
- Ethernet remote ID option
- Stateless auto-configuration

You can find information about these features at this location:

[IP Addressing: DHCP Configuration Guide, Cisco IOS Release 15E](#)

[IP Addressing: DHCP Configuration Guide, Cisco IOS XE Release 3E](#)

Security

The following security features are supported for IPv6 on the Catalyst 4500 series switch:

- Secure Shell (SSH) support over IPv6
- Traffic filters
- standard access control lists (ACL)
- extended access control lists
- ACL accounting
- ACL addressing
- ACL DSCP
- ACL flags

- ACL flows
- ACL fragments
- ACL ICMP codes
- ACL logging
- ACL protocols

You can find information about these features at this location:

[Security Configuration Guide: Access Control Lists, Cisco IOS Release 15E](#)

[Security Configuration Guide: Access Control Lists, Cisco IOS XE Release 3E](#)

First-Hop Security

The following First-Hop Security (FHS) features are supported for IPv6 on the Catalyst 4500-E and -X series switch:

- SISF-Based Device Tracking
- IPv6 RA Guard
- IPv6 Source Guard and Prefix Guard
- IPv6 Snooping
- DHCPv6 Guard
- IPv6 Neighbor Discovery Multicast Suppress
- IPv6 Destination Guard
- IPv6 RFCs

You can find information about these features at this location:

[IPv6 First-Hop Security Configuration Guide, Cisco IOS Release 15E](#)

[IPv6 First-Hop Security Configuration Guide, Cisco IOS Release XE 3E](#)

QoS

The following QoS features are supported for IPv6 on the Catalyst 4500 series switch:

- MQC packet classification
- MQC traffic shaping
- MQC traffic policing
- MQC packet marking and remarking
- Queueing

You can find information about these features at this location:

[QoS: Classification Configuration Guide, Cisco IOS XE Release 3S](#)

Management

The following management features are supported for IPv6 on the Catalyst 4500 series switch:

- Ping
- Syslog
- Netconf support
- SNMP
- SOAP
- HTTP(s)

You can find information about these features at this location:

[Network Management Configuration Guide Library, Cisco IOS Release 15E](#)

[Network Management Configuration Guide Library, Cisco IOS XE Release 3E](#)

Multicast

Traditional IP communication allows a host to send packets to a single host (unicast transmission) or to all hosts (broadcast transmission). IPv6 multicast, allows a host to send a single data stream to a subset of all hosts (group transmission) simultaneously.

The following multicast features are supported for IPv6 on the Catalyst 4500 series switch:

- Multicast Listener Discovery (MLD) protocol, versions 1 and 2

You can find information about IPv6 MLD Snooping at this location:

<http://www.cisco.com/en/US/docs/switches/lan/catalyst4500/12.2/53SG/configuration/mldsnoop.html>

- MLD access group
- MLD snooping
- Scope boundaries
- Protocol Independent Multicast (PIM) features:

A switch running the LAN Base license level supports PIM for IPv6. VRF support is not available at the LAN Base level.

- PIM embedded Rendezvous Point (RP) support
 - PIM Sparse Mode (PIM-SM)
 - PIM Source Specific Multicast (PIM-SSM)
- Static multicast routing (mroute)
- Explicit tracking of receivers
- Bootstrap routers (BSR)

You can find information about these features at this location:

[IP Multicast Configuration Guide Library, Cisco IOS XE Release 3E](#)

Static Routes

Networking devices forward packets using route information that is either manually configured or dynamically learned using a routing protocol. Static routes are manually configured and define an explicit path between two networking devices. Unlike a dynamic routing protocol, static routes are not

automatically updated and must be manually reconfigured if the network topology changes. The benefits of using static routes include security and resource efficiency. Static routes use less bandwidth than dynamic routing protocols and no CPU cycles are used to calculate and communicate routes. The main disadvantage to using static routes is the lack of automatic reconfiguration if the network topology changes.

Static routes can be redistributed into dynamic routing protocols but routes generated by dynamic routing protocols cannot be redistributed into the static routing table. No algorithm exists to prevent the configuration of routing loops that use static routes.

Static routes are useful for smaller networks with only one path to an outside network and to provide security for a larger network for certain types of traffic or links to other networks that need more control. In general, most networks use dynamic routing protocols to communicate between networking devices but may have one or two static routes configured for special cases.

You can find more information regarding static routes at:

[IP Routing: Protocol-Independent Configuration Guide, Cisco IOS Release 15E](#)

[IP Routing: Protocol-Independent Configuration Guide, Cisco IOS XE Release 3E](#)

[IP Routing: Protocol-Independent Configuration Guide, Cisco IOS XE Release 3S](#)

First-Hop Redundancy Protocols

IPv6 routing protocols ensure router-to-router resilience and failover. However, in situations in which the path between a host and the first-hop router fails, or the first-hop router itself fails, First-Hop Redundancy Protocols (FHRPs) ensure host-to-router resilience and failover.

The Hot Standby Router Protocol (HSRP) protects data traffic in case of a gateway failure.

You can find more information about First-Hop Redundancy Protocols at:

[First Hop Redundancy Protocols Configuration Guide, Cisco IOS Release 15E](#)

[First Hop Redundancy Protocols Configuration Guide, Cisco IOS XE Release 3E](#)

Unicast Routing

These sections describe the IPv6 unicast routing protocol features supported by the switch:

- [RIP, page 64-5](#)
- [OSPF, page 64-6](#)
- [EIGRP, page 64-6](#)
- [IS-IS, page 64-6](#)
- [Multiprotocol BGP, page 64-6](#)

RIP

A switch running the LAN Base license level supports the Routing Information Protocol (RIP) for IPv6. VRF support is not available at the LAN Base level.

RIP is a distance-vector protocol that uses hop count as a routing metric. It includes support for IPv6 addresses and prefixes and the all-RIP routers multicast group address FF02::9 as the destination address for RIP update messages.

You can find more about RIP at this location:

[IP Routing: RIP Configuration Guide, Cisco IOS XE Release 3E](#)

OSPF

A switch running the LAN Base license level supports the Open Shortest Path First (OSPF) for IPv6. VRF support is not available at the LAN Base level

OSPF is a link-state protocol for IP.

You can find more information about OSPF at this location:

[IP Routing: OSPF Configuration Guide, Cisco IOS Release 15E](#)

[IP Routing: OSPF Configuration Guide, Cisco IOS XE Release 3E](#)

EIGRP

The switch running the IP-services feature set supports Enhanced Interior Gateway Routing Protocol (EIGRP) for IPv6. It is configured on the interfaces on which it runs and does not require a global IPv6 address.

Before running, an instance of EIGRP IPv6 requires an implicit or explicit router ID. An implicit router ID is derived from a local IPv4 address, so any IPv4 node always has an available router ID. However, EIGRP IPv6 might be running in a network with only IPv6 nodes and therefore might not have an available IPv4 router ID.

You can find more information about EIGRP at this location:

[IP Routing: EIGRP Configuration Guide, Cisco IOS Release 15E](#)

[IP Routing: EIGRP Configuration Guide, Cisco IOS XE Release 3E](#)

IS-IS

Intermediate System-to-Intermediate System (IS-IS) is an Interior Gateway Protocol (IGP) that advertises link-state information throughout the network to create a picture of the network topology. IS-IS is an Open Systems Interconnection (OSI) hierarchical routing protocol that designates an intermediate system as a Level 1 or Level 2 device. Level 2 devices route between Level 1 areas to create an intradomain routing backbone. Integrated IS-IS uses a single routing algorithm to support several network address families, such as IPv6, IPv4, and OSI.

You can find more information about Is-IS at this location:

[IP Routing: ISIS Configuration Guide, Cisco IOS XE Release 3E](#)

Multiprotocol BGP

Multiprotocol Border Gateway Protocol (BGP) is an Exterior Gateway Protocol (EGP) used mainly to connect separate routing domains that contain independent routing policies (autonomous systems). Connecting to a service provider for access to the Internet is a common use for BGP. BGP can also be used within an autonomous system, which is referred to as internal BGP (iBGP). Multiprotocol BGP is an enhanced BGP that carries routing information for multiple network layer protocol address families, for example, IPv6 address family and for IP multicast routes. All BGP commands and routing policy capabilities can be used with multiprotocol BGP.

You can find more information about multiprotocol BGP at this location:

IP Routing: [BGP Configuration Guide, Cisco IOS Release 15E](#)

IP Routing: [BGP Configuration Guide, Cisco IOS XE Release 3E](#)

Tunneling

The following tunneling features are supported for IPv6 on the Catalyst 4500 series switch:

- Automatic 6to4
- ISATAP
- Configured tunnels

**Note**

Tunneling is not supported in hardware but is supported in software.

You can find information about these features at this location:

IPv6 Implementation Guide, Cisco IOS XE Release 3S > [Implementing Tunneling for IPv6](#)

IPv6 Default States

[Table 64-1](#) shows the default states of IPv6 configuration.

Table 64-1 **Default IPv6 Configuration**

Feature	Default Setting
IPv6 routing	Disabled globally and on all interfaces
CEFv6 or dCEFv6	Disabled (IPv4 CEF and dCEF are enabled by default) Note When IPv6 routing is enabled, CEFv6 and dCEF6 are automatically enabled.
IPv6 addresses	None configured



IPv6 Addressing and Basic Connectivity

Internet Protocol version 6 (IPv6) expands the number of network address bits from 32 bits (in IPv4) to 128 bits, which provides more than enough globally unique IP addresses for every networked device on the planet. The unlimited address space provided by IPv6 allows Cisco to deliver more and newer applications and services with reliability, improved user experience, and increased security.

Implementing basic IPv6 connectivity in the Cisco software consists of assigning IPv6 addresses to individual device interfaces. IPv6 traffic forwarding can be enabled globally, and Cisco Express Forwarding switching for IPv6 can also be enabled. The user can enhance basic connectivity functionality by configuring support for AAAA record types in the Domain Name System (DNS) name-to-address and address-to-name lookup processes, and by managing IPv6 neighbor discovery.



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).



Note

The switch supports the following features:

IPv6 address types: Anycast, IPv6 default router preferences, IPv6 MTU path discovery, Neighbor discovery duplicate address detection, Cisco Discovery Protocol — IPv6 address family support for neighbor information, ICMPv6 redirect, ICMP rate limiting, DNS lookups over an IPv6 transport, uRPF, ICMPv6

- [About IPv6 Addressing and Basic Connectivity, page 65-1](#)
- [Configuring IPv6 Addressing and Basic Connectivity, page 65-13](#)
- [Configuration Examples for IPv6 Addressing and Basic Connectivity, page 65-16](#)

About IPv6 Addressing and Basic Connectivity

- [IPv6 for Cisco Software, page 65-2](#)
- [Large IPv6 Address Space for Unique Addresses, page 65-2](#)
- [IPv6 Address Formats, page 65-3](#)
- [IPv6 Address Output Display, page 65-4](#)

- [IPv6 Address Output Display, page 65-4](#)
- [Simplified IPv6 Packet Header, page 65-4](#)
- [Path MTU Discovery for IPv6, page 65-8](#)
- [IPv6 Prefix Aggregation, page 65-8](#)
- [IPv6 Site Multihoming, page 65-9](#)
- [IPv6 Data Links, page 65-9](#)
- [Dual IPv4 and IPv6 Protocol Stacks, page 65-9](#)
- [Cisco Discovery Protocol IPv6 Address Support, page 65-10](#)
- [ICMP for IPv6, page 65-11](#)
- [IPv6 Neighbor Discovery, page 65-11](#)
- [IPv6 Neighbor Solicitation Message, page 65-11](#)

IPv6 for Cisco Software

IPv6, formerly named IPng (next generation), is the latest version of the Internet Protocol (IP). IP is a packet-based protocol used to exchange data, voice, and video traffic over digital networks. IPv6 was proposed when it became clear that the 32-bit addressing scheme of IP version 4 (IPv4) was inadequate to meet the demands of Internet growth. After extensive discussion it was decided to base IPng on IP but add a much larger address space and improvements such as a simplified main header and extension headers. IPv6 is described initially in RFC 2460, Internet Protocol, Version 6 (IPv6) Specification, issued by the Internet Engineering Task Force (IETF). Further RFCs describe the architecture and services supported by IPv6.

The architecture of IPv6 has been designed to allow existing IPv4 users to transition easily to IPv6 while providing services such as end-to-end security, quality of service (QoS), and globally unique addresses. The larger IPv6 address space allows networks to scale and provide global reachability. The simplified IPv6 packet header format handles packets more efficiently. IPv6 prefix aggregation, simplified network renumbering, and IPv6 site multihoming capabilities provide an IPv6 addressing hierarchy that allows for more efficient routing. IPv6 supports widely deployed routing protocols such as Integrated Intermediate System-to-Intermediate System (IS-IS), Open Shortest Path First (OSPF) for IPv6, and multiprotocol Border Gateway Protocol (BGP). Other available features include stateless autoconfiguration and an increased number of multicast addresses.

Large IPv6 Address Space for Unique Addresses

The primary motivation for IPv6 is the need to meet the demand for globally unique IP addresses. IPv6 quadruples the number of network address bits from 32 bits (in IPv4) to 128 bits, which provides more than enough globally unique IP addresses for every networked device on the planet. By being globally unique, IPv6 addresses inherently enable global reachability and end-to-end security for networked devices, functionality that is crucial to the applications and services that are driving the demand for the addresses. Additionally, the flexibility of the IPv6 address space reduces the need for private addresses; therefore, IPv6 enables new application protocols that do not require special processing by border devices at the edge of networks.

IPv6 Address Formats

IPv6 addresses are represented as a series of 16-bit hexadecimal fields separated by colons (:) in the format: x:x:x:x:x:x:x. Following are two examples of IPv6 addresses:

```
2001:DB8:7654:3210:FEDC:BA98:7654:3210
```

```
2001:DB8:0:0:8:800:200C:417A
```

IPv6 addresses commonly contain successive hexadecimal fields of zeros. Two colons (::) may be used to compress successive hexadecimal fields of zeros at the beginning, middle, or end of an IPv6 address (the colons represent successive hexadecimal fields of zeros). The table below lists compressed IPv6 address formats.

A double colon may be used as part of the ipv6-address argument when consecutive 16-bit values are denoted as zero. You can configure multiple IPv6 addresses per interfaces, but only one link-local address.



Note Two colons (::) can be used only once in an IPv6 address to represent the longest successive hexadecimal fields of zeros. The hexadecimal letters in IPv6 addresses are not case-sensitive.

Table 65-1 Compressed IPv6 Address Formats

IPv6 Address Type	Preferred Format	Compressed Format
Unicast	2001:0:0:0:DB8:800:200C:417A	2001::DB8:800:200C:417A
Multicast	FF01:0:0:0:0:0:101	FF01::101
Loopback	0:0:0:0:0:0:1	::1
Unspecified	0:0:0:0:0:0:0	::

The loopback address listed in the table above may be used by a node to send an IPv6 packet to itself. The loopback address in IPv6 functions the same as the loopback address in IPv4 (127.0.0.1).



Note The IPv6 loopback address cannot be assigned to a physical interface. A packet that has the IPv6 loopback address as its source or destination address must remain within the node that created the packet. IPv6 devices do not forward packets that have the IPv6 loopback address as their source or destination address.

The unspecified address listed in the table above indicates the absence of an IPv6 address. For example, a newly initialized node on an IPv6 network may use the unspecified address as the source address in its packets until it receives its IPv6 address.



Note The IPv6 unspecified address cannot be assigned to an interface. The unspecified IPv6 addresses must not be used as destination addresses in IPv6 packets or the IPv6 routing header.

An IPv6 address prefix, in the format ipv6-prefix/prefix-length, can be used to represent bit-wise contiguous blocks of the entire address space. The ipv6-prefix must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons. The prefix length is a decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). For example, 2001:DB8:8086:6502::/32 is a valid IPv6 prefix.

IPv6 Address Output Display

When IPv6 or IPv4 command output displays an IPv6 address, a long IPv6 address can overflow into neighboring fields, causing the output to be difficult to read. The output fields were designed to work with the longest possible IPv4 address, which has 15 characters; IPv6 addresses can be up to 39 characters long. The following scheme has been adopted in IPv4 and IPv6 commands to allow the appropriate length of IPv6 address to be displayed and move the following fields to the next line, if necessary. The fields that are moved are kept in alignment with the header row.

The following example displays eight connections. The first six connections feature IPv6 addresses; the last two connections feature IPv4 addresses.

Switch#	where					
Conn	Host	Address	Byte	Idle	Conn	Name
1	test5	2001:DB8:3333:4::5	6	24		test5
2	test4	2001:DB8:3333:44::5				
			6	24		test4
3	2001:DB8:3333:4::5	2001:DB8:3333:4::5	6	24		2001:DB8:3333:4::5
4	2001:DB8:3333:44::5					
		2001:DB8:3333:44::5				
			6	23		2001:DB8:3333:44::5
5	2001:DB8:3000:4000:5000:6000:7000:8001					
		2001:DB8:3000:4000:5000:6000:7000:8001				
			6	20		2001:DB8:3000:4000:5000:6000:
6	2001:DB8:1::1	2001:DB8:1::1	0	1		2001:DB8:1::1
7	10.1.9.1	10.1.9.1	0	0		10.1.9.1
8	10.222.111.222	10.222.111.222	0	0		10.222.111.222

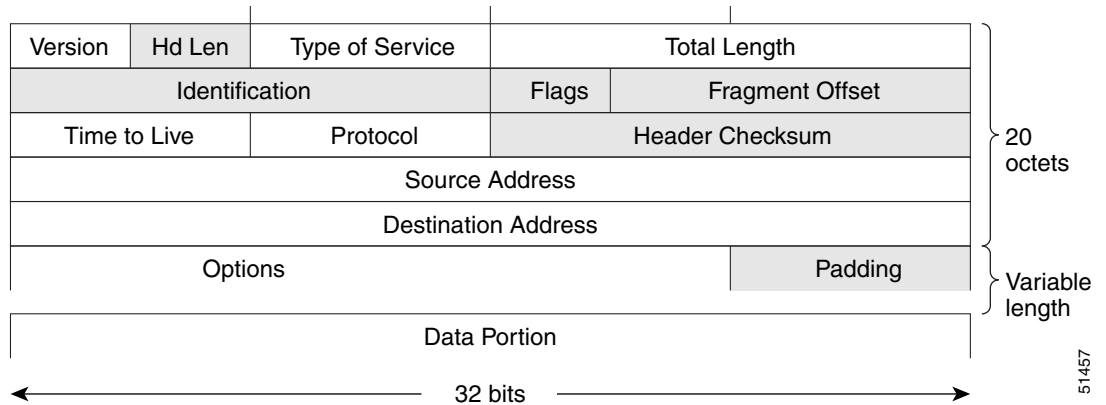
Connection 1 contains an IPv6 address that uses the maximum address length in the address field. Connection 2 shows the IPv6 address overflowing the address field and the following fields moved to the next line, but in alignment with the appropriate headers. Connection 3 contains an IPv6 address that fills the maximum length of the hostname and address fields without wrapping any lines. Connection 4 shows the effect of both the hostname and address fields containing a long IPv6 address. The output is shown over three lines keeping the correct heading alignment. Connection 5 displays a similar effect as connection 4 with a very long IPv6 address in the hostname and address fields. Note that the connection name field is actually truncated. Connection 6 displays a very short IPv6 address that does not require any change in the display. Connections 7 and 8 display short and long IPv4 addresses.



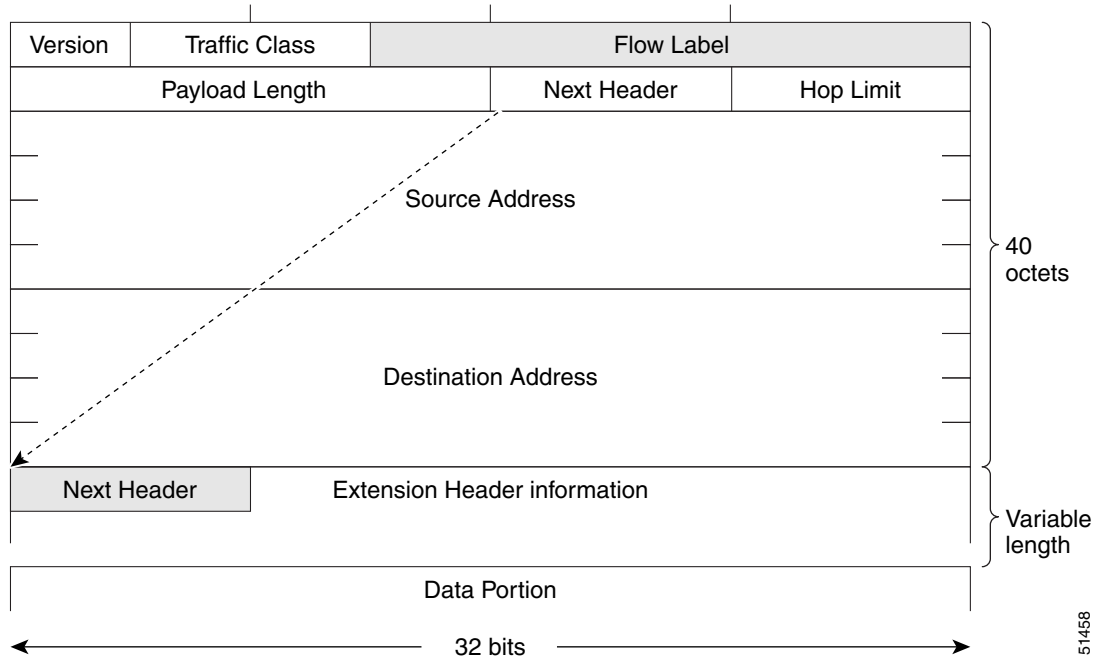
Note The IPv6 address output display applies to all commands that display IPv6 addresses.

Simplified IPv6 Packet Header

The basic IPv4 packet header has 12 fields with a total size of 20 octets (160 bits) (see the figure below). The 12 fields may be followed by an Options field, which is followed by a data portion that is usually the transport-layer packet. The variable length of the Options field adds to the total size of the IPv4 packet header. The shaded fields of the IPv4 packet header shown in the figure below are not included in the IPv6 packet header.

Figure 65-1 IPv4 Packet Header Format

The basic IPv6 packet header has 8 fields with a total size of 40 octets (320 bits) (see the figure below). Fields were removed from the IPv6 header because, in IPv6, fragmentation is not handled by devices and checksums at the network layer are not used. Instead, fragmentation in IPv6 is handled by the source of a packet and checksums at the data link layer and transport layer are used. (In IPv4, the UDP transport layer uses an optional checksum. In IPv6, use of the UDP checksum is required to check the integrity of the inner packet.) Additionally, the basic IPv6 packet header and Options field are aligned to 64 bits, which can facilitate the processing of IPv6 packets.

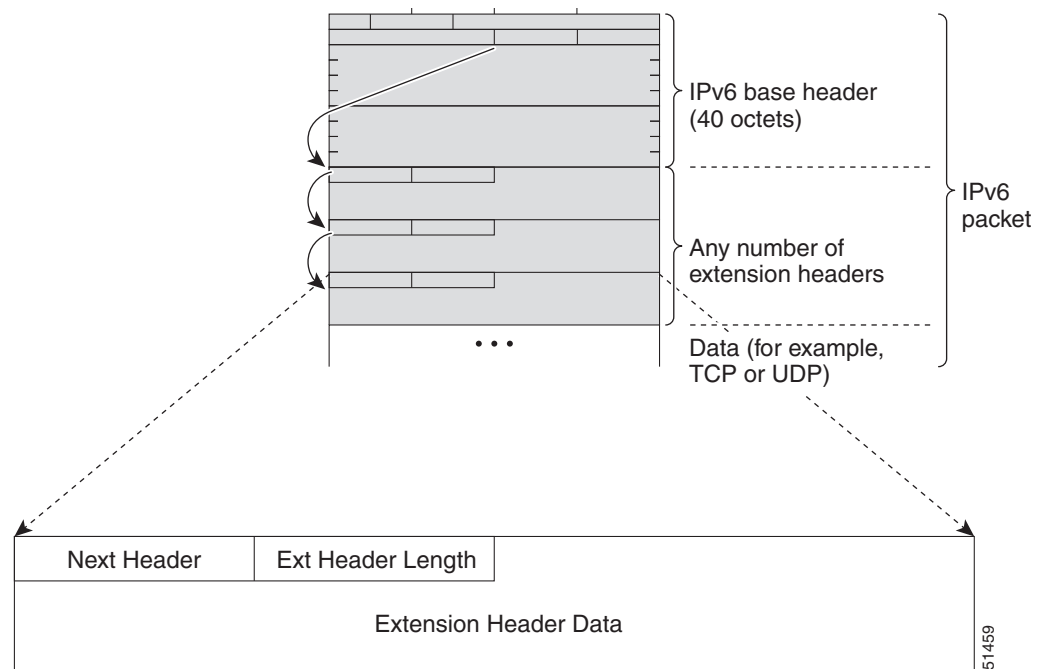
Figure 65-2 IPv6 Packet Header Format

The table below lists the fields in the basic IPv6 packet header.

Table 65-1 Basic IPv6 Packet Header Fields

Field	Description
Version	Similar to the Version field in the IPv4 packet header, except that the field lists number 6 for IPv6 instead of number 4 for IPv4.
Traffic Class	Similar to the Type of Service field in the IPv4 packet header. The Traffic Class field tags packets with a traffic class that is used in differentiated services.
Flow Label	A new field in the IPv6 packet header. The Flow Label field tags packets with a specific flow that differentiates the packets at the network layer.
Payload Length	Similar to the Total Length field in the IPv4 packet header. The Payload Length field indicates the total length of the data portion of the packet.
Next Header	Similar to the Protocol field in the IPv4 packet header. The value of the Next Header field determines the type of information following the basic IPv6 header. The type of information following the basic IPv6 header can be a transport-layer packet, for example, a TCP or UDP packet, or an Extension Header, as shown in the figure immediately above.
Hop Limit	Similar to the Time to Live field in the IPv4 packet header. The value of the Hop Limit field specifies the maximum number of devices that an IPv6 packet can pass through before the packet is considered invalid. Each device decrements the value by one. Because no checksum is in the IPv6 header, the device can decrement the value without needing to recalculate the checksum, which saves processing resources.
Source Address	Similar to the Source Address field in the IPv4 packet header, except that the field contains a 128-bit source address for IPv6 instead of a 32-bit source address for IPv4.
Destination Address	Similar to the Destination Address field in the IPv4 packet header, except that the field contains a 128-bit destination address for IPv6 instead of a 32-bit destination address for IPv4.

Following the eight fields of the basic IPv6 packet header are optional extension headers and the data portion of the packet. If present, each extension header is aligned to 64 bits. There is no fixed number of extension headers in an IPv6 packet. The extension headers form a chain of headers. Each extension header is identified by the Next Header field of the previous header. Typically, the final extension header has a Next Header field of a transport-layer protocol, such as TCP or UDP. The figure below shows the IPv6 extension header format.

Figure 65-3 IPv6 Extension Header Format

The table below lists the extension header types and their Next Header field values.

Table 65-2 IPv6 Extension Header Types

Header Type	Next Header Value	Description
Hop-by-hop options header	0	This header is processed by all hops in the path of a packet. When present, the hop-by-hop options header always follows immediately after the basic IPv6 packet header.
Destination options header	60	The destination options header can follow any hop-by-hop options header, in which case the destination options header is processed at the final destination and also at each visited address specified by a routing header. Alternatively, the destination options header can follow any Encapsulating Security Payload (ESP) header, in which case the destination options header is processed only at the final destination.
Routing header	43	The routing header is used for source routing.
Fragment header	44	The fragment header is used when a source must fragment a packet that is larger than the maximum transmission unit (MTU) for the path between itself and a destination. The Fragment header is used in each fragmented packet.
Authentication header and ESP header	51 50	The Authentication header and the ESP header are used within IP Security Protocol (IPsec) to provide authentication, integrity, and confidentiality of a packet. These headers are identical for both IPv4 and IPv6.

Table 65-2 IPv6 Extension Header Types

Upper-layer headers	6 (TCP) 17 (UDP)	The upper-layer (transport) headers are the typical headers used inside a packet to transport the data. The two main transport protocols are TCP and UDP.
Mobility headers	135	Extension headers used by mobile nodes, correspondent nodes, and home agents in all messaging related to the creation and management of bindings.

Path MTU Discovery for IPv6

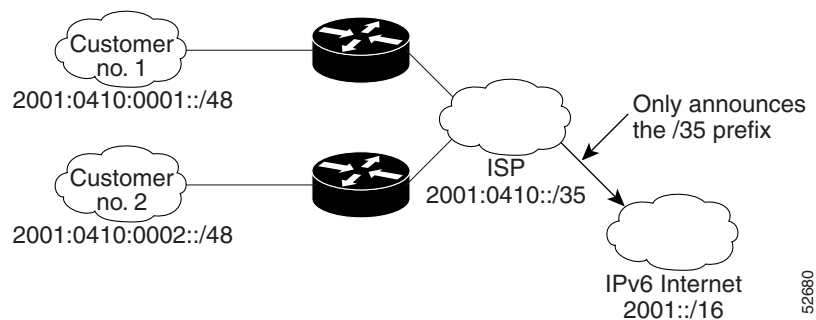
As in IPv4, you can use path MTU discovery in IPv6 to allow a host to dynamically discover and adjust to differences in the MTU size of every link along a data path. In IPv6, however, fragmentation is handled by the source of a packet when the path MTU of one link along a given data path is not large enough to accommodate the size of the packets. Having IPv6 hosts handle packet fragmentation saves IPv6 router processing resources and helps IPv6 networks run more efficiently. Once the path MTU is reduced by the arrival of an ICMP Too Big message, Cisco NX-OS retains the lower value. The connection does not increase the segment size to gauge the throughput.


Note

In IPv6, the minimum link MTU is 1280 octets. We recommend that you use an MTU value of 1500 octets for IPv6 links.

IPv6 Prefix Aggregation

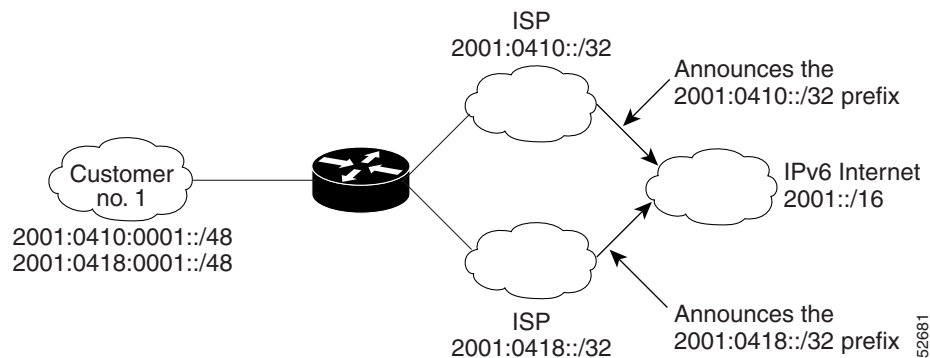
The aggregatable nature of the IPv6 address space enables an IPv6 addressing hierarchy. For example, an enterprise can subdivide a single IPv6 prefix from a service provider into multiple, longer prefixes for use within its internal network. Conversely, a service provider can aggregate all of the prefixes of its customers into a single, shorter prefix that the service provider can then advertise over the IPv6 internet (see the figure below).

Figure 65-4 IPv6 Prefix Aggregation

IPv6 Site Multihoming

Multiple IPv6 prefixes can be assigned to networks and hosts. Having multiple prefixes assigned to a network allows that network to connect easily to multiple ISPs without breaking the global routing table (see the figure below).

Figure 65-5 IPv6 Site Multihoming



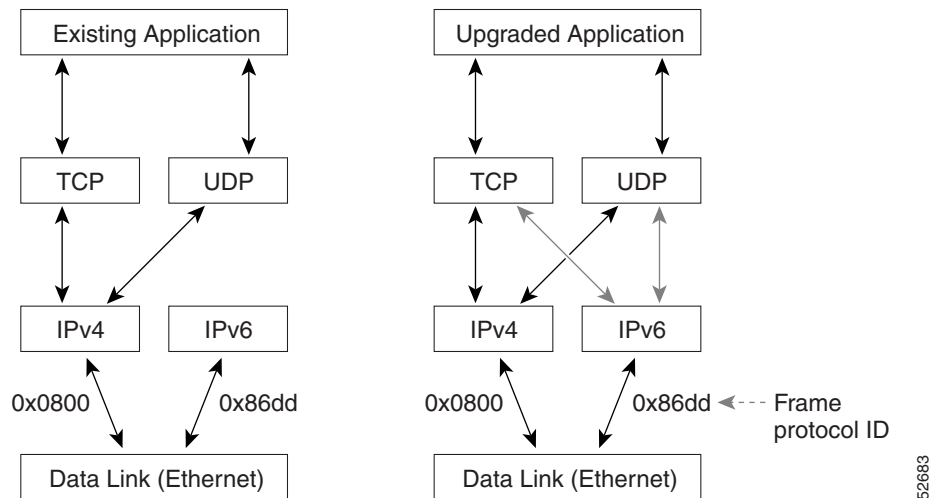
IPv6 Data Links

In IPv6 networks, a data link is a network sharing a particular link-local prefix. Data links are networks arbitrarily segmented by a network administrator in order to provide a multilevel, hierarchical routing structure while shielding the subnetwork from the addressing complexity of attached networks. The function of a subnetwork in IPv6 is similar to a subnetwork in IPv4. A subnetwork prefix is associated with one data link; multiple subnetwork prefixes may be assigned to the same data link.

The following data links are supported for IPv6: FDDI, Frame Relay PVC, Cisco High-Level Data Link Control (HDLC), PPP over Packet over SONET, ISDN, and serial interfaces.

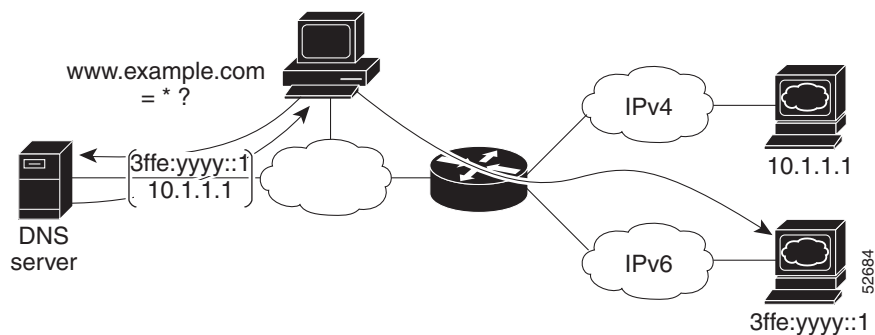
Dual IPv4 and IPv6 Protocol Stacks

The dual IPv4 and IPv6 protocol stack technique can be used to transition to IPv6. It enables gradual, one-by-one upgrades to applications running on nodes. Applications running on nodes are upgraded to make use of the IPv6 protocol stack. Applications that are not upgraded (for example, they support only the IPv4 protocol stack) can coexist with upgraded applications on a node. New and upgraded applications make use of both the IPv4 and IPv6 protocol stacks (see the figure below).

Figure 65-6 Dual IPv4 and IPv6 Protocol Stack Technique

One application program interface (API) supports both IPv4 and IPv6 addresses and DNS requests. An application can be upgraded to the new API and still use only the IPv4 protocol stack. The Cisco software supports the dual IPv4 and IPv6 protocol stack technique. When an interface is configured with both an IPv4 and an IPv6 address, the interface will forward both IPv4 and IPv6 traffic.

In the figure below, an application that supports dual IPv4 and IPv6 protocol stacks requests all available addresses for the destination hostname `www.example.com` from a DNS server. The DNS server replies with all available addresses (both IPv4 and IPv6 addresses) for `www.example.com`. The application chooses an address (in most cases, IPv6 addresses are the default choice), and connects the source node to the destination using the IPv6 protocol stack.

Figure 65-7 Dual IPv4 and IPv6 Protocol Stack Applications

Cisco Discovery Protocol IPv6 Address Support

The Cisco Discovery Protocol IPv6 address support for neighbor information feature adds the ability to transfer IPv6 addressing information between two Cisco devices. Cisco Discovery Protocol support for IPv6 addresses provides IPv6 information to network management products and troubleshooting tools.

ICMP for IPv6

You can use ICMP in IPv6 to provide information about the health of the network. ICMPv6, the version that works with IPv6, reports errors if packets cannot be processed correctly and sends informational messages about the status of the network. For example, if a router cannot forward a packet because it is too large to be sent out on another network, the router sends out an ICMPv6 message to the originating host. Additionally, ICMP packets in IPv6 are used in IPv6 neighbor discovery and path MTU discovery. The path MTU discovery process ensures that a packet is sent using the largest possible size that is supported on a specific route.

A value of 58 in the Next Header field of the base IPv6 packet header identifies an IPv6 ICMP packet. The ICMP packet follows all the extension headers and is the last piece of information in the IPv6 packet. Within the IPv6 ICMP packets, the ICMPv6 Type and ICMPv6 Code fields identify IPv6 ICMP packet specifics, such as the ICMP message type. The value in the Checksum field is computed by the sender and checked by the receiver from the fields in the IPv6 ICMP packet and the IPv6 pseudo header.

**Note**

The IPv6 header does not have a checksum. But a checksum on the transport layer can determine if packets have not been delivered correctly. All checksum calculations that include the IP address in the calculation must be modified for IPv6 to accommodate the new 128-bit address. A checksum is generated using a pseudo header.

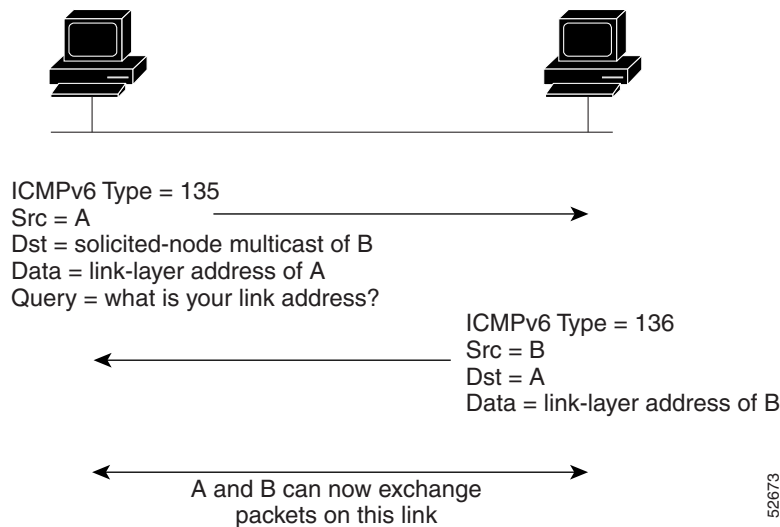
The ICMPv6 Payload field contains error or diagnostic information that relates to IP packet processing.

IPv6 Neighbor Discovery

You can use the IPv6 Neighbor Discovery Protocol (NDP) to determine whether a neighboring router is reachable. IPv6 nodes use neighbor discovery to determine the addresses of nodes on the same network (local link), to find neighboring routers that can forward their packets, to verify whether neighboring routers are reachable or not, and to detect changes to link-layer addresses. NDP uses ICMP messages to detect whether packets are sent to neighboring routers that are unreachable.

IPv6 Neighbor Solicitation Message

A node sends a neighbor solicitation message, which has a value of 135 in the Type field of the ICMP packet header, on the local link when it wants to determine the link-layer address of another node on the same local link. The source address is the IPv6 address of the node that sends the neighbor solicitation message. The destination address is the solicited-node multicast address that corresponds to the IPv6 address of the destination node. The neighbor solicitation message also includes the link-layer address of the source node.

Figure 65-8 IPv6 Neighbor Discovery-Neighbor Solicitation Message

After receiving the neighbor solicitation message, the destination node replies by sending a neighbor advertisement message, which has a value of 136 in the Type field of the ICMP packet header, on the local link. The source address is the IPv6 address of the node (the IPv6 address of the node interface that sends the neighbor advertisement message). The destination address is the IPv6 address of the node that sends the neighbor solicitation message. The data portion includes the link-layer address of the node that sends the neighbor advertisement message.

After the source node receives the neighbor advertisement, the source node and destination node can communicate.

Neighbor solicitation messages can verify the reachability of a neighbor after a node identifies the link-layer address of a neighbor. When a node wants to verify the reachability of a neighbor, it uses the destination address in a neighbor solicitation message as the unicast address of the neighbor.

Neighbor advertisement messages are also sent when there is a change in the link-layer address of a node on a local link. When there is a change, the destination address for the neighbor advertisement is the all-nodes multicast address.

Neighbor unreachability detection identifies the failure of a neighbor or the failure of the forward path to the neighbor and is used for all paths between hosts and neighboring nodes (hosts or routers).

Neighbor unreachability detection is performed for neighbors to which only unicast packets are being sent and is not performed for neighbors to which multicast packets are being sent.

A neighbor is considered reachable when a positive acknowledgment is returned from the neighbor (indicating that packets previously sent to the neighbor have been received and processed). A positive acknowledgment—from an upper-layer protocol (such as TCP)—indicates that a connection is making forward progress (reaching its destination). If packets are reaching the peer, they are also reaching the next-hop neighbor of the source. Forward progress is also a confirmation that the next-hop neighbor is reachable.

For destinations that are not on the local link, forward progress implies that the first-hop router is reachable. When acknowledgments from an upper-layer protocol are not available, a node probes the neighbor using unicast neighbor solicitation messages to verify that the forward path is still working. The return of a solicited neighbor advertisement message from the neighbor is a positive acknowledgment that the forward path is still working (neighbor advertisement messages that have the solicited flag set to a value of 1 are sent only in response to a neighbor solicitation message). Unsolicited messages confirm only the one-way path from the source to the destination node; solicited neighbor advertisement messages indicate that a path is working in both directions.

**Note**

A neighbor advertisement message that has the solicited flag set to a value of 0 is not considered as a positive acknowledgment that the forward path is still working.

Neighbor solicitation messages are also used in the stateless autoconfiguration process to verify the uniqueness of unicast IPv6 addresses before the addresses are assigned to an interface. Duplicate address detection is performed first on a new, link-local IPv6 address before the address is assigned to an interface (the new address remains in a tentative state while duplicate address detection is performed). A node sends a neighbor solicitation message with an unspecified source address and a tentative link-local address in the body of the message. If another node is already using that address, the node returns a neighbor advertisement message that contains the tentative link-local address. If another node is simultaneously verifying the uniqueness of the same address, that node also returns a neighbor solicitation message. If no neighbor advertisement messages are received in response to the neighbor solicitation message and no neighbor solicitation messages are received from other nodes that are attempting to verify the same tentative address, the node that sent the original neighbor solicitation message considers the tentative link-local address to be unique and assigns the address to the interface.

Configuring IPv6 Addressing and Basic Connectivity

- [Guidelines for Implementing IPv6 Addressing and Basic Connectivity, page 65-13](#)
- [Configuring IPv6 Addressing and Enabling IPv6 Routing, page 65-14](#)
- [Mapping Hostnames to IPv6 Addresses, page 65-15](#)
- [Configuration Examples for IPv6 Addressing and Basic Connectivity, page 65-16](#)

Guidelines for Implementing IPv6 Addressing and Basic Connectivity

- The switch supports only IPv6 unicast addresses. It does not support site-local unicast addresses or multicast addresses.
- The IPv6 128-bit addresses are represented as a series of eight 16-bit hexadecimal fields separated by colons in the format: *n:n:n:n:n:n:n:n*. It is an example of an IPv6 address:
2031:0000:130F:0000:0000:09C0:080F:130B
- The leading zeros in each field are optional, implementation is easier without them. It is the same address without leading zeros:
2031:0:130F:0:0:9C0:80F:130B
- You can also use two colons (::) to represent successive hexadecimal fields of zeros, but you can use this short version only once in each address:
2031:0:130F::09C0:080F:130B
- Multiple IPv6 global addresses within the same prefix can be configured on an interface; however, multiple IPv6 link-local addresses on an interface are not supported.

Configuring IPv6 Addressing and Enabling IPv6 Routing

Perform this task to assign IPv6 addresses to individual device interfaces and enable IPv6 traffic forwarding globally on the device. By default, IPv6 addresses are not configured and IPv6 routing is disabled.

Multiple IPv6 link-local addresses on an interface are not supported.

	Command	Purpose
Step 1	configure terminal Example: Switch# configure terminal	Enters global configuration mode.
Step 2	interface type number Example: Switch(config)# ip name-server 192.0.2.1 192.0.2.2	Specifies an interface type and number, and places the device in interface configuration mode.
Step 3	ipv6 address ipv6-prefix prefix-length eui-64 OR ipv6 address ipv6-prefix prefix-length link-local OR ipv6 enable Example: Switch(config-if)# ipv6 address 2001:DB8:0:1::/64 eui-64 OR Switch(config-if)# ipv6 address FE80::260:3EFF:FE11:6770 link-local OR Switch(config-if)# ipv6 enable	Specifies an IPv6 network assigned to the interface and enables IPv6 processing on the interface. or Specifies an IPv6 address assigned to the interface and enables IPv6 processing on the interface. or Automatically configures an IPv6 link-local address on the interface while also enabling the interface for IPv6 processing. The link-local address can be used only to communicate with nodes on the same link. Specifying the ipv6 address eui-64 command configures global IPv6 addresses with an interface identifier (ID) in the low-order 64 bits of the IPv6 address. Only the 64-bit network prefix for the address needs to be specified; the last 64 bits are automatically computed from the interface ID. Specifying the ipv6 address link-local command configures a link-local address on the interface that is used instead of the link-local address that is automatically configured when IPv6 is enabled on the interface.
Step 4	exit Example: Switch(config-if)# exit	Exits interface configuration mode, and returns the device to global configuration mode.
Step 5	ipv6 unicast-routing Example: Switch(config)# ipv6 unicast-routing	Enables the forwarding of IPv6 unicast datagrams.

Mapping Hostnames to IPv6 Addresses

A name server is used to track information associated with domain names. A name server can maintain a database of hostname-to-address mappings. Each name can map to one or more IPv4 addresses, IPv6 addresses, or both address types. In order to use this service to map domain names to IPv6 addresses, you must specify a name server and enable the DNS, which is the global naming scheme of the Internet that uniquely identifies network devices.

Cisco software maintains a cache of hostname-to-address mappings for use by the connect, telnet, and ping commands, related Telnet support operations, and many other commands that generate command output. This cache speeds the conversion of names to addresses.

Similar to IPv4, IPv6 uses a naming scheme that allows a network device to be identified by its location within a hierarchical name space that provides for domains. Domain names are joined with periods (.) as the delimiting characters. For example, Cisco is a commercial organization that is identified by a com domain name, so its domain name is cisco.com. A specific device in this domain, the FTP server, for example, is identified as ftp.cisco.com.

	Command	Purpose
Step 1	configure terminal Example: Switch# configure terminal	Enters global configuration mode.
Step 2	ip host name ipv6-address1 [ipv6-address2...ipv6-address4] Example: Switch(config)# ip host cisco0-sj 2001:DB8:20:1::12	Defines a static hostname-to-address mapping in the hostname cache. Typically, it is easier to refer to network devices by symbolic names rather than numerical addresses (services such as Telnet can use hostnames or addresses). Hostnames and IPv6 addresses can be associated with one another through static or dynamic means. Manually assigning hostnames to addresses is useful when dynamic mapping is not available.
Step 3	ip domain name [vrf vrf-name] name OR ip domain list [vrf vrf-name] name Example: Switch(config)# ip domain-name example1.com Switch(config)# ip domain list example2.com	(Optional) ip domain name —Defines a default domain name that Cisco software will use to complete unqualified hostnames. or (Optional) ip domain list —Defines a list of default domain names to complete unqualified hostnames. You can specify a default domain name that Cisco software will use to complete domain name requests. You can specify either a single domain name or a list of domain names. Any hostname that does not contain a complete domain name will have the default domain name you specify appended to it before the name is looked up. Note The ip domain name and ip domain list commands are used to specify default domain names that can be used by both IPv4 and IPv6.

	Command	Purpose
Step 4	ip name-server [vrf vrf-name] <i>server-address1</i> [<i>server-address2...server-address6</i>] Example: Switch(config)# ip name-server 2001:DB8::250:8bff:fee8:f800 2001:DB8:0:f004::1	Specifies one or more hosts that supply name information. Specifies one or more hosts (up to six) that can function as a name server to supply name information for DNS. Note The server-address argument can be either an IPv4 or IPv6 address.
Step 5	ip domain-lookup Example: Switch(config)# ip domain-lookup	Enables DNS-based address translation. DNS is enabled by default.

Displaying IPv6 Redirect Messages

Enter the following commands in the privileged EXEC mode:

Table 65-3 Displaying IPv6 Redirect Messages

Command or Action	Purpose
show ipv6 interface [brief] [<i>type number</i>] [<i>prefix</i>]	Displays the usability status of interfaces configured for IPv6.
show ipv6 route [<i>ipv6-address ipv6-prefix/prefix-length protocol interface-type interface-number</i>]	Displays the current contents of the IPv6 routing table.
show ipv6 traffic	Displays statistics about IPv6 traffic.
show hosts [vrf vrf-name all hostname summary]	Displays the default domain name, the style of name lookup service, a list of name server hosts, and the cached list of hostnames and addresses.
show running-config	Displays the current configuration running on the device.

Configuration Examples for IPv6 Addressing and Basic Connectivity

Example: IPv6 Addressing and IPv6 Routing Configuration

In the following example, IPv6 is enabled on the device with both a link-local address and a global address based on the IPv6 prefix 2001:DB8:c18:1::/64. The EUI-64 interface ID is used in the low-order 64 bits of both addresses. Output from the show ipv6 interface command is included to show how the interface ID (260:3EFF:FE47:1530) is appended to the link-local prefix FE80::/64 of Gigabit Ethernet interface 0/0/0.

```
ipv6 unicast-routing
```

```
interface gigabitethernet 0/0/0
  ipv6 address 2001:DB8:c18:1::/64 eui-64
Device# show ipv6 interface gigabitethernet 0/0/0
Gigabitethernet0/0/0 is up, line protocol is up
IPv6 is enabled, link-local address is FE80::260:3EFF:FE47:1530
Global unicast address(es):
  2001:DB8:C18:1:260:3EFF:FE47:1530, subnet is 2001:DB8:C18:1::/64
Joined group address(es):
  FE02::1
  FE02::2
  FE02::1:FE47:1530
  FE02::9
MTU is 1500 bytes
ICMP error messages limited to one every 500 milliseconds
ND reachable time is 30000 milliseconds
ND advertised reachable time is 0 milliseconds
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 200 seconds
ND router advertisements live for 1800 seconds
Hosts use stateless autoconfig for addresses.
```

Example: Dual-Protocol Stacks Configuration

The following example enables the forwarding of IPv6 unicast datagrams globally on the device and configures Gigabit Ethernet interface 0/0/0 with both an IPv4 address and an IPv6 address:

```
ipv6 unicast-routing
interface gigabitethernet0/0/0
  ip address 192.168.99.1 255.255.255.0
  ipv6 address 2001:DB8:c18:1::3/64
```

Example: Hostname-to-Address Mappings Configuration

The following example defines two static hostname-to-address mappings in the hostname cache, establishes a domain list with several alternate domain names to complete unqualified hostnames, specifies host 2001:DB8::250:8bff:fee8:f800 and host 2001:DB8:0:f004::1 as the name servers, and reenables the DNS service:

```
ip host cisco-sj 2001:DB8:700:20:1::12
ip host cisco-hq 2001:DB8:768::1 2001:DB8:20:1::22
ip domain list domain1-list.com
ip domain list serviceprovider2-name.com
ip domain list college2-name.edu
ip name-server 2001:DB8::250:8bff:fee8:f800 2001:DB8:0:f004::1
ip domain-lookup
```




Configuring SISF-Based Device Tracking

The Switch Integrated Security Features based (SISF-based) device tracking is part of the suite of first-hop security features. Starting from Cisco IOS XE Release 3.10.1E, this feature is available on Catalyst 4500E Series Switches with Supervisor Engines 9-E, 8-E, 8L-E, 7-E, 7L-E, and Catalyst 4500-X Series Switches.

This chapter describes how to enable and configure the feature. It consists of the following major sections:

- [About SISF-Based Device Tracking, page 66-1](#)
- [Guidelines for Enabling SISF-Based Device Tracking, page 66-2](#)
- [Migrating from Legacy Commands to SISF-Based Device Tracking, page 66-3](#)
- [Manually Enabling SISF-Based Device Tracking, page 66-4](#)
- [Programmatically Enabling SISF-Based Device Tracking, page 66-10](#)
- [Configuration Examples for SISF-Based Device Tracking, page 66-12](#)

About SISF-Based Device Tracking

The main purpose of the feature is to track the presence, location, and movement of end-nodes in the network. SISF snoops traffic received by the switch, extracts device identity (MAC and IP address), and stores them in a binding table. Many other features, such as, IEEE 802.1X, web authentication, IP Source Guard (IPSG), Cisco TrustSec and Cisco Locator/ID Separation Protocol (LISP) etc., depend on the accuracy of this information to operate properly.

SISF-based device tracking supports both IPv4 and IPv6.

If you are using the legacy IPv6 Snooping commands, we recommend that you migrate to SISF-based device tracking. See [Migrating from Legacy Commands to SISF-Based Device Tracking, page 66-3](#).



Note

Even if there is no existing IP device tracking (IPDT) or IPv6 Snooping configuration, you still have to enter the **device-tracking upgrade-cli** command in global configuration mode, before you can start using SISF-based device tracking.

After you migrate to SISF-based device tracking, the IPv6 Snooping commands will no longer be available; the IPDT commands continue to be supported and are required, to create IPv4 device tracking policies.

Guidelines for Enabling SISF-Based Device Tracking

SISF-Based device tracking is disabled by default.

You can enable the feature by manually creating the required policy, or it can be enabled programmatically. Review all the available options and choose the method that best suits your requirements.

This section includes the following major subsections:

- [Manually Enabling SISF-Based Device Tracking, page 66-4](#)
- [Programmatically Enabling SISF-Based Device Tracking, page 66-10](#)

Manually Enabling the Device Tracking

You can manually enable the feature by defining a device tracking policy and attaching the policy to a specific target. The target can be an interface or a VLAN. However, the steps you have to follow for IPv4 and IPv6 are different:



Tip

Use this method only if the programmatic policy's settings do not meet your requirements.

For IPv4, Use IPDT Commands

Enter the **ip device tracking maximum *n*** command in interface configuration mode; the system creates and attaches the IPDT-MAX-N policy to the interface. Note these guidelines and limitations:

- To attach a policy to a VLAN, configure the **ip device tracking maximum *n*** command on a switched virtual interface (SVI). See [Attaching a Device Tracking Policy to a VLAN, page 66-9](#).
- You cannot create a custom policy by specifying a name; The policy name can only be IPDT-MAX-N.

For IPv6, Use SISF-Based Device Tracking Commands

- If you enter the **device-tracking** command in the interface configuration mode or in the VLAN configuration mode, the system attaches a policy called **default**, to the interface or VLAN.
The **default** policy is a built-in policy with default settings; you cannot change any of the attributes of this policy.
- If you enter the **device-tracking policy** command in global configuration mode, you can specify a policy name; the system creates a policy with the name you specify. You can then configure the available settings, in the device tracking configuration mode (config-device-tracking) and attach the policy to a target.

For a Policy that Supports IPv4 and IPv6



Note

Create a policy for IPv4 and IPv6 separately and then attach it to the same interface or VLAN; the system merges the two policies and attaches it to the specified target. See example: [Example: Applying a Merged \(IPv4 and IPv6\) Device Tracking Policy to the Same Target, page 66-12](#).

Programmatically Enabling Device Tracking

Some features rely on device tracking and utilize the trusted database of binding entries that SISF-based device tracking builds and maintains. These features, also called device tracking clients, enable device tracking programmatically (create and attach the device tracking policy).

- The IEEE 802.1X, web authentication, IPSG, Cisco TrustSec features programmatically enable SISF-Based Device Tracking; the IPDT-DEFAULT policy is created and attached to the interface. The IPDT-DEFAULT policy supports only IPv4.
- When the LISP feature programmatically enables SISF-Based Device Tracking, the DT-PROGRAMMATIC policy is created and attached to the VLAN. The DT-PROGRAMMATIC policy supports both IPv4 and IPv6.



Tip

The IPDT-DEFAULT settings cannot be changed; DT-PROGRAMMATIC policy settings can.

Migrating from Legacy Commands to SISF-Based Device Tracking

You can migrate to SISF-based device tracking by entering the **device-tracking upgrade-cli** command in global configuration mode. Consider the following existing configuration scenarios and the corresponding migration results, before you migrate.

This section covers the following scenarios:

- [No IPDT and No IPv6 Snooping Configuration Exists, page 66-3](#)
- [Only IPDT Configuration Exists, page 66-3](#)
- [Only IPv6 Snooping Configuration Exists, page 66-4](#)
- [Both IPDT and IPv6 Snooping Configurations Exist, page 66-4](#)

No IPDT and No IPv6 Snooping Configuration Exists

Enter the **device-tracking upgrade-cli** command in global configuration mode. After this:

- SISF-based device tracking commands are available.
- IPDT commands continue to be available. Continue to use the IPDT commands to manually create a device tracking policy for IPv4.
- IPv6 snooping commands are not available.

Only IPDT Configuration Exists

After migration:

- SISF-based device tracking commands are available.
- IPDT commands continue to be available. Continue to use the IPDT commands to manually create a device tracking policy for IPv4.

- Entering the **[no] ip device tracking probe interval** command in interface configuration mode causes running configuration to display **ip device tracking probe interval** configuration and **device-tracking binding reachable-lifetime** command configurations. But if you configure **device-tracking binding reachable-lifetime** the system does not generate the **ip device tracking probe interval** command.
- IPv6 snooping commands are not available.

**Note**

Starting from Cisco IOS XE Release 3.10.1E, irrespective of whether you migrate to SISF-Based device tracking or not, the following IPDT commands are deprecated; there are no replacement commands:

- **[no] ip device tracking probe count**
- **[no] ip device tracking probe delay**

Only IPv6 Snooping Configuration Exists

After migration:

- SISF-based device-tracking commands are available.
- IPv6 snooping commands are not available.
- IPDT commands continue to be available. Continue to use the IPDT commands if you manually create a device tracking policy for IPv4.

Both IPDT and IPv6 Snooping Configurations Exist

After migration:

- SISF-based device-tracking commands are available.
- IPv6 snooping commands are not available.
- IPDT commands continue to be available. Continue to use the IPDT commands when you manually create a device tracking policy for IPv4.

Manually Enabling SISF-Based Device Tracking

This section includes the following major subsections:

- [Applying the Default Device Tracking Policy to a Target, page 66-5](#)
- [Creating a Custom Device Tracking Policy with Custom Settings, page 66-6](#)
- [Attaching a Device Tracking Policy to an Interface, page 66-9](#)
- [Attaching a Device Tracking Policy to a VLAN, page 66-9](#)
- [Attaching a Device Tracking Policy to a VLAN, page 66-9](#)

Applying the Default Device Tracking Policy to a Target

Beginning in privileged EXEC mode, follow these steps to apply the default device tracking policy to an interface or VLAN:



Note This procedure applies only to IPv6.

	Command or Action	Purpose
Step 1	configure terminal Example: Switch# configure terminal	Enters global configuration mode.
Step 2	interface <i>type/id</i> or vlan configuration <i>vlan_list</i> Example: Switch# interface gigabitethernet 1/1/4 or Switch# vlan configuration 333	<ul style="list-style-type: none"> • interface <i>type/id</i>—Specifies the interface and enters the interface configuration mode. The device tracking policy is attached to the specified interface. • vlan configuration <i>vlan_list</i>—Specifies the VLANs and enters the VLAN feature configuration mode. The device tracking policy is attached to the specified VLAN.
Step 3	device-tracking Example: Switch(config-if)# device-tracking or Switch(config-vlan-config)# device-tracking	<p>Enables SISF-based device tracking and attaches the default policy it to the interface or VLAN.</p> <p>The default policy is a built-in policy with default settings; none of the attributes of the default policy can be changed.</p>
Step 4	exit Example: Switch(config-if)# exit or Switch(config-vlan-config)# exit	Exits configuration mode.
Step 5	show device-tracking policy <i>policy-name</i> Example: Switch# show device-tracking policy default	Displays device-tracking policy configuration, and all the targets it is applied to.

Creating a Custom Device Tracking Policy with Custom Settings

Beginning in privileged EXEC mode, follow these steps to create and configure a device tracking policy:



Note

This procedure applies only to IPv6

	Command or Action	Purpose
Step 1	configure terminal Example: Switch# configure terminal	Enters global configuration mode.
Step 2	[no] device-tracking policy <i>policy-name</i> Example: Switch(config)# device-tracking policy example_policy	Creates the policy and enters the device-tracking configuration mode.

	Command or Action	Purpose
Step 3	<p>[data-glean default destination-glean device-role distribution-switch exit limit no prefix-glean protocol security-level tracking trusted-port vpc]</p> <p>Example:</p> <pre>Switch (config-device-tracking) # destination-glean log-only</pre>	<p>Enter the question mark (?) at the system prompt to obtain a list of available options in this mode. You can configure the following for IPv6:</p> <ul style="list-style-type: none"> (Optional) data-glean—Enables learning of addresses from a data packet snooped from a source inside the network and populates the binding table with the data traffic source address. Enter one of these options: <ul style="list-style-type: none"> log-only—Generates a system log message upon data packet notification. recovery—Uses a protocol to enable binding table recovery. Enter NDP or DHCP. (Optional) default—Sets the policy attribute to its default value. You can set these policy attributes to their default values: data-glean, destination-glean, device-role, limit, prefix-glean, protocol, security-level, tracking, trusted-port. (Optional) destination-glean—Populates the binding table by gleaning data traffic destination address. Enter one of these options: <ul style="list-style-type: none"> log-only—Generates a system log message upon data packet notification. recovery—Uses a protocol to enable binding table recovery. Enter DHCP. (Optional) device-role—Sets the role of the device attached to the port. It can be a node or a switch. Enter one of these options: <ul style="list-style-type: none"> node—Configures the attached device as a node. This is the default option. switch—Configures the attached device as a switch. (Optional) distribution-switch—Although visible on the CLI, this option is not supported. Any configuration settings you make will not take effect. (Optional) exit—Exits the device-tracking policy configuration mode. (Optional) limit address-count—Specifies an address count limit per port. The range is 1 to 32000. (Optional) no—Negates the command or sets it to defaults. (Optional) prefix-glean—Enables learning of prefixes from either IPv6 Router Advertisements or from DHCP-PD. You have the following option: <ul style="list-style-type: none"> (Optional) only—Gleans only prefixes and not host addresses. (Optional) protocol—Sets the protocol to glean; by default, all are gleaned. Enter one of these options:

Step 3
continued

Command or Action	Purpose
	<ul style="list-style-type: none"> – arp [<i>prefix-list name</i>] —Gleans addresses in ARP packets. Optionally, enter the name of prefix-list that is to be matched. – dhcp4 [<i>prefix-list name</i>] —Glean addresses in DHCPv4 packets. Optionally, enter the name of prefix-list that is to be matched. – dhcp6 [<i>prefix-list name</i>] —Glean addresses in DHCPv6 packets. Optionally, enter the name of prefix-list that is to be matched. – ndp [<i>prefix-list name</i>] —Glean addresses in NDP packets. Optionally, enter the name of prefix-list that is to be matched. – udp [<i>prefix-list name</i>] —Although visible on the CLI, this option is not supported. Any configuration settings you make will not take effect. • (Optional) security-level —Specifies the level of security enforced by the feature. Enter one of these options: <ul style="list-style-type: none"> – glean —Gleans addresses passively. – guard —Inspects and drops un-authorized messages. This is the default. – inspect —Gleans and validates messages. • (Optional) tracking —Specifies a tracking option. Enter one of these options: <ul style="list-style-type: none"> – disable [stale-lifetime [1-86400-seconds infinite]] —Turns off device-tracking. Optionally, you can enter the duration for which the entry is kept inactive before deletion, or keep it permanently inactive. – enable [reachable-lifetime [1-86400-seconds infinite]] —Turns on device-tracking. Optionally, you can enter the duration for which the entry is kept reachable, or keep it permanently reachable. • (Optional) trusted-port —Sets up a trusted port. Disables the guard on applicable targets. Bindings learned through a trusted port have preference over bindings learned through any other port. A trusted port is given preference in case of a collision while making an entry in the table. • (Optional) vpc —Although visible on the CLI, this option is not supported. Any configuration settings you make will not take effect.

	Command or Action	Purpose
Step 4	exit Example: Switch(config-if)# exit or Switch(config-vlan-config)# exit	Exits configuration mode.
Step 5	show device-tracking policy <i>policy-name</i> Example: Switch# show device-tracking policy default	Displays device-tracking policy configuration, and all the targets it is applied to.

Attaching a Device Tracking Policy to an Interface

Beginning in privileged EXEC mode, follow these steps to attach a device tracking policy to an interface:

	Command or Action	Purpose
Step 1	configure terminal Example: Switch# configure terminal	Enters global configuration mode.
Step 2	interface <i>type/id</i> Example: Switch# interface gigabitethernet 1/1/4	Specifies the interface and enters the interface configuration mode. The device tracking policy is attached to the specified interface.
Step 3	[no] device-tracking attach-policy <i>policy-name</i> Example: Switch(config-if)# device-tracking attach-policy example_policy	Attaches the device tracking policy to the interface. Device tracking is also supported on EtherChannels Note SISF based device-tracking policies can be detached only if they are custom policies. Programmatically created policies can be removed only if the corresponding device-tracking client feature configuration is removed.
Step 4	show device-tracking policies [interface <i>type/id</i>] Example: Switch# show device-tracking policies interface gigabitethernet 1/1/4	Displays policies that match the specified interface type and number.

Attaching a Device Tracking Policy to a VLAN

Beginning in privileged EXEC mode, follow these steps to attach a device-tracking policy to VLANs across multiple interfaces:

	Command or Action	Purpose
Step 1	configure terminal Example: Switch# configure terminal	Enters global configuration mode.
Step 2	vlan configuration <i>vlan_list</i> Example: Switch# vlan configuration 333	Specifies the VLANs to which the device tracking policy will be attached; enters the VLAN interface configuration mode.
Step 3	[no] device-tracking attach-policy <i>policy-name</i> Example: Switch (config-vlan-config)# device-tracking attach-policy example_policy	Attaches the device tracking policy to the specified VLANs across all switch interfaces. Note SISF based device-tracking policies can be detached only if they are custom policies. Programmatically created policies can be removed only if the corresponding device-tracking client feature configuration is removed.
Step 4	end Example: Switch(config-if)# end	Returns to the privileged EXEC mode.
Step 5	do show device-tracking policies vlan <i>vlan-ID</i> Example: Switch(config-vlan-config)# do show device-tracking policies vlan 333	Verifies that the policy is attached to the specified VLAN, without exiting the VLAN interface configuration mode.

Programmatically Enabling SISF-Based Device Tracking

Device tracking client features that enable SISF-based device tracking	These features can programmatically enable SISF-based device tracking: <ul style="list-style-type: none"> • IEEE 802.1X, web authentication, Cisco TrustSec, and IPSG. • Cisco LISP.
--	--

Policy Name	<p>For IEEE 802.1X, web authentication, IPSG, and Cisco TrustSec, the IPDT-DEFAULT policy is created and attached to the interface.</p> <p>For LISP, the DT-PROGRAMMATIC policy is created and attached to the VLAN.</p>
User Options	<ul style="list-style-type: none"> • The policy cannot be replaced by another policy. • The policy cannot be removed unless the device tracking client feature configuration is removed. • Only one policy can be attached to the same interface or VLAN. <p>If you create an IPv4 and an IPv6 policy, the system merges it and applies the merged policy to the interface or VLAN.</p> <ul style="list-style-type: none"> • You cannot change the settings of a merged policy. <p>You can change the settings of the IPv6 policy separately. Changing the settings of such a custom IPv6 policy (which is part of a merged policy) will cause the settings to also be applied to the IPv4 policy which is part of the merged policy.</p> <ul style="list-style-type: none"> • You cannot change the settings of any IPv4 policy. • You can change the settings of <ul style="list-style-type: none"> – a DT-PROGRMMATC policy; IPv4 and IPv6 settings are changed – a custom IPv6 policy <p>In case of DT-PROGRMMATC and a custom IPv6 policy, these are the settings you can change: (device-tracking policy command, in the device tracking configuration mode (config-device-tracking))</p> <ul style="list-style-type: none"> – data-glean – default – device-role – destination-glean – exit – limit – no – prefix-glean – protocol – security-level – tracking – trusted-port – The distribution-switch and vpc options are visible on the CLI, but any configuration changes are not effective. <p>The address count limit per MAC setting cannot be changed (This refers to the limit address-count for IPv4 per mac and limit address-count for IPv6 per mac commands), but the address count limit per port or interface can be changed.</p> <p>When a device-tracking policy is attached to an interface under a VLAN, the policy settings on the interface take precedence over those on its VLAN; exceptions here are the values for limit address-count for IPv4 per mac and limit address-count for IPv6 per mac, which are aggregated from the policy on both the interface and VLAN.</p>

Configuration Examples for SISF-Based Device Tracking

This section provides examples for the following scenarios:

- [Example: Configuring a Multi-Switch Network to Stop Creating Binding Entries from a Trunk Port, page 66-12](#)
- [Example: Applying a Merged \(IPv4 and IPv6\) Device Tracking Policy to the Same Target, page 66-12](#)
- [Example: Manually Attaching an IPv4 Device Tracking Policy to a VLAN, page 66-14](#)
- [Example: Disabling IPv6 Device Tracking, page 66-15](#)
- [Example: Enabling IPv6 for SVI on VLAN \(To Mitigate the Duplicate Address Problem\), page 66-15](#)
- [Example: Mitigating the IPv4 Duplicate Address Problem, page 66-15](#)
- [Example: Avoiding a Short Device-Tracking Binding Reachable Time, page 66-17](#)

Example: Configuring a Multi-Switch Network to Stop Creating Binding Entries from a Trunk Port

In a multi-switch network, SISF-based device tracking provides the capability to distribute binding table entries between switches running the feature. Binding entries are only created on the switches where the host appears on an access port. No entry is created for a host that appears over a trunk port. This is achieved by configuring a custom policy with the **trusted-port** and **device-role switch** options, and attaching it to the trunk port. The following example shows you how:



Note

Both, the **trusted-port**, and **device-role switch** options, must be configured in the policy. Further, we recommended that you apply such a policy on a port facing a device, which also has SISF-based device tracking enabled.

```
Switch# configure terminal
Switch(config)# device-tracking policy example_trusted_policy
Switch(config-device-tracking)# device-role switch
Switch(config-device-tracking)# trusted-port
Switch(config-device-tracking)# end
Switch(config)# interface gigabitethernet 1/0/25
Switch(config-if)# device-tracking attach-policy example_trusted_policy
```

Example: Applying a Merged (IPv4 and IPv6) Device Tracking Policy to the Same Target

This example shows you how to create IPv4 and IPv6 policies separately and then attach them to the same interface or VLAN:

Create an IPv4 policy using the IPDT command, and use the **show** command to display settings and verify that it is attached to the target:

```
Switch# configure terminal
Switch(config)# interface gigabitethernet2/47
Switch(config-if)# ip device tracking max 10
Switch(config-if)# end
```

```
Switch# show device-tracking policy IPDT-MAX-10
Policy IPDT-MAX-10 configuration:
  security-level guard
  device-role node
  NOT glean from Neighbor Discovery
  NOT glean from DHCP
  glean from ARP
  glean from DHCP4
  NOT glean from protocol unkn
  limit address-count for IPv4 per mac 1
  limit address-count for IPv4 per target 10
  tracking enable
Policy IPDT-MAX-10 is applied on the following targets:
Target          Type Policy          Feature    Target range
Gi2/47          PORT IPDT-MAX-10    Snooping   vlan all
```

Create a custom IPv6 policy and use the **show** command to display its settings. Note that this policy has not been attached to a target yet:

```
Switch# configure terminal
Switch(config)# device-tracking policy eg_cust_v6
Switch(config-device-tracking)# end

Switch# show device-tracking policy eg_cust_v6
Policy eg_cust_v6 configuration:
  security-level guard
  device-role node
  glean from Neighbor Discovery
  glean from DHCP
  NOT glean from ARP
  NOT glean from DHCP4
  NOT glean from protocol unkn
Policy eg_cust_v6 is applied on the following targets:
Target          Type Policy          Feature    Target range
eg_cust_v6
```

Attach the custom IPv6 policy to the same target as the IPv4 policy, that is, interface GigabitEthernet 2/47:

```
Switch# configure terminal
Switch(config)# interface gigabitethernet2/47
Switch(config-if)# device-tracking attach-policy eg_cust_v6
Switch(config-if)# end
```

Enter the **show** command to display the merged policy that is now attached to interface GigabitEthernet 2/47:



Note You cannot change the settings of the merged **IPDT-MAX-10+_eg_cust_v6** policy. You can however change the settings of **eg_cust_v6**. Changes made to **eg_cust_v6**, are applied to **IPDT-MAX-10** as well.

```
Switch# show device-tracking policy IPDT-MAX-10+_eg_cust_v6
Policy IPDT-MAX-10+_eg_cust_v6 configuration:
  security-level guard
  device-role node
  glean from Neighbor Discovery
  glean from DHCP
  glean from ARP
  glean from DHCP4
  NOT glean from protocol unkn
  limit address-count for IPv4 per mac 1
  limit address-count for IPv4 per target 10
```

```

tracking enable
Policy IPDT-MAX-10+_eg_cust_v6 is applied on the following targets:
Target  Type  Policy                      Feature      Target range
Gi2/47  PORT   IPDT-MAX-10+_eg_cust_v6 Snooping     vlan all

```

Example: Manually Attaching an IPv4 Device Tracking Policy to a VLAN

This example shows how to attach an IPv4 device tracking policy to a VLAN:

```

Switch# configure terminal
Switch(config)# interface
Switch(config)# interface vlan 300
Switch(config-if)# ip device tracking maximum 10
Switch(config-if)# end

Switch# show device-tracking policies
Target      Type  Policy      Feature      Target range
vlan 300    VLAN  IPDT-MAX-10 Snooping     vlan all

```



Note The value you specify for **ip device tracking maximum *n*** refers to the maximum number of entries that are permitted for the specified VLAN. The permitted value range is 0 to 65535; entering 0 means entries are not learned.

Example: Programmatically Enabling SISF-Based Device Tracking

This example show you the settings of a programmatic policy that is created when you configure LISP on the switch.

Device tracking client: LISP (The LISP configuration here is only meant to serve as an example).

After you configure LISP, enter the **show device-tracking policy** command in privileged EXEC mode, to display the DT-PROGRAMMATIC policy that is created and the corresponding settings:

```

Switch(config)# router lisp
<output truncated>
Switch(config-router-lisp)# instance-id 3
Switch(config-router-lisp-instance)# service ethernet
Switch(config-router-lisp-instance-service)# eid-table vlan 10
Switch(config-router-lisp-instance-dynamic-eid)# database-mapping 10.1.1.0/24
locator-set set1
Switch(config-router-lisp-instance-service)# exit-service-ethernet
Switch(config-router-lisp-instance)# exit-instance-id
Switch(config-router-lisp)# exit-router-lisp

Switch# show device-tracking policy DT-PROGRAMMATIC
Policy DT-PROGRAMMATIC configuration:
  security-level guard (*)
  device-role node
  gleaning from Neighbor Discovery
  gleaning from DHCP
  gleaning from ARP
  gleaning from DHCP4
  NOT gleaning from protocol unkn
  limit address-count for IPv4 per mac 1 (*)
  limit address-count for IPv6 per mac 8 (*)
  tracking enable
Policy DT-PROGRAMMATIC is applied on the following targets:

```

Target	Type	Policy	Feature	Target range
vlan 10	VLAN	DT-PROGRAMMATIC	Device-tracking	vlan all

note:
 Binding entry Down timer: 10 minutes (*)
 Binding entry Stale timer: 30 minutes (*)

Example: Disabling IPv6 Device Tracking

If SISF-based device tracking is enabled by a custom policy, detach the policy from the target by entering the **no device-tracking attach-policy *policy-name*** command in interface configuration mode. For example, to detach custom policy `example_policy` from interface GigabitEthernet 1/1/4:

```
Switch# configure terminal
Switch(config)# interface gigabitethernet 1/1/4
Switch(config-if)# no device-tracking attach-policy example_policy
Switch# end
```

If SISF-based device tracking is programmatically enabled, by configuring LISP, enter the following commands to change the programmatic policy and thus disable SISF-based device tracking:

```
Switch(config)# device-tracking policy DT-PROGRAMMATIC
Switch(config-device-tracking)# no protocol ndp
Switch(config-device-tracking)# no protocol dhcp6
Switch(config-device-tracking)# end
```

Example: Enabling IPv6 for SVI on VLAN (To Mitigate the Duplicate Address Problem)

For an IPv6 device-tracking entry, its reachability is verified by sending an SISF probe to its end-node, which is a neighbor solicitation message. Selection of the source IP address for this neighbor solicitation probe follows these rules:

- If an SVI is configured on the VLAN, the link-local IPv6 address of the SVI is selected. Please ensure that the SVI IP address is unique in the subnet.
- Otherwise, an address with all zeros (0:0:0:0:0:0:0:0) is selected.

When IPv6 is enabled in the network and a switched virtual interface (SVI) is configured on a VLAN, we recommend that you add the following to the SVI configuration. This enables the SVI to acquire a link-local address automatically; this address is used as the source IP address of the SISF probe, thus preventing the duplicate IP address issue.

```
Switch(config)# interface vlan 10
Switch(config-if)# ipv6 enable
Switch(config-if)# end
```

Example: Mitigating the IPv4 Duplicate Address Problem

For an IPv4 device-tracking entry, its reachability is verified by sending an SISF probe to its end-node, which is an ARP request message. Selection of the source IP address for this ARP probe follows these rules:

- If an SVI is configured on the VLAN, the IPv4 address of the SVI is selected. Please ensure that the SVI IP address is unique in the subnet.
- If SVI does not exist and the **device-tracking tracking auto-source [fallback host-ip mask] [override]** command is configured, source IP is selected according to the table below.
- Otherwise, an address with all zeros (0.0.0.0) is selected.

This example shows how you can tackle the Duplicate IP Address 0.0.0.0 error message problem encountered by clients that run Microsoft Windows:

Configure the **ip device tracking probe auto-source** command in global configuration mode. This command determines the source IP and MAC address used in the Address Resolution Packet (ARP) request sent by the switch to probe a client, in order to maintain its entry in the device-tracking table. The purpose, is to avoid using 0.0.0.0 as source IP address.



Note Configure the **ip device tracking probe auto-source** command only when a switch virtual interface (SVI) is not configured. You do not have to configure it when a SVI is configured with an IPv4 address on the VLAN.

Command	Required Action (In order to select source IP and MAC address for device tracking ARP probe)	Notes
ip device tracking probe auto-source	Set source to VLAN SVI if present. Look for IP and MAC binding in device-tracking table from same subnet. Use 0.0.0.0	We recommend that you disable device-tracking on all trunk ports to avoid MAC flapping.
ip device tracking probe auto-source override	Set source to VLAN SVI if present Use 0.0.0.0	Not recommended when there is no SVI.
ip device tracking probe auto-source fallback 0.0.0.X 255.255.255.0	Set source to VLAN SVI if present. Look for IP and MAC binding in device-tracking table from same subnet. Compute source IP from client IP using host bit and mask provided. Source MAC is taken from the MAC address of the switchport facing the client ¹ .	We recommend that you disable device-tracking on all trunk ports to avoid MAC flapping. The computed IPv4 address must not be assigned to any client or network device.
ip device tracking probe auto-source fallback 0.0.0.X 255.255.255.0 override	Set source to VLAN SVI if present. Compute source IP from client IP using host bit and mask provided ¹ . Source MAC is taken from the MAC address of the switchport facing the client ¹ .	-

1. Depending on the client IP address, an IPv4 address has to be reserved for the source IP.

A reserved source IPv4 address = (client-ip and mask) | host-ip

Client IP = 192.0.2.25

Source IP = (192.0.2.25 and 255.255.255.0) | (0.0.0.1) = 192.0.2.1

IP address 192.0.2.1 should not be assigned to any client or network device.

Example: Avoiding a Short Device-Tracking Binding Reachable Time

When migrating from an older release, the following configuration may be present:

device-tracking binding reachable-time 10

Remove this by entering the **no** version of the command.



Port Unicast and Multicast Flood Blocking

This chapter describes how to configure multicast and unicast flood blocking on the Catalyst 4500 series switch. This chapter contains these topics:

- [About Flood Blocking, page 67-1](#)
- [Configuring Port Blocking, page 67-1](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About Flood Blocking

Occasionally, unknown unicast or multicast traffic is flooded to a switch port because a MAC address has timed out or has not been learned by the switch. (This condition is especially undesirable for a private VLAN isolated port.) To guarantee that no unicast and multicast traffic is flooded to the port, use the **switchport block unicast** and **switchport block multicast** commands to enable flood blocking on the switch.



Note

The flood blocking feature is supported on all switched ports (including PVLAN ports) and is applied to all VLANs on which the port is forwarding.

Configuring Port Blocking

By default, a switch floods packets with unknown destination MAC addresses to all ports. If unknown unicast and multicast traffic is forwarded to a switch port, there might be security issues. To prevent forwarding such traffic, you can configure a port to block unknown unicast or multicast packets.



Note

Blocking of unicast or multicast traffic is not automatically enabled on a switch port; you must explicitly configure it.

Blocking Flooded Traffic on an Interface



Note The interface can be a physical interface (for example, GigabitEthernet 1/1) or an EtherChannel group (such as port-channel 5). When you block multicast or unicast traffic for a port channel, it is blocked on all ports in the port channel group.



Note Starting with Cisco IOS Release 12.2(52)SG, only IPV4 and IPv6 unknown multicast traffic flooding is blocked; Layer 2 unknown multicast flooding is not. This behavior stems from a fix for the following problem: when you configure blocking of unknown multicast flooding on a port, broadcast traffic to the port is also blocked.

To disable the flooding of multicast and unicast packets to an interface, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Enters interface configuration mode and enter the type and number of the switch port interface (for example, GigabitEthernet 1/1).
Step 3	Switch(config-if)# switchport block multicast	Blocks unknown multicast forwarding to the port.
Step 4	Switch(config-if)# switchport block unicast	Blocks unknown unicast forwarding to the port.
Step 5	Switch(config)# end	Returns to privileged EXEC mode.
Step 6	Switch# show interface <i>interface-id</i> switchport	Verifies your entry.
Step 7	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to block unicast and multicast flooding on a GigabitEthernet interface1/1 and how to verify the configuration:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface gigabitethernet1/1
Switch(config-if)# switchport block multicast
Switch(config-if)# switchport block unicast
Switch(config-if)# end
Switch# show interface gigabitethernet1/1 switchport
Name: Gi1/3
Switchport: Enabled

<output truncated>

Port Protected: On
Unknown Unicast Traffic: Not Allowed
Unknown Multicast Traffic: Not Allowed

Broadcast Suppression Level: 100
Multicast Suppression Level: 100
Unicast Suppression Level: 100
```

Resuming Normal Forwarding on a Port

To resume normal forwarding on a port, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Enters interface configuration mode and enter the type and number of the switch port interface (GigabitEthernet1/1).
Step 3	Switch(config-if)# no switchport block multicast	Enables unknown multicast flooding to the port.
Step 4	Switch(config-if)# no switchport block unicast	Enables unknown unicast flooding to the port.
Step 5	Switch(config)# end	Returns to privileged EXEC mode.
Step 6	Switch# show interface <i>interface-id</i> switchport	Verifies your entry.
Step 7	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.



Configuring Storm Control

This chapter describes how to configure port-based traffic control on a switch and consists of these sections:

- [About Storm Control, page 68-1](#)
- [Enabling Broadcast Storm Control, page 68-3](#)
- [Enabling Multicast Storm Control, page 68-4](#)
- [Disabling Broadcast Storm Control, page 68-5](#)
- [Disabling Multicast Storm Control, page 68-6](#)
- [Displaying Storm Control, page 68-6](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About Storm Control

This section contains the following subsections:

- [Hardware-Based Storm Control Implementation, page 68-1](#)
- [Software-Based Storm Control Implementation, page 68-2](#)

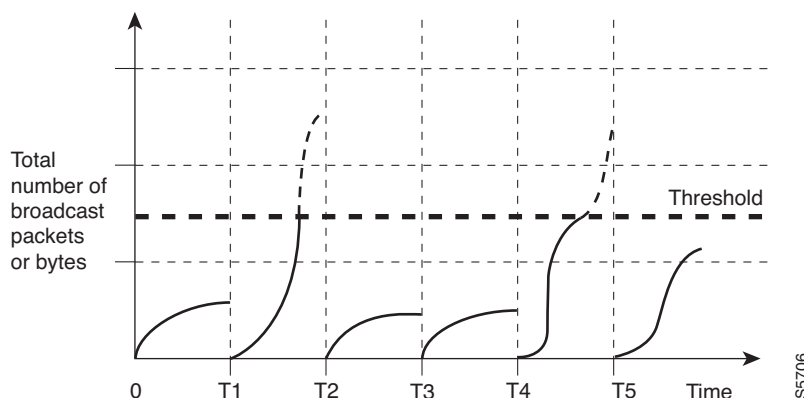
Storm control prevents LAN interfaces from being disrupted by a broadcast storm. A broadcast storm occurs when broadcast packets flood the subnet, creating excessive traffic and degrading network performance. Errors in the protocol-stack implementation or in the network configuration can cause a broadcast storm.

Hardware-Based Storm Control Implementation

Broadcast suppression uses filtering that measures broadcast activity in a subnet over a one-second interval and compares the measurement with a predefined threshold. If the threshold is reached, further broadcast activity is suppressed for the duration of the interval. Broadcast suppression is disabled by default.

Figure 68-1 shows the broadcast traffic patterns on a LAN interface over a given interval. In this example, broadcast suppression occurs between times T1 and T2 and between T4 and T5. During those intervals, the amount of broadcast traffic exceeded the configured threshold.

Figure 68-1 Storm Control Example—Hardware-based Implementation



The broadcast suppression threshold numbers and the time interval combination make the broadcast suppression algorithm work with different levels of granularity. A higher threshold allows more broadcast packets to pass through.

Broadcast suppression on the Catalyst 4500 series switches is implemented in hardware. The suppression circuitry monitors packets passing from a LAN interface to the switching bus. If the packet destination address is broadcast, then the broadcast suppression circuitry tracks the current count of broadcasts within the one-second interval, and when a threshold is reached, it filters out subsequent broadcast packets.

Because hardware broadcast suppression uses a bandwidth-based method to measure broadcast activity, the most significant implementation factor is setting the percentage of total available bandwidth that can be used by broadcast traffic. Because packets do not arrive at uniform intervals, the one-second interval during which broadcast activity is measured can affect the behavior of broadcast suppression.

Software-Based Storm Control Implementation

When storm control is enabled on an interface, the switch monitors packets received on the interface and determines whether the packets are broadcast. The switch monitors the number of broadcast packets received within a one-second time interval. When the interface threshold is met, all incoming data traffic on the interface is dropped. This threshold is specified as a percentage of total available bandwidth that can be used by broadcast traffic. If the lower threshold is specified, all data traffic is forwarded as soon as the incoming traffic falls below that threshold.




Note

A Cisco Catalyst 4500-X series switch checks for a broadcast storm in real-time, too. When a broadcast storm occurs on a Cisco Catalyst 4500-X series switch and the threshold is reached within a fraction of a second, the broadcast is suppressed. This behavior on Cisco Catalyst 4500-X series switches is in addition to the hardware-based storm control and software-based storm control described earlier.

Enabling Broadcast Storm Control

To enable storm control, perform this task:

	Command or Action	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Enters interface configuration mode and enters the port to configure.
Step 3	Switch(config-if)# storm-control broadcast level { <i>high-level</i> bps <i>bps</i> [k m g] pps <i>pps</i> [k m g]}	<p>Configures broadcast storm control. The keywords and arguments are described here.</p> <ul style="list-style-type: none"> <i>high-level</i>—Specifies the upper threshold levels for broadcast traffic as a percentage (up to two decimal places) of the bandwidth. The port blocks traffic that exceeds this level. The range is from 0 to 100. <p> Note For the Catalyst 4500-X Series Switch, on ports operating at 1Gigabit, thresholds less than 0.02% are not supported.</p> <ul style="list-style-type: none"> bps <i>bps</i>—Specifies the threshold level for broadcast traffic in bits per second (bps) (up to one decimal place). The port blocks only the traffic that exceeds this level. The range is 0.0 to 10000000000.0. pps <i>pps</i>—Specifies the threshold level for broadcast traffic in packets per second (pps) (up to one decimal place). The port blocks all traffic when traffic utilization exceeds this level. The range is 0.0 to 10000000000.0. (Optional) [k m g]—Specifies the metric suffixes for large number thresholds, in bps and pps settings.
Step 4	Switch(config-if)# storm-control action { shutdown trap }	<p>Specifies the action to be taken when a storm is detected.</p> <p>The default is to filter out the broadcast traffic and not to send out traps.</p> <p>The shutdown keyword sets the port to the error-disable state during a storm. If the recover interval is not set, the port remains in shutdown state.</p>
Step 5	Switch(config-if)# exit	Returns to configuration mode.
Step 6	Switch(config)# end	Returns to privileged EXEC mode.
Step 7	Switch# show storm-control [<i>interface</i>] broadcast	Displays the number of packets suppressed.
Step 8	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to enable storm control on an interface and verify configuration:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface gigabitethernet 2/1
Switch(config-if)# storm-control broadcast level bps 100k
```

```

Switch(config-if)# end

Switch# show storm-control gigabitethernet 2/1
Interface  Filter State    Broadcast Multicast Level
-----
Gi2/1      Link Down    Enabled   Disabled   100k bps

Switch# show interfaces counters storm-control
Port          Broadcast  Multicast    Level    TotalSuppressedPackets
Gi2/1         Enabled    Disabled      100k bps          0

Switch# show interface gigabitethernet 2/1 capabilities
GigabitEthernet2/1
  Model:                WS-X4648-RJ45V+E-RJ-45
  Type:                  10/100/1000-TX
  Speed:                 10,100,1000,auto
  Duplex:                 half,full,auto
  Auto-MDIX:              yes
  EEE:                    no
  Trunk encap. type:      802.1Q
  Trunk mode:              on,off,desirable,nonegotiate
  Channel:                yes
  Broadcast suppression: percentage(0-100), hw
  Multicast suppression: percentage(0-100), hw
  Flowcontrol:             rx-(off,on,desired),tx-(off,on,desired)
  VLAN Membership:        static, dynamic
  Fast Start:              yes
  CoS rewrite:             yes
  ToS rewrite:             yes
  Inline power:            yes (Cisco Voice Protocol/IEEE Protocol 802.3af)
  SPAN:                   source/destination
  UDLD:                   yes
  Link Debounce:           no
  Link Debounce Time:      no
  Port Security:           yes
  Dot1x:                  yes
  Maximum MTU:             9198 bytes (Jumbo Frames)
  Multiple Media Types:    no
  Diagnostic Monitoring:   N/A

```

Enabling Multicast Storm Control

Per-interface multicast suppression, which allows you to subject incoming multicast and broadcast traffic to interface-level suppression.



Note

Multicast and broadcast suppression share a common threshold per interface. Multicast suppression takes effect *only* if broadcast suppression is enabled. Disabling broadcast suppression on an interface also disables multicast suppression.

To enable multicast suppression, perform this task:

	Command or Action	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Enters interface configuration mode and specifies the port to be configured.

	Command or Action	Purpose
Step 3	Switch(config-if)# storm-control broadcast include multicast	Enables multicast suppression.
Step 4	Switch(config-if)# exit	Returns to configuration mode.
Step 5	Switch(config)# end	Returns to privileged EXEC mode.
Step 6	Switch# show storm-control [interface] multicast	Verifies the configuration.

This example shows how to enable multicast suppression on ports that already have broadcast suppression enabled:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface gigabitethernet 2/1
Switch(config-if)# storm-control broadcast include multicast
Switch(config-if)# end
```

```
Switch# show storm-control gigabitethernet 2/1
Interface  Filter State  Broadcast Multicast Level
-----  -
Gi2/1      Forwarding  Enabled   Enabled   50.00%
```

```
Switch# show interface counters storm-control
Port          Broadcast  Multicast  Level  TotalSuppressedPackets
Gi2/1         Enabled    Enabled    50.00% 0
```

Disabling Broadcast Storm Control

To disable storm control, perform this task:

	Command or Action	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface interface-id	Enters interface configuration mode and specifies the port to be configured.
Step 3	Switch(config-if)# no storm-control broadcast level	Disables port storm control.
Step 4	Switch(config-if)# no storm-control action {shutdown trap}	Disables the specified storm control action and returns to default filter action.
Step 5	Switch(config-if)# exit	Returns to configuration mode.
Step 6	Switch(config)# end	Returns to privileged EXEC mode.
Step 7	Switch# show storm-control broadcast	Verifies your entries.
Step 8	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to enable multicast suppression on ports that already have broadcast suppression enabled:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# interface gigabitethernet 2/1
```

```

Switch(config-if)# storm-control broadcast include multicast
Switch(config-if)# end

Switch# show storm-control gigabitethernet 2/1
Interface  Filter State    Broadcast Multicast Level
-----
Gi2/1      Forwarding    Enabled   Enabled   50.00%

Switch# show interface counters storm-control
Port          Broadcast  Multicast    Level    TotalSuppressedPackets
Gi2/1         Enabled   Enabled      50.00%      0

```

Disabling Multicast Storm Control

To disable multicast suppression, perform this task:

	Command or Action	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# [no] storm-control broadcast include multicast	Enables and disables multicast suppression.
Step 3	Switch(config-if)# no storm-control broadcast level	Disables port storm control (broadcast and multicast).
Step 4	Switch(config-if)# exit	Returns to configuration mode.
Step 5	Switch(config)# end	Returns to privileged EXEC mode.

Displaying Storm Control

Use the **show interface capabilities** command to determine the mode in which storm control is supported in an interface.

This example shows how an interface that supports broadcast suppression in software:

```

Switch# show interface gigabitethernet 2/1 capabilities
GigabitEthernet2/1
  Model: WS-X4648-RJ45V-E-RJ-45
  Type: 10/100/1000-TX
  Speed: 10,100,1000,auto
  Duplex: half,full,auto
  Auto-MDIX: yes
  EEE: no
  Trunk encap. type: 802.1Q
  Trunk mode: on,off,desirable,nonegotiate
  Channel: yes
  Broadcast suppression: percentage(0-100), hw
  Multicast suppression: percentage(0-100), hw
  Flowcontrol: rx-(off,on,desired),tx-(off,on,desired)
  VLAN Membership: static, dynamic
  Fast Start: yes
  CoS rewrite: yes
  ToS rewrite: yes
  Inline power: yes (Cisco Voice Protocol/IEEE Protocol 802.3af)
  SPAN: source/destination
  UDLD: yes

```

```

Link Debounce:          no
Link Debounce Time:     no
Port Security:          yes
Dot1x:                  yes
Maximum MTU:            9198 bytes (Jumbo Frames)
Multiple Media Types:    no
Diagnostic Monitoring:   N/A

```

Use the **show interfaces counters storm-control** command to display a count of discarded packets.

```
Switch# show interfaces counters storm-control
```

Port	Broadcast	Multicast	Level	TotalSuppressedPackets
Fa3/1	Enabled	Disabled	10.00%	46516510
Gi2/1	Enabled	Enabled	50.00%	0

```
Switch# show storm-control
```

Interface	Filter State	Broadcast	Multicast	Level
Fa3/1	Blocking	Enabled	Disabled	10.00%
Gi2/1	Link Down	Enabled	Enabled	50.00%



Configuring SPAN and RSPAN

This chapter describes how to configure the Switched Port Analyzer (SPAN) and Remote SPAN (RSPAN) on the Catalyst 4500 series switches. SPAN selects network traffic for analysis by a network analyzer, such as a SwitchProbe device or other Remote Monitoring (RMON) probe.

This chapter consists of the following sections:

- [About SPAN and RSPAN, page 69-1](#)
- [Configuring SPAN, page 69-7](#)
- [CPU Port Sniffing, page 69-10](#)
- [Encapsulation Configuration, page 69-11](#)
- [Ingress Packets, page 69-12](#)
- [Access List Filtering, page 69-13](#)
- [Packet Type Filtering, page 69-14](#)
- [Configuration Example, page 69-15](#)
- [Configuring RSPAN, page 69-16](#)
- [Displaying SPAN and RSPAN Status, page 69-24](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About SPAN and RSPAN

This sections includes the following subsections:

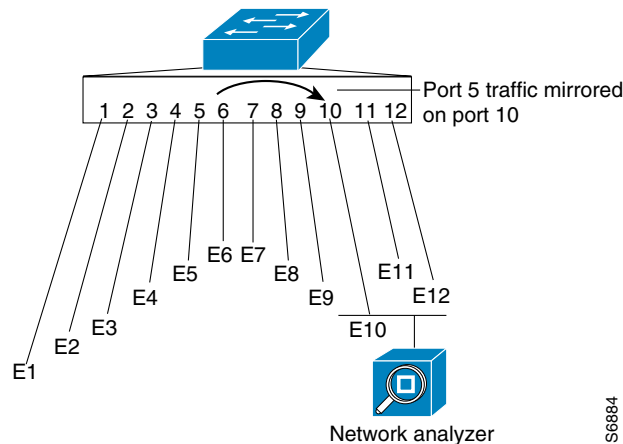
- [SPAN and RSPAN Concepts and Terminology, page 69-3](#)
- [SPAN and RSPAN Session Limits, page 69-6](#)
- [Default SPAN and RSPAN Configuration, page 69-6](#)

SPAN mirrors traffic from one or more source interfaces on any VLAN or from one or more VLANs to a destination interface for analysis. In [Figure 69-1](#), all traffic on Ethernet interface 5 (the source interface) is mirrored to Ethernet interface 10. A network analyzer on Ethernet interface 10 receives all network traffic from Ethernet interface 5 without being physically attached to it.

For SPAN configuration, the source interfaces and the destination interface must be on the same switch.

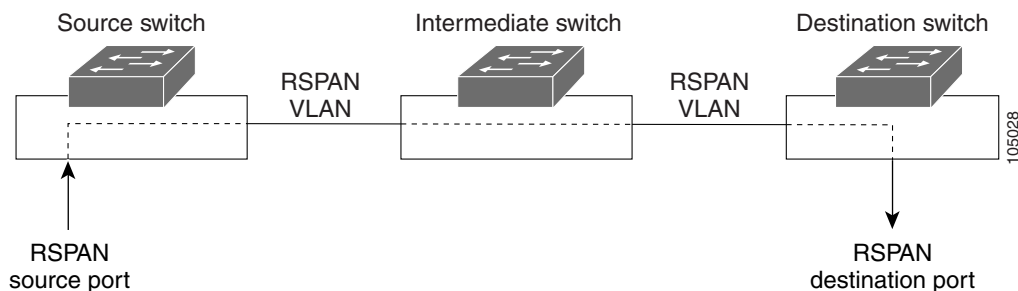
SPAN does not affect the switching of network traffic on source interfaces; copies of the packets received or transmitted by the source interfaces are sent to the destination interface.

Figure 69-1 Example SPAN Configuration



RSPAN extends SPAN by enabling remote monitoring of multiple switches across your network. The traffic for each RSPAN session is carried over a user-specified RSPAN VLAN that is dedicated for that RSPAN session in all participating switches. The SPAN traffic from the sources is copied onto the RSPAN VLAN and then forwarded over trunk ports that are carrying the RSPAN VLAN to any RSPAN destination sessions monitoring the RSPAN VLAN, as shown in [Figure 69-2](#).

Figure 69-2 Example of RSPAN Configuration



SPAN and RSPAN do not affect the switching of network traffic on source ports or source VLANs; a copy of the packets received or sent by the sources is sent to the destination. Except for traffic that is required for the SPAN or RSPAN session, by default, destination ports do not receive or forward traffic.

You can use the SPAN or RSPAN destination port to forward transmitted traffic from a network security device. For example, if you connect a Cisco Intrusion Detection System (IDS) sensor appliance to a destination port, the IDS device can send TCP reset packets to close down the TCP session of a suspected attacker.

SPAN and RSPAN Concepts and Terminology

This section describes concepts and terminology associated with SPAN and RSPAN configuration and includes the following subsections:

- [SPAN Session, page 69-3](#)
- [Traffic Types, page 69-3](#)
- [Source Port, page 69-4](#)
- [Destination Port, page 69-5](#)
- [VLAN-Based SPAN, page 69-5](#)
- [SPAN Traffic, page 69-6](#)

SPAN Session

A local SPAN session associates a destination port with source ports. You can monitor incoming or outgoing traffic on a series or range of ports and source VLANs. An RSPAN session associates source ports and source VLANs across your network with an RSPAN VLAN. The destination source is the RSPAN VLAN.

You configure SPAN sessions by using parameters that specify the source of network traffic to monitor.

You can configure multiple SPAN or RSPAN sessions with separate or overlapping sets of SPAN sources. Both switched and routed ports can be configured as SPAN sources or destination ports.

An RSPAN source session associates SPAN source ports or VLANs with a destination RSPAN VLAN. An RSPAN destination session associates an RSPAN VLAN with a destination port.

SPAN sessions do not interfere with the normal operation of the switch; however, an oversubscribed SPAN destination (for example, a 10-Mbps port monitoring a 100-Mbps port) results in dropped or lost packets.

You can configure SPAN sessions on disabled ports; however, a SPAN session does not become active unless you enable the destination port and at least one source port or VLAN for that session.

A SPAN session remains inactive after system startup until the destination port is operational.

Traffic Types

SPAN sessions include these traffic types:

- **Receive (Rx) SPAN**—The goal of receive (or ingress) SPAN is to monitor as much as possible all packets received by the source interface or VLAN before any modification or processing is performed by the switch. A copy of each packet received by the source is sent to the destination port for that SPAN session. You can monitor a series or range of ingress ports or VLANs in a SPAN session.

On tagged packets (Inter-Switch Link IEEE 802.1Q), the tagging is removed at the ingress port. At the destination port, if tagging is enabled, the packets appear with 802.1Q headers. If no tagging is specified, packets appear in the native format.

Packets that are modified because of routing are copied without modification for Rx SPAN; that is, the original packet is copied. Packets that are modified because of quality of service (QoS)—for example, modified Differentiated Services Code Point (DSCP)—are copied without modification for Rx SPAN.

Some features that can cause a packet to be dropped during receive processing have no effect on SPAN; the destination port receives a copy of the packet even if the actual incoming packet is dropped. These features include IP standard and extended input access control lists (ACLs), IP standard and extended output ACLs for unicast and ingress QoS policing, VLAN maps, ingress QoS policing, and policy-based routing. Switch congestion that causes packets to be dropped also has no effect on SPAN.

- **Transmit (Tx) SPAN**—The goal of transmit (or egress) SPAN is to monitor as much as possible all packets sent by the source interface after the switch performs all modification and processing. After the packet is modified, the source sends a copy of each packet to the destination port for that SPAN session. You can monitor a range of egress ports in a SPAN session.

Packets that are modified because of routing—for example, with a time-to-live (TTL) or MAC-address modification—are duplicated at the destination port. On packets that are modified because of QoS, the modified packet might not have the same DSCP (IP packet) or CoS (non-IP packet) as the SPAN source.

Some features that can cause a packet to be dropped during transmit processing might also affect the duplicated copy for SPAN. These features include VLAN maps, IP standard and extended output ACLs on multicast packets, and egress QoS policing. In the case of output ACLs, if the SPAN source drops the packet, the SPAN destination would also drop the packet. In the case of egress QoS policing, if the SPAN source drops the packet, the SPAN destination might not drop it. If the source port is oversubscribed, the destination ports have different dropping behavior.

- **Both**—In a SPAN session, you can monitor a single port series or a range of ports for both received and sent packets.

Source Port

A source port (also called a *monitored port*) is a switched or routed port that you monitor for network traffic analysis. In a single local SPAN session or RSPAN source session, you can monitor source port traffic, such as received (Rx), transmitted (Tx), or bidirectional (both). The switch supports any number of source ports (up to the maximum number of available ports on the switch) and any number of source VLANs.

A source port has these characteristics:

- It can be any port type (for example, EtherChannel, Fast Ethernet, Gigabit Ethernet, and so forth).
- It can be monitored in multiple SPAN sessions.
- It cannot be a destination port.
- Each source port can be configured with a direction (ingress, egress, or both) to monitor. For EtherChannel sources, the monitored direction would apply to all physical ports in the group.
- Source ports can be in the same or different VLANs.
- For VLAN SPAN sources, all active ports in the source VLAN are included as source ports.

You can configure a trunk port as a source port. By default, all VLANs active on the trunk are monitored. You can limit SPAN traffic monitoring on trunk source ports to specific VLANs by using VLAN filtering. Only switched traffic in the selected VLANs is sent to the destination port. This feature affects only traffic forwarded to the destination SPAN port and does not affect the switching of normal traffic. This feature is not allowed in sessions with VLAN sources.

Destination Port

Each local SPAN session or RSPAN destination session must have a destination port (also called a *monitoring port*) that receives a copy of traffic from the source ports and VLANs.

A destination port has these characteristics:

- A destination port must reside on the same switch as the source port (for a local SPAN session).
- A destination port can be any Ethernet physical port.
- A destination port can participate in only one SPAN session at a time. (A destination port in one SPAN session cannot be a destination port for a second SPAN session.)
- A destination port cannot be a source port.
- A destination port cannot be an EtherChannel group.
- A destination port can be a physical port that is assigned to an EtherChannel group, even if the EtherChannel group has been specified as a SPAN source. The port is removed from the group while it is configured as a SPAN destination port.
- The port does not transmit any traffic except that traffic required for the SPAN session unless learning is enabled. If learning is enabled, the port also transmits traffic directed to hosts that have been learned on the destination port.
- If ingress traffic forwarding is enabled for a network security device, the destination port forwards traffic at Layer 2.
- A destination port does not participate in spanning tree while the SPAN session is active.
- When it is a destination port, it does not participate in any of the Layer 2 protocols (STP, VTP, CDP, DTP, PagP).
- A destination port that belongs to a source VLAN of any SPAN session is excluded from the source list and is not monitored.
- A destination port receives copies of sent and received traffic for all monitored source ports. If a destination port is oversubscribed, it could become congested and result in packet drops at the destination port. This congestion does not affect traffic forwarding on the source ports.

VLAN-Based SPAN

VLAN-based SPAN (VSPAN) is the monitoring of the network traffic in one or more VLANs.

Use these guidelines for VSPAN sessions:

- Traffic on RSPAN VLANs is not monitored by VLAN-based SPAN sessions.
- Only traffic on the monitored VLAN is sent to the destination port.
- If a destination port belongs to a source VLAN, it is excluded from the source list and is not monitored.
- If ports are added to or removed from the source VLANs, the traffic on the source VLAN received by those ports is added to or removed from the sources being monitored.
- VLAN pruning and the VLAN allowed list have no effect on SPAN monitoring.

- VSPAN monitors only traffic that enters the switch, not traffic that is routed between VLANs. For example, if a VLAN is being Rx-monitored, and the multilayer switch routes traffic from another VLAN to the monitored VLAN, that traffic is not monitored and is not received on the SPAN destination port.
- You cannot use filter VLANs in the same session with VLAN sources.
- You can monitor only Ethernet VLANs.

SPAN Traffic

You can use local SPAN to monitor all network traffic, including multicast and bridge protocol data unit (BPDU) packets, Cisco Discovery Protocol (CDP), VLAN Trunk Protocol (VTP), Dynamic Trunking Protocol (DTP), Spanning Tree Protocol (STP), and Port Aggregation Protocol (PAgP) packets. You cannot use RSPAN to monitor Layer 2 protocols. See the [“RSPAN Configuration Guidelines” section on page 69-16](#) for more information.)

In some SPAN configurations, multiple copies of the same source packet are sent to the SPAN destination port. For example, a bidirectional (both Rx and Tx) SPAN session is configured for the sources a1 Rx monitor and the a2 Rx and Tx monitor to destination port d1. If a packet enters the switch through a1 and is switched to a2, both incoming and outgoing packets are sent to destination port d1. Both packets are the same (unless a Layer-3 rewrite occurs, in which case the packets are different because of the added Layer 3 information).

SPAN and RSPAN Session Limits

You can configure a maximum of sixteen SPAN/RSPAN sessions (eight concurrent sessions with ingress-only sources and eight concurrent sessions with egress-only sources). Bidirectional sources count as both ingress and egress. RSPAN destination sessions count as a session containing an ingress source.

Default SPAN and RSPAN Configuration

[Table 69-1](#) shows the default SPAN and RSPAN configuration.

Table 69-1 *Default SPAN and RSPAN Configuration*

Feature	Default Setting
SPAN state	Disabled.
Source port traffic to monitor	Both received and sent traffic (both).
Filters	All VLANs, all packet types, all address types.
Encapsulation type (destination port)	Native form (no encapsulation type header).
Ingress forwarding (destination port)	Disabled.
Host learning (destination port)	Disabled.

Configuring SPAN

The following sections describe how to configure SPAN:

- [SPAN Configuration Guidelines and Restrictions, page 69-7](#)
- [Configuring SPAN Sources, page 69-8](#)
- [Configuring SPAN Destinations, page 69-9](#)
- [Monitoring Source VLANs on a Trunk Interface, page 69-9](#)
- [Configuration Scenario, page 69-10](#)
- [Verifying a SPAN Configuration, page 69-10](#)

**Note**

Entering SPAN configuration commands does not clear previously configured SPAN parameters. You must enter the **no monitor session** command to clear configured SPAN parameters.

SPAN Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when configuring SPAN:

- You must use a network analyzer to monitor interfaces.
- You cannot mix source VLANs and filter VLANs within a SPAN session. You can have source VLANs or filter VLANs, but not both at the same time.
- EtherChannel interfaces can be SPAN source interfaces; they cannot be SPAN destination interfaces.
- When you specify source interfaces and do not specify a traffic type (Tx, Rx, or both), “both” is used by default. To change from both to either “tx” or “rx,” unconfigure the corresponding other type “rx” or “tx” with the **no monitor session {session_number} {source {interface interface_list | {vlan vlan_ids | cpu [queue queue_ids] } {rx | tx}}** command.
- If you specify multiple SPAN source interfaces, the interfaces can belong to different VLANs.
- You must enter the **no monitor session number** command with no other parameters to clear the SPAN session *number*.
- The **no monitor** command clears all SPAN sessions.
- SPAN destinations never participate in any spanning tree instance. SPAN includes BPDUs in the monitored traffic, so any BPDUs seen on the SPAN destination are from the SPAN source.
- SPAN is limited to one destination port per session.
- When you create a SPAN session, it sets the packet filter to **good** automatically (default) and hence you will see another configuration line:

monitor session 1 filter packet-type good rx

To remove or change this filter, first enter the **no monitor session 1 filter packet-type good rx** and then configure the kind of span filter you want to. But do not reload the switch after this. Reloading the switches automatically re-creates the default configuration, and does not delete user-configuration. Having both **good** and **bad** packet types configured, disables the filtering function.

Configuring SPAN Sources

To configure the source for a SPAN session, perform this task:

Command	Purpose
<pre>Switch(config)# [no] monitor session {session_number} {source {interface interface_list {vlan vlan_IDs cpu [queue queue_ids] } [rx tx both]}</pre>	<p>Specifies the SPAN session number (1 through 6), the source interfaces (FastEthernet or GigabitEthernet), VLANs (1 through 4094), whether traffic received or sent from the CPU is copied to the session destination, and the traffic direction to be monitored.</p> <p>For <i>session_number</i>, specifies the session number identified with this RSPAN session (1 through 6).</p> <p>For <i>interface-list</i>, specifies the source port to monitor. Valid interfaces include physical interfaces and port-channel logical interfaces (port-channel <i>port-channel-number</i>).</p> <p>For <i>vlan_IDs</i>, specifies the source VLAN.</p> <p>For <i>queue_ids</i>, specifies the queue(s) involved.</p> <p>(Optional) [, l -] Specifies a series or range of interfaces. Enter a space after the comma; enter a space before and after the hyphen.</p> <p>(Optional) Specifies the direction of traffic to monitor. If you do not specify a traffic direction, the source interface sends both transmitted (Tx) and received (Rx) traffic. Only received traffic can be monitored on additional source ports.</p> <ul style="list-style-type: none"> • Rx—Monitor received traffic. • Tx—Monitor transmitted traffic. • both—Monitor both received and transmitted traffic (bidirectional). <p>Queues may be identified either by number or by name. Queue names may subsume multiple numbered queues for convenience.</p> <p>Use the no keyword to restore the defaults.</p>

This example shows how to configure SPAN session 1 to monitor bidirectional traffic from source interface Fast Ethernet 5/1:

```
Switch(config)# monitor session 1 source interface fastethernet 5/1
```

This example shows how to configure sources with differing directions within a SPAN session:

```
Switch(config)# monitor session 1 source interface fa2/3 rx
Switch(config)# monitor session 1 source interface fa2/2 tx
Switch(config)#
```

Configuring SPAN Destinations

To configure the destination for a SPAN session, perform this task:

Command	Purpose
Switch(config)# [no] monitor session <session_number> destination interface <interface> [encapsulation {dot1q}] [ingress [vlan vlan_IDs] [learning]]	Specifies the SPAN session number (1 through 6) and the destination interfaces or VLANs. For <i>session_number</i> , specifies the session number identified with this RSPAN session (1 through 6). For <i>interface</i> , specifies the destination interface. For <i>vlan_IDs</i> , specifies the destination VLAN. Use the no keyword to restore the defaults.



Note

SPAN is limited to one destination port per session.

This example shows how to configure interface Fast Ethernet 5/48 as the destination for SPAN session 1:

```
Switch(config)# monitor session 1 destination interface fastethernet 5/48
```

Monitoring Source VLANs on a Trunk Interface

To monitor specific VLANs when the SPAN source is a trunk interface, perform this task:

Command	Purpose
Switch(config)# [no] monitor session {session_number} filter {vlan vlan_IDs [, -]} {packet-type {good bad}} {address-type {unicast multicast broadcast} [rx tx both]}	Monitors specific VLANs when the SPAN source is a trunk interface. The filter keyword restricts monitoring to traffic that is on the specified VLANs; it is typically used when monitoring a trunk interface. For <i>session_number</i> , specifies the session number identified with this RSPAN session (1 through 6). For <i>vlan_IDs</i> , specifies the VLAN. Monitoring is established through all the ports in the specified VLANs Use the no keyword to restore the defaults.

This example shows how to monitor VLANs 1 through 5 and VLAN 9 when the SPAN source is a trunk interface:

```
Switch(config)# monitor session 2 filter vlan 1 - 5 , 9
```

Configuration Scenario

This example shows how to use the commands described in this chapter to completely configure and unconfigure a span session. Assume that you want to monitor bidirectional traffic from source interfaces Fast Ethernet 4/10, 4/11 and 4/12, Interface 4/10 is configured as a trunk interface carrying VLANs 1 through 4094. Interface Fast Ethernet 4/11 is configured as an access port in VLAN 57 and interface Fast Ethernet 4/12 is configured as an access port in VLAN 58. You want to monitor only traffic in VLAN 57 in that session. Using Fast Ethernet 4/15 as your destination interface, you would enter the following commands:

```
Switch(config)# monitor session 1 source interface fastethernet 4/10 - 12
Switch(config)# monitor session 1 filter vlan 57
Switch(config)# monitor session 1 destination interface fastethernet 4/15
```

You are now monitoring traffic from interface Fast Ethernet 4/10 that is on VLAN 57 out of interface FastEthernet 4/15. To disable the span session enter the following command:

```
Switch(config)# no monitor session 1
```

Verifying a SPAN Configuration

This example shows how to verify the configuration of SPAN session 2:

```
Switch# show monitor session 2
Session 2
-----
Source Ports:
  RX Only:      Fa5/12
  TX Only:      None
  Both:         None
Source VLANs:
  RX Only:      None
  TX Only:      None
  Both:         None
Destination Ports: Fa5/45
Filter VLANs:    1-5,9
Switch#
```

CPU Port Sniffing

When configuring a SPAN session, you can specify the CPU (or a subset of CPU queues) as a SPAN source. Queues may be specified either by number or by name. When such a source is specified, traffic going to the CPU through one of the specified queues is mirrored and sent out of the SPAN destination port in the session. This traffic includes both control packets and regular data packets that are sent to or from the CPU (due to software forwarding).

You can mix the CPU source with either regular port sources or VLAN sources.

To configure CPU source sniffing, perform this task:

Command	Purpose
<pre>Switch(config)# [no] monitor session {session_number} {source {interface interface_list {vlan vlan_ids cpu [queue queue_ids] } [rx tx both]}</pre>	<p>Specifies that the CPU causes traffic received by or sent from the CPU to be copied to the destination of the session. The queue identifier optionally allows sniffing-only traffic (received) on the specified CPU queue(s).</p> <p>For <i>session_number</i>, specifies the session number identified with this SPAN session (1 through 6).</p> <p>For <i>interface-list</i>, specifies the source port to monitor. Valid interfaces include physical interfaces and port-channel logical interfaces (port-channel <i>port-channel-number</i>).</p> <p>For <i>vlan_ids</i>, specifies the source VLAN.</p> <p>For <i>queue_ids</i>, specifies the queue(s) involved.</p> <p>(Optional) [, -] Specifies a series or range of interfaces. Enter a space after the comma; enter a space before and after the hyphen.</p> <p>(Optional) Specifies the direction of traffic to monitor. If you do not specify a traffic direction, the source interface sends both transmitted (Tx) and received (Rx) traffic. Only received traffic can be monitored on additional source ports.</p> <ul style="list-style-type: none"> • Rx—Monitor received traffic. • Tx—Monitor transmitted traffic. • both—Monitor both received and transmitted traffic (bidirectional). <p>Queues may be identified either by number or by name. Queue names may subsume multiple numbered queues for convenience.</p> <p>Use the no keyword to restore the defaults.</p>

This example shows how to configure a CPU source to sniff all packets received by the CPU:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# monitor session 1 source cpu rx
```

Encapsulation Configuration

When configuring a SPAN destination port, you can explicitly specify the encapsulation type used by the port. Packets sent out the port are tagged in accordance with the specified mode. (The encapsulation mode also controls how tagged packets are handled when the ingress packet option is enabled.) The Catalyst 4500 series switch supervisor engines support 802.1q encapsulation and untagged packets.

**Note**

Only 802.1q encapsulation is supported.

The “replicate” encapsulation type (in which packets are transmitted from the destination port using whatever encapsulation applied to the original packet) is not supported. If no encapsulation mode is specified, the port default is untagged.

Ingress Packets

When ingress is enabled, the SPAN destination port accepts incoming packets (potentially tagged depending on the specified encapsulation mode) and switches them normally. When configuring a SPAN destination port, you can specify whether the ingress feature is enabled and what VLAN to use to switch untagged ingress packets. Although the port is STP forwarding, it does not participate in the STP, so use caution when configuring this feature lest a spanning-tree loop be introduced in the network. When both ingress and a trunk encapsulation are specified on a SPAN destination port, the port goes forwarding in all active VLANs. Configuring a non-existent VLAN as an ingress VLAN is not allowed.

By default, host learning is disabled on SPAN destination ports with ingress enabled. The port is also removed from VLAN floodsets, so regular traffic is not switched out of the destination port. If learning is enabled, then traffic for hosts learned on the destination port is switched out the destination port. A host connected to the SPAN destination port will not receive broadcast ARP requests and will not respond. You can also configure static host entries (including a static ARP entry and a static entry in the MAC-address table) on SPAN destination ports.

**Note**

This configuration does not work if the SPAN session does not have a source configured; the session is half configured with only the SPAN destination port.

To configure ingress packets and encapsulation, perform this task:

Command	Purpose
Switch(config)# [no] monitor session <i>session_number</i> destination interface <i>interface</i> [encapsulation {dot1q}] [ingress [vlan <i>vlan_IDs] [learning]</i>	Specifies the configuration of the ingress packet and the encapsulation type of the destination port. For <i>session_number</i> , specifies the session number identified with this SPAN session (1 through 6). For <i>interface</i> , specifies the destination interface. For <i>vlan_IDs</i> , specifies the destination VLAN. Use the no keyword to restore the defaults.

This example shows how to configure a destination port with 802.1q encapsulation and ingress packets using native VLAN 7:

```
Switch(config)# monitor session 1 destination interface fastethernet 5/48
encapsulation dot1q ingress vlan 7
```


With this configuration, traffic from SPAN sources associated with session 1 would be copied out of interface Fast Ethernet 5/48, with 802.1q encapsulation. Incoming traffic would be accepted and switched, with untagged packets being classified into VLAN 7.

Access List Filtering

When configuring a SPAN session, you can apply access list filtering. Access list filtering applies to all packets passing through a SPAN destination port that might be sniffed in the egress or ingress direction. Access list filters are allowed on local SPAN sessions only. If the SPAN destination is an RSPAN VLAN, the access list filter is rejected.



Note

Access list filtering is available in Cisco IOS Release 12.2(20)EW and later releases.

ACL Configuration Guidelines

You can configure ACLs on a SPAN session. Use these guidelines for ACL/SPAN sessions:

- If an ACL is associated with a SPAN session, the rules associated with that ACL are applied against all packets exiting the SPAN destination interface. Rules pertaining to other VACLs or RACLs previously associated with the SPAN destination interface are not applied.
- Only one IP named ACL and one IPv6 ACL can be associated with a SPAN session.
- When no ACLs are applied to packets exiting a SPAN destination interface, all traffic is permitted regardless of the PACLs, VACLs, or RACLs that have been previously applied to the destination interface or VLAN to which the SPAN destination interface belongs.
- If an ACL is removed from a SPAN session, all traffic is permitted once again.
- If SPAN configuration is removed from the SPAN session, all rules associated with the SPAN destination interface are applied once again.
- If a SPAN destination port is configured as a trunk port and the VLANs to which it belongs have ACLs associated with them, the traffic is not subjected to the VACLs.
- ACL configuration applies normally to the RSPAN VLAN and to trunk ports carrying the RSPAN VLAN. This configuration enables you to apply VACLs on RSPAN VLANs. If a user attempts to configure an ACL on a SPAN session with the destination port as an RSPAN VLAN, the configuration is rejected.
- If CAM resources are exhausted and packets are passed to the CPU for lookup, any output port ACLs associated with a SPAN session are not applied.
- If a named IP ACL or IPv6 ACL is configured on a SPAN session before an ACL is created, the configuration is accepted, and the software creates an empty ACL with no ACEs. (An empty ACL permits all packets.) Subsequently, the rules can be added to the ACL.
- The ACLs associated with a SPAN session are applied on the destination interface on output.
- No policing is allowed on traffic exiting SPAN ports.
- IP and IPv6 ACLs are supported on SPAN sessions.

Configuring Access List Filtering

To configure access list filtering, perform this task:

Command	Purpose
Switch(config)# [no] monitor session { <i>session_number</i> } filter { ip ipv6 } access-group [<i>name</i> <i>id</i>] [{ vlan <i>vlan_IDs</i> [, -]}] [{ packet-type { good bad }} { address-type { unicast multicast broadcast } [rx tx both }]	Specifies filter sniffing based on the access list. For <i>session_number</i> , specify the session number identified with this SPAN session (1 through 6). You can specify either a name or a numeric ID for the access list. For <i>name</i> , specify the IP access list name. For <i>id</i> , specify a standard (1 to 199) or extended (1300-2699) IP access list.



Note

IP and IPv6 access lists must be created in configuration mode as described in the chapter “Configuring Network Security with ACLs.”

This example shows how to configure IP access group 10 on a SPAN session and verify that an access list has been configured:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# monitor session 1 source interface fa6/1 both
Switch(config)# monitor session 1 destination interface fa6/2
Switch(config)# monitor session 1 filter vlan 1
Switch(config)# monitor session 1 filter ip access-group 10
Switch(config)# exit
Switch# show monitor
```

```
Session 1
-----
Type                : Local Session
Source Ports        :
    Both            : Fa6/1
Destination Ports   : Fa6/2
    Encapsulation   : Native
    Ingress         : Disabled
    Learning        : Disabled
Filter VLANs        : 1
IP Access-group     : 10
```

Packet Type Filtering

When configuring a SPAN session, you can specify packet filter parameters similar to VLAN filters. When specified, the packet filters indicate types of packets that may be sniffed. If no packet filters are specified, packets of all types may be sniffed. Different types of packet filters may be specified for ingress and egress traffic.

The two categories of packet filtering are packet-based (good, error) or address-based (unicast/multicast/broadcast). Packet-based filters can only be applied in the ingress direction. Packets are classified as broadcast, multicast, or unicast by the hardware based on the destination address.

**Note**

When filters of both types are configured, only packets that pass both filters are spanned. For example, if you set both “error” and “multicast,” only multicast packets with errors are spanned.

To configure packet type filtering, perform this task:

Command	Purpose
Switch(config)# [no] monitor session { <i>session_number</i> } filter { vlan <i>vlan_IDs</i> [, -] } { packet-type { good bad } { address-type { unicast multicast broadcast } [rx tx both]}	Specifies filter sniffing of the specified packet types in the specified directions. For <i>session_number</i> , specifies the session number identified with this SPAN session (1 through 6). For <i>vlan_IDs</i> , specifies the VLAN. You can specify both Rx and Tx type filters, as well as specify multiple type filters at the same time (such as good and unicast to only sniff non-error unicast frames). As with VLAN filters, if no type or filter is specified, then the session sniffs all packet types. Use the no keyword to restore the defaults.

This example shows how to configure a session to accept only unicast packets in the ingress direction:

```
Switch(config)# monitor session 1 filter address-type unicast rx
```

Configuration Example

The following is an example of SPAN configuration using some of the SPAN enhancements.

In this example, you configure a session to sniff unicast traffic arriving on interface Gi1/1.

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# monitor session 1 source interface gi1/1 rx
Switch(config)# monitor session 1 destination interface gi1/2 encapsulation dot1q ingress
Switch(config)# monitor session 1 filter address-type unicast rx
Switch(config)# exit
Switch# show monitor

Session 1
-----
Type                : Local Session
Source Ports        :
    RX Only         : Gi1/1
Destination Ports   : Gi1/2
    Encapsulation   : DOT1Q
    Ingress         : Enabled
    Learning        : Disabled
Filter Addr Type    :
    RX Only         : Unicast
```

Configuring RSPAN

This section describes how to configure RSPAN on your switch and it contains this configuration information:

- [RSPAN Configuration Guidelines, page 69-16](#)
- [Creating an RSPAN Session, page 69-17](#)
- [Creating an RSPAN Destination Session, page 69-18](#)
- [Creating an RSPAN Destination Session and Enabling Ingress Traffic, page 69-19](#)
- [Removing Ports from an RSPAN Session, page 69-20](#)
- [Specifying VLANs to Monitor, page 69-21](#)
- [Specifying VLANs to Filter, page 69-23](#)

RSPAN Configuration Guidelines

Follow these guidelines when configuring RSPAN:

**Note**

Since RSPAN VLANs have special properties, you should reserve a few VLANs across your network for use as RSPAN VLANs; do not assign access ports to these VLANs.

**Note**

You can apply an output access control list (ACL) to RSPAN traffic to selectively filter or monitor specific packets. Specify these ACLs on the RSPAN VLAN in the RSPAN source switches.

- RSPAN sessions can coexist with SPAN sessions within the limits described in the “[SPAN and RSPAN Session Limits](#)” section on page 69-6.
- For RSPAN configuration, you can distribute the source ports and the destination ports across multiple switches in your network.
- RSPAN does not support BPDU packet monitoring or other Layer 2 switch protocols.
- The RSPAN VLAN is configured only on trunk ports and not on access ports. To avoid unwanted traffic in RSPAN VLANs, make sure that all participating switches support the VLAN remote-span feature. Access ports on the RSPAN VLAN are silently disabled.
- You should create an RSPAN VLAN before configuring an RSPAN source or destination session.
- If you enable VTP and VTP pruning, RSPAN traffic is pruned in the trunks to prevent the unwanted flooding of RSPAN traffic across the network for VLAN-IDs that are lower than 1005.
- Because RSPAN traffic is carried across a network on an RSPAN VLAN, the original VLAN association of the mirrored packets is lost. RSPAN can only support forwarding of traffic from an IDS device onto a single user-specified VLAN.

Creating an RSPAN Session

First create an RSPAN VLAN that *does not* exist for the RSPAN session in any of the switches that participate in RSPAN. With VTP enabled in the network, you can create the RSPAN VLAN in one switch, and then VTP propagates it to the other switches in the VTP domain for VLAN-IDs that are lower than 1005.

Use VTP pruning to get efficient flow of RSPAN traffic, or manually delete the RSPAN VLAN from all trunks that do not need to carry the RSPAN traffic.

To start an RSPAN source session and to specify the source (monitored) ports and the destination RSPAN VLAN, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# no monitor session { <i>session_number</i> all local remote }	Clears any existing RSPAN configuration for the session. For <i>session_number</i> , specifies the session number identified with this RSPAN session (1 through 6). Specifies all to remove all RSPAN sessions, local to remove all local sessions, or remote to remove all remote SPAN sessions.
Step 3	Switch(config)# vlan { <i>remote_vlan_ID</i> }	Specifies a remote VLAN ID. Ensure that the VLAN ID is not being used for any user traffic.
Step 4	Switch(config-vlan)# remote-span	Converts the VLAN ID to a remote VLAN ID.
Step 5	Switch(config-vlan)# exit	Returns to global configuration mode.
Step 6	Switch(config)# [no] monitor session { <i>session_number</i> } { source { interface <i>interface_list</i> { vlan <i>vlan_IDs</i> cpu [<i>queue queue_ids</i>]}} [rx tx both]	Specifies the RSPAN session and the source port (monitored port). For <i>session_number</i> , specifies the session number identified with this RSPAN session (1 through 6). For <i>interface-list</i> , specifies the source port to monitor. Valid interfaces include physical interfaces and port-channel logical interfaces (port-channel <i>port-channel-number</i>). For <i>vlan-IDs</i> , specifies the source VLAN or VLANs to monitor. Valid VLANs are in the range from 1 to 4094. For <i>queue_ids</i> , specifies either a set of CPU queue numerical identifiers from 1 to 32, or a named queue. (Optional) [, -] Specifies a series or range of interfaces. Enter a space after the comma; enter a space before and after the hyphen. (Optional) Specifies the direction of traffic to monitor. If you do not specify a traffic direction, the source interface sends both transmitted (Tx) and received (Rx) traffic. Only received traffic can be monitored on additional source ports. <ul style="list-style-type: none"> • Rx—Monitor received traffic. • Tx—Monitor transmitted traffic. • both—Monitor both received and transmitted traffic (bidirectional).

	Command	Purpose
Step 7	Switch(config)# monitor session <i>session_number</i> destination remote vlan <i>vlan-ID</i>	Specifies the RSPAN session and the destination remote VLAN. For <i>session_number</i> , specifies the session number identified with this RSPAN session (1 through 6). For <i>vlan-ID</i> , specifies the RSPAN VLAN to carry the monitored traffic to the destination port.
Step 8	Switch(config)# end	Returns to privileged EXEC mode.
Step 9	Switch# show monitor [session <i>session_number</i>]	Verifies your entries.
Step 10	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to clear any existing RSPAN configuration for session 1, configure RSPAN session 1 to monitor multiple source interfaces, and configure the destination RSPAN VLAN.

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# no monitor session 1
Switch(config)# monitor session 1 source interface fastEthernet3/10 tx
Switch(config)# monitor session 1 source interface fastEthernet3/2 rx
Switch(config)# monitor session 1 source interface fastEthernet3/3 rx
Switch(config)# monitor session 1 source interface port-channel 102 rx
Switch(config)# monitor session 1 destination remote vlan 901
Switch(config)# end
```

Creating an RSPAN Destination Session

To create an RSPAN destination session and to specify the source RSPAN VLAN and the destination port, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# monitor session <i>session_number</i> source remote vlan <i>vlan-ID</i>	Specifies the RSPAN session and the source RSPAN VLAN. For <i>session_number</i> , specifies the session number identified with this RSPAN session (1 through 6). For <i>vlan-ID</i> , specifies the source RSPAN VLAN to monitor.

	Command	Purpose
Step 3	Switch(config)# [no] monitor session <i>session_number</i> destination interface <i>interface</i> [encapsulation {dot1q}] [ingress [vlan <i>vlan_IDs</i>] [learning]]	Specifies the RSPAN session and the destination interface. For <i>session_number</i> , specifies the session number identified with this RSPAN session (1 through 6). For <i>interface</i> , specifies the destination interface. For <i>vlan_IDs</i> , specifies the ingress VLAN, if necessary. (Optional) [, -] Specifies a series or range of interfaces. Enter a space after the comma; enter a space before and after the hyphen. (Optional) Specifies the direction of traffic to monitor. If you do not specify a traffic direction, the source interface sends both sent and received traffic. Only received (rx) traffic can be monitored on additional source ports. <ul style="list-style-type: none"> dot1q—Use 802.1Q encapsulation.
Step 4	Switch(config)# end	Returns to privileged EXEC mode.
Step 5	Switch# show monitor [session <i>session_number</i>]	Verifies your entries.
Step 6	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to configure VLAN 901 as the source remote VLAN and port 5 as the destination interface:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# monitor session 1 source remote vlan 901
Switch(config)# monitor session 1 destination interface gigabitEthernet1/2
Switch(config)# end
```

Creating an RSPAN Destination Session and Enabling Ingress Traffic

To create an RSPAN destination session, to specify the source RSPAN VLAN, and to enable ingress traffic on the destination port for a network security device (such as a Cisco IDS [Intrusion Detection System] sensor appliance), perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# monitor session { <i>session_number</i> } source vlan <i>vlan_IDs</i>	Specifies the RSPAN session and the source RSPAN VLAN. For <i>session_number</i> , specifies the session number identified with this RSPAN session (1 through 6). For <i>vlan_IDs</i> , specifies the source VLAN or VLANs to monitor. Valid VLANs are in the range from 1 to 4094.

	Command	Purpose
Step 3	Switch(config)# monitor session <i>session_number</i> destination interface <i>interface-id</i> [encapsulation { dot1q ingress vlan <i>vlan id</i> } ingress vlan <i>vlan id</i>] [learning]	<p>Specifies the RSPAN session, the destination port, the packet encapsulation, and the ingress VLAN.</p> <p>For <i>session_number</i>, specifies the session number identified with this RSPAN session (1 through 6).</p> <p>For <i>interface-id</i>, specifies the destination port. Valid interfaces include physical interfaces.</p> <p>(Optional) Specifies the encapsulation of the packets transmitted on the RSPAN destination port. If no encapsulation is specified, all transmitted packets are sent in native format (untagged).</p> <ul style="list-style-type: none"> Enter encapsulation dot1q to send native VLAN packets untagged, and all other VLAN tx packets tagged dot1q. <p>(Optional) Specifies whether forwarding is enabled for ingress traffic on the RSPAN destination port.</p> <ul style="list-style-type: none"> For native (untagged) and dot1q encapsulation, specify ingress vlan <i>vlan id</i> to enable ingress forwarding with <i>vlan id</i> as the native VLAN; <i>vlan id</i> is also used as the native VLAN for transmitted packets. Specify learning to enable learning when ingress is enabled.
Step 4	Switch(config)# end	Returns to privileged EXEC mode.
Step 5	Switch# show monitor [session <i>session_number</i>]	Verifies your entries.
Step 6	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to configure VLAN 901 as the source remote VLAN and how to configure the destination port for ingress traffic on VLAN 5 by using a security device that supports 802.1Q encapsulation:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# monitor session 1 source remote vlan 901
Switch(config)# monitor session 1 destination interface gigabitEthernet1/2 ingress vlan 5
Switch(config)# end
```

Removing Ports from an RSPAN Session

To remove a port as an RSPAN source for a session, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# [no] monitor session {session_number} {source {interface interface_list {vlan vlan_ids cpu [queue queue_ids]} [rx tx both]	Specifies the characteristics of the RSPAN source port (monitored port) to remove. For <i>session_number</i> , specifies the session number identified with this RSPAN session (1 through 6). For <i>interface-list</i> , specifies the source port to no longer monitor. Valid interfaces include physical interfaces and port-channel logical interfaces (port-channel <i>port-channel-number</i>). For <i>vlan_ids</i> , specifies the source vlan or vlans to monitor. Valid VLANs are in the range from 1 to 4094. For <i>queue_ids</i> , specifies either a set of CPU queue numerical identifiers from 1 to 32, or a named queue. (Optional) [, -] Specifies a series or range of interfaces. Enter a space after the comma; enter a space before and after the hyphen. (Optional) Specifies the direction of traffic to monitor. If you do not specify a traffic direction, the source interface sends both transmitted (Tx) and received (Rx) traffic. Only received traffic can be monitored on additional source ports. <ul style="list-style-type: none"> • Rx—Monitor received traffic. • Tx—Monitor transmitted traffic. • both—Monitor both received and transmitted traffic (bidirectional).
Step 3	Switch(config)# end	Returns to privileged EXEC mode.
Step 4	Switch# show monitor [session session_number]	Verifies your entries.
Step 5	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to remove port 1 as an RSPAN source for RSPAN session 1:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# no monitor session 1 source interface gigabitEthernet1/1
Switch(config)# end
```

This example shows how to disable received traffic monitoring on port 1, which was configured for bidirectional monitoring:

```
Switch(config)# no monitor session 1 source interface gigabitEthernet1/1 rx
```

The monitoring of traffic received on port 1 is disabled, but traffic transmitted from this port continues to be monitored.

Specifying VLANs to Monitor

VLAN monitoring is similar to port monitoring. To specify VLANs to monitor, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# no monitor session { <i>session_number</i> all local remote }	<p>Clears any existing SPAN configuration for the session.</p> <p>For <i>session_number</i>, specifies the session number identified with this RSPAN session (1 through 6).</p> <p>Specify all to remove all SPAN sessions, local to remove all local sessions, or remote to remove all remote SPAN sessions.</p>
Step 3	Switch(config)# [no] monitor session { <i>session_number</i> } { source { interface <i>interface_list</i> { vlan <i>vlan_ids</i> cpu [<i>queue queue_ids</i>]}} [rx tx both]	<p>Specifies the RSPAN session and the source VLANs (monitored VLANs). You can monitor only received (rx) traffic on VLANs.</p> <p>For <i>session_number</i>, specifies the session number identified with this RSPAN session (1 through 6).</p> <p>For <i>interface-list</i>, specifies the source port to no longer monitor. Valid interfaces include physical interfaces and port-channel logical interfaces (port-channel <i>port-channel-number</i>).</p> <p>For <i>vlan-ids</i>, the range is 1 to 4094; do not enter leading zeros.</p> <p>For <i>queue_ids</i>, specifies the source queue.</p> <p>(Optional) [rx tx both] Specifies a series or range of interfaces. Enter a space after the comma; enter a space before and after the hyphen.</p> <p>(Optional) Specifies the direction of traffic to monitor. If you do not specify a traffic direction, the source interface sends both transmitted (Tx) and received (Rx) traffic. Only received traffic can be monitored on additional source ports.</p> <ul style="list-style-type: none"> • Rx—Monitor received traffic. • Tx—Monitor transmitted traffic. • both—Monitor both received and transmitted traffic (bidirectional).
Step 4	Switch(config)# monitor session <i>session_number</i> destination remote vlan <i>vlan-id</i>	<p>Specifies the RSPAN session, the destination remote VLAN.</p> <p>For <i>session_number</i>, specifies the session number identified with this RSPAN session (1 through 6).</p> <p>For <i>vlan-id</i>, specifies the RSPAN VLAN to carry the monitored traffic to the destination port.</p>
Step 5	Switch(config)# end	Returns to privileged EXEC mode.
Step 6	Switch# show monitor [session <i>session_number</i>]	Verifies your entries.
Step 7	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To remove one or more source VLANs from the RSPAN session, use the **no monitor session session_number source vlan vlan-id rx** global configuration command.

This example shows how to clear any existing configuration on RSPAN session 2, configure RSPAN session 2 to monitor received traffic on all ports belonging to VLANs 1 through 3, and send it to destination remote VLAN 902. The configuration is then modified to also monitor received traffic on all ports belonging to VLAN 10.

```

Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# no monitor session 2
Switch(config)# monitor session 2 source vlan 1 - 3 rx
Switch(config)# monitor session 2 destination remote vlan 902
Switch(config)# monitor session 2 source vlan 10 rx
Switch(config)# end

```

Specifying VLANs to Filter

To limit RSPAN source traffic to specific VLANs, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# no monitor session { <i>session_number</i> all local remote }	<p>Clears any existing SPAN configuration for the session.</p> <p>For <i>session_number</i>, specifies the session number identified with this RSPAN session (1 through 6).</p> <p>Specify all to remove all SPAN sessions, local to remove all local sessions, or remote to remove all remote SPAN sessions.</p>
Step 3	Switch(config)# [no] monitor session { <i>session_number</i> } { source { interface <i>interface_list</i> { vlan <i>vlan_IDs</i> cpu [<i>queue</i> <i>queue_ids</i>]}} [rx tx both]	<p>Specifies the characteristics of the source port (monitored port) and RSPAN session.</p> <p>For <i>session_number</i>, specifies the session number identified with this RSPAN session (1 through 6).</p> <p>For <i>interface-list</i>, specifies the source port to monitor. The interface specified must already be configured as a trunk port.</p> <p>For <i>vlan-IDs</i>, the range is 1 to 4094; do not enter leading zeros.</p> <p>For <i>queue_ids</i>, specifies the source queue.</p> <p>(Optional) [, -] Specifies a series or range of interfaces. Enter a space after the comma; enter a space before and after the hyphen.</p> <p>(Optional) Specifies the direction of traffic to monitor. If you do not specify a traffic direction, the source interface sends both transmitted (Tx) and received (Rx) traffic. Only received traffic can be monitored on additional source ports.</p> <ul style="list-style-type: none"> • Rx—Monitor received traffic. • Tx—Monitor transmitted traffic. • both—Monitor both received and transmitted traffic (bidirectional).
Step 4	Switch(config)# monitor session <i>session_number</i> filter vlan <i>vlan-id</i> [, -]	<p>Limits the RSPAN source traffic to specific VLANs.</p> <p>For <i>session_number</i>, specifies the session number identified with this RSPAN session (1 through 6).</p> <p>For <i>vlan-id</i>, the range is 1 to 4094; do not enter leading zeros.</p> <p>(Optional) Use a comma (,) to specify a series of VLANs or use a hyphen (-) to specify a range of VLANs. Enter a space after the comma; enter a space before and after the hyphen.</p>

	Command	Purpose
Step 5	Switch(config)# monitor session <i>session_number</i> destination remote vlan <i>vlan-id</i>	Specifies the RSPAN session, the destination remote VLAN. For <i>session_number</i> , specifies the session number identified with this RSPAN session (1 through 6). For <i>vlan-id</i> , specifies the RSPAN VLAN to carry the monitored traffic to the destination port.
Step 6	Switch(config)# end	Returns to privileged EXEC mode.
Step 7	Switch# show monitor [session <i>session_number</i>]	Verifies your entries.
Step 8	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To monitor all VLANs on the trunk port, use the **no monitor session** *session_number* **filter vlan** global configuration command.

This example shows how to clear any existing configuration on RSPAN session 2, configure RSPAN session 2 to monitor traffic received on trunk port 4, and send traffic for only VLANs 1 through 5 and 9 to destination remote VLAN 902.

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# no monitor session 2
Switch(config)# monitor session 2 source interface gigabitethernet1/1 rx
Switch(config)# monitor session 2 filter vlan 1 - 5 , 9
Switch(config)# monitor session 2 destination remote vlan 902
Switch(config)# end
```

Displaying SPAN and RSPAN Status

To display the status of the current SPAN or RSPAN configuration, use the **show monitor** privileged EXEC command.

This example displays the output for the **show monitor** command for SPAN source session 1:

```
Switch# show monitor session 1
Session 1
-----
Type: Local Source Session
Source Ports:
    RX Only: Fa3/13
    TX Only:   None
    Both:      None

Source VLANs:
    RX Only:   None
    TX Only:   None
    Both:      None
Source RSPAN VLAN: None
Destination Ports: None
    Encapsulation: DOT1Q
    Ingress:Enabled, default VLAN=5
Filter VLANs:   None
Dest RSPAN VLAN: None
Ingress : Enabled, default VLAN=2
Learning : Disabled
```



Configuring ERSPAN

This module describes how to configure Encapsulated Remote Switched Port Analyzer (ERSPAN). The Cisco ERSPAN feature allows you to monitor traffic on ports or VLANs and send the monitored traffic to destination ports.



Note

The ERSPAN feature is not supported on Layer 2 switching interfaces.

This module describes the feature and consists of these sections:

- [Information About ERSPAN, page 70-2](#)
- [How to Configure ERSPAN, page 70-5](#)
- [Configuration Examples for ERSPAN, page 70-6](#)
- [Verifying ERSPAN, page 70-6](#)
- [Additional References for Configuring ERSPAN, page 70-8](#)
- [Feature Information for ERSPAN, page 70-9](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

Prerequisites for ERSPAN

- Only IPv4 delivery/transport header is supported.
- Access control list (ACL) filter is applied before sending the monitored traffic on to the tunnel.
- Only supports Type-II ERSPAN header.
- The extended VLAN ID that is reserved for each ERSPAN session cannot be used for any other purpose, until the ERSPAN source session is present in the configuration. Use the **show vlan internal usage** command to display the extended VLAN ID.
- This feature is supported on Cisco Catalyst 4500E Series Switches with Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, and 7-E, and 4500-X Series Switches.

Restrictions for ERSPAN

The following restrictions apply for this feature:

- The maximum number of available ports for each ERSPAN session is 128.
- A maximum of 8 ingress and 8 egress ERSPAN sessions (a total of 16 sessions) are supported.
- You can configure either a list of ports or a list of VLANs as a source, but cannot configure both for a given session.
- When a session is configured through the ERSPAN CLI, the session ID and the session type cannot be changed. To change them, you must use the **no** form of the configuration commands to remove the session and then reconfigure the session.
- ERSPAN source sessions do not copy locally-sourced Remote SPAN (RSPAN) VLAN traffic from source trunk ports that carry RSPAN VLANs.
- ERSPAN source sessions do not copy locally-sourced ERSPAN GRE-encapsulated traffic from source ports.

The Catalyst 4500 series switches do not support the following for this feature:

- ERSPAN destination type.
- Timestamp indication in the ERSPAN header.
- Bad/Short/Oversized (BSO) packet indication in the ERSPAN header.
- The original VLAN ID or Class of Service (COS) in the ERSPAN header.
- Generic routing encapsulation (GRE) header flags.
- GRE header sequence number or key.
- Maximum transmission unit (MTU) checking and fragmentation. Hence, traffic exceeding the configured MTU size (as determined by Layer 3 protocols) is dropped.
- Truncation of the original packet; because of which the T-bit in the ERSPAN header is always zero.
- Setting of the Differentiated Services Code Point (DSCP)/Time to Live (TTL) field for IP encapsulation differently for various ERSPAN source sessions. Use the **erspan {ttl | tos}** command to set these values. The configured values are used in all ERSPAN source sessions originating from the device.
- Simple Network Management Protocol (SNMP).

Information About ERSPAN

- [ERSPAN Overview, page 70-2](#)
- [ERSAN Sources, page 70-4](#)

ERSPAN Overview

The Cisco ERSPAN feature allows you to monitor traffic on ports or VLANs, and send the monitored traffic to destination ports. ERSPAN sends traffic to a network analyzer, such as a Switch Probe device or a Remote Monitoring (RMON) probe. ERSPAN supports source ports, source VLANs, and destination ports on different devices, which helps remote monitoring of multiple devices across a network.

ERSPAN supports encapsulated packets of up to 9180 bytes. ERSPAN consists of an ERSPAN source session, routable ERSPAN GRE-encapsulated traffic, and an ERSPAN destination session.

ERSPAN consists of an ERSPAN source session, routable ERSPAN GRE-encapsulated traffic, and an ERSPAN destination session. You can configure an ERSPAN source session, an ERSPAN destination session, or both on a device. A device on which only an ERSPAN source session is configured is called an ERSPAN source device, and a device on which only an ERSPAN destination session is configured is called an ERSPAN termination device. A device can act as both; an ERSPAN source device and a termination device.

**Note**

Catalyst 4500 series switches do not support ERSPAN destination type/session.

For a source port or a source VLAN, the ERSPAN can monitor the ingress, egress, or both ingress and egress traffic. By default, ERSPAN monitors all traffic, including multicast, and Bridge Protocol Data Unit (BPDU) frames.

An ERSPAN source session is defined by the following parameters:

- A session ID
- List of source ports or source VLANs to be monitored by the session
- The destination and origin IP addresses, which are used as the destination and source IP addresses of the GRE envelope for the captured traffic, respectively
- ERSPAN flow ID
- Optional attributes, such as, IP type of service (TOS) and IP Time to Live (TTL), related to the GRE envelope

**Note**

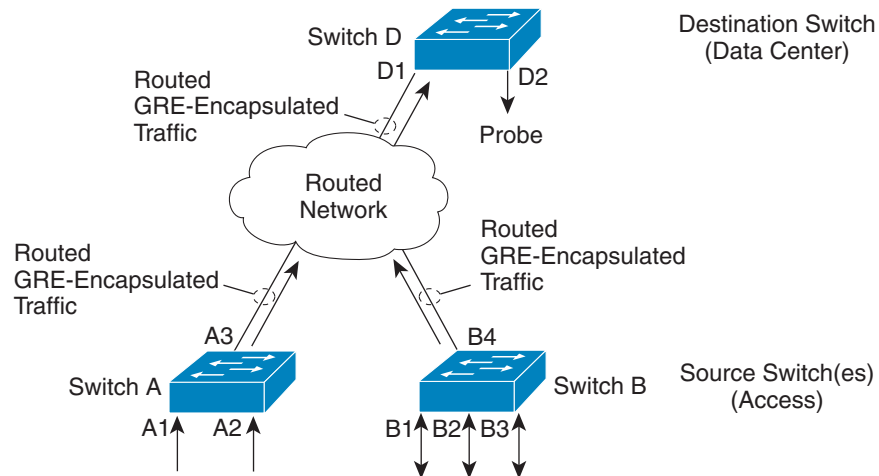
ERSPAN source sessions do not copy ERSPAN GRE-encapsulated traffic from source ports. Each ERSPAN source session can have either ports or VLANs as sources, but not both.

An ERSPAN destination session is defined by the following parameters:

- Session ID
- Destination ports
- Source IP address, which is the same as the destination IP address of the corresponding source session
- ERSPAN flow ID, which is used to match the destination session with the source session
- The ERSPAN source sessions copy traffic from the source ports or source VLANs and forwards the traffic using routable GRE-encapsulated packets to the ERSPAN destination session. The ERSPAN destination session switches the traffic to the destination ports

**Note**

Because encapsulation is performed in the hardware, the CPU performance is not impacted.

Figure 70-1 ERSPAN Configuration

120377

ERSAN Sources

The Cisco ERSPAN feature supports the following sources:

- Source ports—A source port that is monitored for traffic analysis. Source ports in any VLAN can be configured and trunk ports can be configured as source ports along with nontrunk source ports.
- Source VLANs—A VLAN that is monitored for traffic analysis.

The following tunnel interfaces are supported as source ports for a source session:


- GRE
- IPv6
- IPv6-over-IP tunnel
- Multipoint GRE (mGRE)


How to Configure ERSPAN

- [Configuring an ERSPAN Source Session, page 70-5](#)

Configuring an ERSPAN Source Session

The ERSPAN source session defines the session configuration parameters and the ports or VLANs to be monitored.

	Command or Action	Purpose
Step 1	Switch> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	Switch# configure terminal	Enters global configuration mode.
Step 3	Switch(config)# monitor session <i>span-session-number</i> type erspan-source	Defines an ERSPAN source session using the session ID and the session type, and enters ERSPAN monitor source session configuration mode. <ul style="list-style-type: none"> • Session IDs for source sessions or destination sessions are in the same global ID space, so each session ID is globally unique for both session types. • The <i>span-session-number</i> and the session type (configured by the erspan-source keyword) cannot be changed once configured. Use the no form of this command to remove the session and then re-create the session with a new session ID or a new session type.
Step 4	Switch(config-mon-erspan-src)# description <i>description</i>	Describes the ERSPAN source session.
Step 5	Switch(config-mon-erspan-src)# source { interface <i>type number</i> vlan <i>vlan-ID</i> } [, - <i>both</i> <i>rx</i> <i>tx</i>]	Configures the source interface or the VLAN, and the traffic direction to be monitored.
Step 6	Switch(config-mon-erspan-src)# filter { ip { <i>standard-access-list</i> <i>expanded-access-list</i> <i>acl-name</i> } ipv6 { access-group <i>acl-name</i> } vlan <i>vlan-ID</i> }	(Optional) Configures source VLAN filtering when the ERSPAN source is a trunk port. <div>  <p>Note You cannot include source VLANs and filter VLANs in the same session.</p> </div>
Step 7	Switch(config-mon-erspan-src)# no shutdown	Disables the shutting down of the configured session.
Step 8	Switch(config-mon-erspan-src)# destination	Defines an ERSPAN destination session and enters ERSPAN monitor destination session configuration mode.
Step 9	Switch(config-mon-erspan-src-dst)# ip address <i>ip-address</i>	Configures an IP address for the ERSPAN destination session.
Step 10	Switch(config-mon-erspan-src-dst)# erspan-id <i>erspan-ID</i>	Configures the ID used by the destination session to identify the ERSPAN traffic.
Step 11	Switch(config-mon-erspan-src-dst)# origin ip address <i>ip-address</i>	Configures the IP address used as the source for the ERSPAN traffic.

	Command or Action	Purpose
Step 12	Switch(config-mon-erspan-src-dst)# vrf vrf-ID	(Optional) Configures the VRF name to use instead of the global routing table.
Step 13	Switch(config-mon-erspan-src-dst)# exit	Exits ERSPAN monitor destination session configuration mode and returns to ERSPAN monitor source session configuration mode.
Step 14	Switch(config-mon-erspan-src)# exit	Exits ERSPAN monitor source session configuration mode and returns to privileged EXEC mode.
Step 15	Switch(config)# erspan {tos tos-value ttl ttl-value}	Configures type of service (ToS) and Time to Live (TTL) values for packets in the ERSPAN traffic.
		 Note The configured ToS and TTL values apply to all configured ERSPAN sessions. The ToS and TTL values for ERSPAN are configured in global configuration mode.
Step 16	Switch(config)# end	Exits global configuration mode and returns to privileged EXEC mode.

Configuration Examples for ERSPAN

- [Example: Configuring an ERSPAN Source Session, page 70-6](#)

Example: Configuring an ERSPAN Source Session

```
Switch> enable
Switch# configure terminal
Switch(config)# monitor session 1 type erspan-source
Switch(config-mon-erspan-src)# description source1
Switch(config-mon-erspan-src)# source interface fastethernet 0/1 rx
Switch(config-mon-erspan-src)# filter vlan 3
Switch(config-mon-erspan-src)# no shutdown
Switch(config-mon-erspan-src)# destination
Switch(config-mon-erspan-src-dst)# ip address 192.0.2.9
Switch(config-mon-erspan-src-dst)# erspan-id 2
Switch(config-mon-erspan-src-dst)# origin ip address 203.0.113.2
Switch(config-mon-erspan-src-dst)# vrf 1
Switch(config-mon-erspan-src)# exit
Switch(config)# erspan ttl 32
Switch(config)# end
Switch#
```

Verifying ERSPAN

To verify the ERSPAN configuration, use the following commands:

The following is sample output from the **show monitor session erspan-source** command:

```
Switch# show monitor session erspan-source session
```

```
Type : ERSPAN Source Session
```

```
Status : Admin Enabled
Source Ports :
RX Only : Gi1/4/33
Destination IP Address : 20.20.163.20
Destination ERSPAN ID : 110
Origin IP Address : 10.10.10.216
IPv6 Flow Label : None
```

The following is sample output from the **show monitor session erspan-source detail** command:

```
Switch# show monitor session erspan-source detail
```

```
Type : ERSPAN Source Session
Status : Admin Enabled
Description : -
Source Ports :
RX Only : Gi1/4/33
TX Only : None
Both : None
Source VLANs :
RX Only : None
TX Only : None
Both : None
Source RSPAN VLAN : None
Destination Ports : None
Filter VLANs : None
Filter Addr Type :
RX Only : None
TX Only : None
Both : None
Filter Pkt Type :
RX Only : None
Dest RSPAN VLAN : None
IP Access-group : None
IPv6 Access-group : None
Destination IP Address : 20.20.163.20
Destination IPv6 Address : None
Destination IP VRF : None
Destination ERSPAN ID : 110
Origin IP Address : 10.10.10.216
Origin IPv6 Address : None
IP QOS PREC : 0
IPv6 Flow Label : None
IP TTL : 255
```

The following output from the **show capability feature monitor erspan-source** command displays information about the configured ERSPAN source sessions:

```
Switch# show capability feature monitor erspan-source
```

```
ERSPAN Source Session Supported: true
No of Rx ERSPAN source session: 8
No of Tx ERSPAN source session: 8
ERSPAN Header Type supported: II
ACL filter Supported: true
Fragmentation Supported: false
Truncation Supported: false
Sequence number Supported: false
QOS Supported: true
```

The following output from the **show capability feature monitor erspan-destination** command displays all the configured global built-in templates:

```
Switch# show capability feature monitor erspan-destination
```

```
ERSPAN Destination Session Supported: false
```

Additional References for Configuring ERSPAN

Related Documents

Related Topic	Document Title
Cisco IOS commands	<i>Cisco IOS Master Commands List, All Releases</i>
Catalyst 4500 switch commands	<i>Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch</i>

Standards & MIBs

MIB	MIBs Link
•	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2784	Generic Routing Encapsulation (GRE)

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for ERSPAN

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

**Note**

Table 1 lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 1 Feature Information for ERSPAN

Feature Name	Releases	Feature Information
ERSPAN	Cisco IOS Release 15.2(4)E1	This module describes how to configure Encapsulated Remote Switched Port Analyzer (ERSPAN). The Cisco ERSPAN feature allows you to monitor traffic on ports or VLANs and send the monitored traffic to destination ports over a generic routing encapsulation (GRE) tunnel in any VRF. The following commands were introduced or modified: destination (ERSPAN) , erspan , filter (ERSPAN) , and show capability feature monitor .



Configuring Wireshark

Beginning with Cisco IOS Release XE 3.3.0SG in the IP Base and Enterprise Services feature sets, the Catalyst 4500 series switch supports Wireshark.



Note

Wireshark is only supported on Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, and 7-E, Catalyst 4500X-16, and Catalyst 4500X-32.



Note

Wireshark is supported on VSS and the functionality is the same as a standalone switch as detailed in the [“Configuring Wireshark on VSS” section on page 71-14](#).

- [Finding Feature Information, page 71-1](#)
- [Prerequisites for Wireshark, page 71-2](#)
- [Guidelines for Wireshark, page 71-2](#)
- [Restrictions for Wireshark, page 71-4](#)
- [Information about Wireshark, page 71-5](#)
- [How to Configure Wireshark, page 71-11](#)
- [Monitoring Wireshark, page 71-14](#)
- [Configuration Examples for Wireshark, page 71-14](#)
- [Usage Examples for Wireshark, page 71-18](#)
- [VSS Specific Examples, page 71-31](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to

<http://www.cisco.com/go/cfn>

An account on Cisco.com is not required.

Prerequisites for Wireshark

For general packet filtering, you will require Wireshark display filters. Refer to

<http://wiki.wireshark.org/DisplayFilters>

Guidelines for Wireshark

- During Wireshark packet capture, hardware forwarding happens concurrently.
- Before starting a Wireshark capture process, ensure that CPU usage is moderate and that sufficient memory (at least 200 MB) is available.
- If you plan to store packets to a storage file, ensure that sufficient space is available before beginning a Wireshark capture process.
- The CPU usage during Wireshark capture depends on how many packets match the specified conditions and on the intended actions for the matched packets (store, decode and display, or both).
- Where possible, keep the capture to the minimum (limit by packets, duration) to avoid high CPU usage and other undesirable conditions.
- Because packet forwarding typically occurs in hardware, packets are not copied to the CPU for software processing. For Wireshark packet capture, packets are copied and delivered to the CPU, which causes an increase in CPU usage.

To avoid high CPU, do the following:

- Attach only relevant ports.
- Use a class map, and secondarily, an access list to express match conditions. If neither is viable, use an explicit, in-line filter.
- Adhere closely to the filter rules. Restrict the traffic type (such as, IPv4 only) with a restrictive, rather than relaxed ACL, which elicits unwanted traffic.
- Always limit packet capture to either a shorter duration or a smaller packet number. The parameters of the **capture** command enable you to specify the following:
 - Capture duration
 - Number of packets captured
 - File size
 - Packet segment size
- Run a capture session without limits if you know that very little traffic matches the core filter.
- You might experience high CPU (or memory usage) if:
 - You leave a capture session enabled and unattended for a long period of time, resulting in unanticipated bursts of traffic.

- You launch a capture session with ring files or capture buffer and leave it unattended for a long time, resulting in performance or system health issues.
- During a capture session, watch for high CPU usage and memory consumption due to Wireshark that may impact switch performance or health. If these situations arise, stop the Wireshark session immediately.
- Avoid decoding and displaying packets from a .pcap file for a large file. Instead, transfer the .pcap file to a PC and run Wireshark on the PC.
- Limit the number of Wireshark instances to two or less to avoid CPU or memory resource drain.
You can use up to eight Wireshark instances. An active **show** command that decodes and displays packets from a .pcap file or capture buffer counts as one instance.
- Whenever an ACL is installed or modified on a switch in the ingress direction, for the first 15 seconds, the software ignores packet classification details sent by the hardware. Instead, it uses software-based classification for the packets received by CPU. So, during this period, the system can only capture fewer packets (compared to the time after the first 15 seconds) and CPU usage will be high.
- To avoid packet loss, consider the following:
 - Use store-only (when you do not specify the display option) while capturing live packets rather than decode and display, which is an CPU-intensive operation (especially in detailed mode).
 - If you use the default buffer size, packets may be dropped. Increase buffer size and avoid packet loss.
 - Writing to flash disk is a CPU-intensive operation, so the capture rate may not be sufficient.
 - The Wireshark capture session operates normally in streaming mode where packets are both captured and processed. However, when you specify a buffer size of at least 32 MB but less than 80MB, the session automatically turns on lock-step mode in which a Wireshark capture session is split into two phases: capture and process. In the capture phase, the packets are stored in the temporary buffer. The duration parameter in lock-step mode serves as capture duration rather than session duration. When the buffer is full or the capture duration has ended, a session transitions to the process phase, wherein it stops accepting packets and starts processing packets in the buffer. With the second approach (lock-step mode), a higher capture throughput can be achieved. Last, when you specify a buffer size of at least 80MB, the session turns on lock-step mode with high-speed capture. This is similar to lock-step mode except that it captures the packets directly from the hardware queue and passes the packet to the wireshark packet queue.
 - The streaming capture mode supports approximately 1500 pps; lock-step mode supports approximately 45 Mbps (measured with 256-byte packets); lock-step mode with high speed capture supports roughly 750Mbps (measure with 256-byte packets). When the matching traffic rate exceeds this number, you may experience packet loss. Only one session can be started when using high-speed capture mode.
- If you want to decode and display live packets in the console window, ensure that the Wireshark session is bounded by a short capture duration.

**Warning**

A Wireshark session with either a longer duration limit or no capture duration (using a terminal with no auto-more support using the term len 0 command) may make the console or terminal unusable.

- When using Wireshark to capture live traffic that leads to high CPU, usage, consider applying a QoS policy temporarily to limit the actual traffic until the capture process concludes.
- All Wireshark-related commands are in EXEC mode; no configuration commands exist for Wireshark.

If you need to use access list or class-map in the Wireshark CLI, you must define an access list and class map with configuration commands.

- No specific order applies when defining a capture point; you can define capture point parameters in any order, provided that CLI allows this. The Wireshark CLI allows as many parameters as possible on a single line. This limits the number of commands required to define a capture point.
- All parameters except attachment points take a single value. Generally, you can replace the value with a new one by reentering the command. After user confirmation, the system accepts the new value and overrides the older one. A **no** form of the command is unnecessary to provide a new value; it is necessary to remove a parameter.
- Wireshark allows you to specify one or more attachment points. To add more than one attachment point, reenter the command with the new attachment point. To remove an attachment point, use the **no** form. You can specify an interface range as an attachment point.

For example, **monitor capture mycap int gi 3/1 in**, where interface gi 3/1 is an attachment point.

If you also need to attach gi 3/2, you specify it in another line, as follows:

monitor capture mycap int gi 3/2 in

- You cannot modify any parameters of a capture point while a session is active. To modify any parameter, stop the session, make the changes, and restart the session. Because an access list is generic to a switch and unrelated to the Wireshark process, it is alterable during a Wireshark session.
- The action you want to perform determines which parameters are mandatory. The Wireshark CLI allows you to specify or modify any parameter prior to entering the **start** command. When you issue the **start** command, Wireshark will start only after determining that all mandatory parameters have been provided.
- If the capture file already exists, it provides a warning and receives confirmation before proceeding. This prevents you from mistakenly overwriting a file.
- The core filter can be an explicit filter, access list, or class map. Specifying a newer filter of these types replaces the existing one.
- You can terminate a Wireshark session with an explicit **stop** command or by entering **q** in automore mode. The session could terminate itself automatically when a stop condition such as duration or packet capture limit is met.

Restrictions for Wireshark

- The CLI for configuring Wireshark requires that the feature be executed only from EXEC mode. Actions that usually occur in configuration submode (such as defining capture points), are handled at the EXEC mode instead. All key commands are not NVGEN'd and are not synchronized to the standby supervisor in NSF and SSO scenarios.
- When packet capture is enabled in the input direction, the matching packets undergo software-based lookup in the CPU for the first 15 seconds. During this time, CPU usage is high and capture rate is low.
- Packets captured in the output direction of an interface might not reflect the changes made by switch rewrite (includes TTL, VLAN tag, CoS, checksum, and MAC addresses).
- Capturing at a physical port that belongs to another logical port may not be supported. For example, capturing at EtherChannel member ports is not supported.
- Limiting circular file storage by file size is not supported.

- Wireshark cannot capture IPv6 packets if the capture point's class-map filter is attempting to match one of the following:
 - Extension headers followed by Hop-by-hop header (as per CSCtt16385)
 - DSCP values (as per CSCtx75765)

Information about Wireshark

**Note**

Wireshark is only supported on Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, and 7-E, Catalyst 4500X-16, and Catalyst 4500X-32.

**Note**

Wireshark is supported on VSS and the functionality is the same as a standalone switch except for a few configuration differences as detailed in the [“Configuring Wireshark on VSS”](#) section on page 71-14.

Wireshark is a packet analyzer program, formerly known as Ethereal, which supports multiple protocols and presents information in a text-based user interface.

To understand what happens inside a network requires the ability to capture and analyze traffic. Prior to Cisco IOS Release XE 3.3.0SG, the Catalyst 4500 series switch offered only two features to address this need: SPAN and **debug platform packet**. Both are limited. SPAN is ideal for capturing packets, but can only deliver them by forwarding them to some specified local or remote destination; it provides no local display or analysis support. The **debug platform packet** command is specific to the Catalyst 4500 series switch and only works on packets that stem from the software process-forwarding path. Although it has limited local display capabilities, it has no analysis support.

So the need exists for a traffic capture and analysis mechanism that is applicable to both hardware and software forwarded traffic and that provides strong packet capture, display and analysis support, preferably using a well known interface.

Wireshark dumps packets to a file using a well known format called .pcap, and is applied or enabled on individual interfaces. You specify an interface in EXEC mode along with the filter and other parameters. The Wireshark application is applied only when you enter a **start** command and is removed only when Wireshark stops capturing packets either automatically or manually.

**Note**

In Cisco IOS Release XE 3.3.0SG, global packet capture on Wireshark is not supported.

These sections describe some key concepts for Wireshark:

- [Capture Points, page 71-6](#)
- [Attachment Points, page 71-6](#)
- [Filters, page 71-6](#)
- [Input and Output Classification, page 71-7](#)
- [Actions, page 71-8](#)
- [Storing Captured Packets to Buffer in Memory, page 71-8](#)
- [Decoding and Displaying Packets, page 71-9](#)
- [Activating and Deactivating Wireshark Capture Points, page 71-10](#)

- [Wireshark Features used in Switches, page 71-10](#)
- [Wireshark on VSS, page 71-11](#)

Capture Points

A capture point is the central policy definition of the Wireshark feature. The point describes all the characteristics associated with a given instance of Wireshark: what packets to capture, where to capture them from, what to do with the captured packets, and when to stop. Capture points can be modified after creation and do not become active until explicitly activated with a **start** command. This process is termed *activating the capture point* or *starting the capture point*. Capture points are identified by name and may also be manually or automatically deactivated or stopped.

Multiple capture points may be defined and activated simultaneously.

Attachment Points

An attachment point is a point in the logical packet process path associated with a capture point. Consider an attachment point as an attribute of the capture point. Packets that impact an attachment point are tested against the capture point's filters; packets that match are copied and sent to the capture point's associated Wireshark instance. A specific capture point can be associated with multiple attachment points, with limits on mixing attachment points of different types. Some restrictions apply when you specify attachment points of different types. Attachment points are directional (input or output or both) with the exception of the Layer 2 VLAN attachment point, which is always unidirectional.

Filters

Filters are attributes of a capture point that identify and limit the subset of traffic traveling through the attachment point of a capture point, which is copied and passed to Wireshark. To be displayed by Wireshark, a packet must pass through an attachment point, as well as all of the filters associated with the capture point.

A capture point has three types of filters:

- Core system filter—The core system filter is applied by hardware, and its match criteria is limited by hardware. This filter determines whether hardware-forwarded traffic is copied to software for Wireshark purposes.
- Capture filter—The capture filter is applied by Wireshark. The match criteria are more granular than those supported by the core system filter. Packets that pass the core filter but fail the capture filter are still copied and sent to the CPU/software, but are discarded by the Wireshark process. The capture filter syntax matches that of the display filter.



Note Wireshark on the Catalyst 4500 series switch does not use the syntax of the capture filter.

- Display filter—The display filter is applied by Wireshark, and its match criteria are similar to those of the capture filter. Packets that fail the display filter are not displayed.

Core System Filter

You can specify core system filter match criteria by using the class map or ACL, or explicitly by using the CLI.

In some installations, you need to obtain authorization to modify the switch configuration, which can lead to extended delays if the approval process is lengthy. This would limit the ability of network administrators to monitor and analyze traffic. To address this situation, Wireshark supports explicit specification of core system filter match criteria from the EXEC mode CLI. The disadvantage is that the match criteria that you can specify is a limited subset of what class map supports, such as MAC, IP source and destination addresses, ether-type, IP protocol, and TCP/UDP source and destination ports.

If you prefer to use configuration mode, you can define ACLs or have class maps refer capture points to them. Explicit and ACL-based match criteria are used internally to construct class maps and policy maps. These implicitly constructed class maps are not reflected in the switch running-config and are not NVGEN'd.

**Note**

The ACL and class map configuration are part of the system and not aspects of the Wireshark feature.

Capture Filter

The capture filter allows you to direct Wireshark to further filter incoming packets based on various conditions. Wireshark applies the capture filter immediately on receipt of the packet; packets that fail the capture filter are neither stored nor displayed.

A switch receives this parameter and passes it unchanged to Wireshark. Because Wireshark parses the application filter definition, the defining syntax is the one provided by the Wireshark display filter. This syntax and that of standard Cisco IOS differ, which allows you to specify ACL match criteria that cannot be expressed with standard syntax.

**Note**

The capture filter syntax matches that of the Wireshark display filter. The syntax for capture and display filters are identical in the Wireshark implementation on the Catalyst 4500 series switch.

Display Filter

With the display filter, you can direct Wireshark to further narrow the set of packets to display when decoding and displaying from a .pcap file. Because the syntax of the display filter is identical to the capture filter, the display filter is superfluous if a capture filter is also defined.

For more details on the syntax of capture and display filters, go to

<http://wiki.wireshark.org/DisplayFilters>

Input and Output Classification

There are four classification results for input and output classification. In the input direction, they are ordered role-based, security, QoS, and forwarding override. In the output direction they are ordered forwarding override, role-based, security, and QoS.

On the input side, the Wireshark capture feature is placed in the forwarding override result type, prioritized above the other FO features (such as multicast local source capture, PBR and ingress WCCP). The packets captured by Wireshark are before any redirection by PBR or WCCP. Because security ACLs are applied ahead of FO-related features, packets that are dropped by security ACLs are not captured by Wireshark.

On the output side, the Wireshark capture feature is placed in the forwarding override result type, prioritized below the other FO features (such as egress WCCP). Wireshark captures packets only if the other egress FO features do not apply.

Actions

Wireshark can be invoked on live traffic or on a previously existing .pcap file. When invoked on live traffic, it can perform four types of actions on packets that pass its capture and display filters:

- Captures to buffer in memory to decode and analyze and store
- Stores to a .pcap file
- Decodes and displays
- Stores and displays

When invoked on a .pcap file only, only the decode and display action is applicable.

Storing Captured Packets to Buffer in Memory

Packets can be stored in the capture buffer in memory for subsequent decode, analysis, or storage to a .pcap file.

The capture buffer can be linear or circular mode. In linear mode, new packets are discarded when the buffer is full. In circular mode, if the buffer is full, the oldest packet are discarded to accommodate the new packet. Although the buffer can also be cleared when needed, this mode is mainly used for debugging network traffic.

Storing Captured Packets to a .pcap File

Wireshark can store captured packets to a .pcap file. The capture file can be located on the following storage devices:

- Catalyst 4500 series switch on-board flash storage (bootflash:)
- external flash disk (slot0:)
- USB drive (usb0:)



Note Do not attempt to use Wireshark with any other devices.

When configuring a Wireshark capture point, you can associate a filename. When the capture point is activated, Wireshark creates a file with the specified name and writes packets to it. If the file already exists when the file is associated or the capture point is activated, Wireshark queries you as to whether the file can be overwritten. Only one capture point may be associated with a given filename.

If the destination of the Wireshark writing process is full, Wireshark fails with partial data in the file. You must ensure that there is sufficient space in the file system before you start the capture session. With Cisco IOS Release IOS XE 3.3.0SG, the file system full status is not detected for some storage devices.

You can reduce the required storage space by retaining only a segment, instead of the entire packet. Typically, you do not require details beyond the first 64 or 128 bytes. The default behavior is to store the entire packet.

To avoid possible packet drops when processing and writing to the file system, Wireshark can optionally use a memory buffer to temporarily hold packets as they arrive. Memory buffer size can be specified when the capture point is associated with a .pcap file.

Decoding and Displaying Packets

Wireshark can decode and display packets to the console. This functionality is possible for capture points applied to live traffic and for capture points applied to a previously existing .pcap file.



Note

Decoding and displaying packets may be CPU intensive.

Wireshark can decode and display packet details for a wide variety of packet formats. The details are displayed by entering the **monitor capture name start** command with one of the following keyword options, which place you into a display and decode mode:

- **brief**—Displays one line per packet (the default).
- **detailed**—Decodes and displays all the fields of all the packets whose protocols are supported. Detailed mode require more CPU than the other two modes.
- **(hexadecimal) dump**—Displays one line per packet as a hexadecimal dump of the packet data and the printable characters of each packet.

When we enter the **capture** command with the decode and display option, the Wireshark output is returned to Cisco IOS and displayed on the console unchanged.

Displaying Live Traffic

Wireshark receives copies of packets from the Catalyst 4500 series switch core system. Wireshark applies its capture and display filters to discard uninteresting packets, and then decodes and displays the remaining packets.

Displaying from the .pcap File

Wireshark can decode and display packets from a previously stored .pcap file and direct the display filter to selectively displayed packets. A capture filter is not applicable in this situation.

Storing and Displaying Packets

Functionally, this mode is a combination of the previous two modes. Wireshark stores packets in the specified .pcap file and decodes and displays them to the console. Only the core and capture filters are applicable here.

Activating and Deactivating Wireshark Capture Points

After a Wireshark capture point has been defined with its attachment points, filters, actions, and other options, it must be activated. Until the capture point is activated, it does not actually capture packets.

Before a capture point is activated, some sanity checks are performed. A capture point cannot be activated if it has neither a core system filter nor attachment points defined. Attempting to activate a capture point that generates an error.

The capture and display filters are specified as needed.

After Wireshark capture points are activated, they can be deactivated in multiple ways. A capture point that is storing only packets to a .pcap file can be halted manually or configured with time or packet limits, after which the capture point halts automatically. Only packets that pass the Wireshark capture filter are counted against the packet limit threshold.

When a Wireshark capture point is activated, a fixed rate filter is applied automatically in the hardware so that the CPU is not flooded with Wireshark-directed packets. The disadvantage of the rate filter is that you cannot capture contiguous packets beyond the established rate even if more resources are available.

Wireshark Features used in Switches

This section describes how Wireshark features function in the Catalyst 4500 series switch environment:

- Layer 2 security features—Packets that are dropped by Layer 2 security features (such as port security, MAC address filtering, and spanning tree) are not captured by Wireshark. This differs from the behavior of SPAN.
- Classification-based security features—Packets that are dropped by input classification-based security features (such as ACLs and IPSG) are not caught by Wireshark capture points that are connected to attachment points at the same layer. In contrast, packets that are dropped by output classification-based security features are caught by Wireshark capture points that are connected to attachment points at the same layer. The logical model is that the Wireshark attachment point occurs after the security feature lookup on the input side, and symmetrically before the security feature lookup on the output side.

Wireshark capture policies connected to Layer 2 attachment points in the input direction capture packets dropped by Layer 3 classification-based security features. Symmetrically, Wireshark capture policies attached to Layer 3 attachment points in the output direction capture packets dropped by Layer 2 classification-based security features.

- Routed ports and Layer 3 port channels—When a routed port or Layer 3 port channel is used as a Wireshark attachment point, the policy that is applied to capture the packets is treated as attached at Layer 3. Wireshark only captures packets that are being routed by the interface.
- VLANs—When a VLAN is used as a Wireshark attachment point, packets are captured in both input and output directions. A packet that is bridged in the VLAN generates two copies, one on input and one on output.
- Private VLANs—Secondary PVLANS are disallowed as Wireshark attachment points. Using a primary PVLAN as a Wireshark attachment point enables capture of packets in the primary PVLAN and all associated secondary PVLANS. The entire PV domain becomes the attachment point.
- Redirection features—In the input direction, features traffic redirected by Layer 3 (such as PBR and WCCP), are logically later than Layer 3 Wireshark attachment points. Wireshark captures these packets even though they might later be redirected out another Layer 3 interface. Symmetrically, output features redirected by Layer 3 (such as egress WCCP) are logically prior to Layer 3 Wireshark attachment points, and Wireshark will not capture them.

- Classification copy features—Features that generate copies of packets from the role-based and Security lookup types are compatible with Wireshark. Multiple copies of these packets are generated.
- SPAN—Wireshark cannot capture packets on interface configured as a SPAN source or destination.

Wireshark on VSS

Wireshark is supported on the VSS, and usage is similar to the Standalone switch. You need to create and delete capture points for the VSS active and VSS standby switches on the VSS active switch. You need to start and stop capture operations on VSS active switch, once you have issued the attachment points for a capture session on the active switch.

How to Configure Wireshark

To configure Wireshark, follow these general steps:

- Step 1** Define, modify or delete a capture point.
- Step 2** Activate or deactivate a capture point.

Default Wireshark Configuration

Table 71-1 shows the default Wireshark configuration.

Table 71-1 Default Wireshark Configuration

Feature	Default Setting
Duration	No limit
Packets	No limit
Packet-length	No limit (full packet)
File size	No limit
Ring file storage	No
Buffer storage mode	Linear

Defining, Modifying, or Deleting a Capture Point

Although listed in sequence, the steps to specify values for the options can be executed in any order. You can also specify them in one, two, or several lines. Except for attachment points, which can be multiple, you can replace any value with a more recent value by redefining the same option, in the following order:

- Step 1** Define the name that identifies the capture point.
- Step 2** Specify the attachment point with which the capture point is associated.

Multiple attachment points can be specified. Range support is also available both for adding and removing attachment points.

- Step 3** Define the core system filter, defined either explicitly, through ACL or through a class map.
- Step 4** Specify the session limit (in seconds or packets captured).
- Step 5** Specify the packet segment length to be retained by Wireshark.
- Step 6** Specify the file association, if the capture point intends to capture packets rather than merely display them.
- Step 7** Specify the size of the memory buffer used by Wireshark to handle traffic bursts.

To filter the capture point, use the following commands:

Command	Purpose
[no] monitor capture mycap match { any mac <i>mac-match-string</i> ipv4 <i>ipv4-match-string</i> ipv6 <i>ipv6-match-string</i> }	Defines an explicitly in-line core filter. To remove the filter, use the no form of this command.
[no] monitor capture mycap match mac { <i>src-mac-addr</i> <i>src-mac-mask</i> any host <i>src-mac-addr</i> } { <i>dest-mac-addr</i> <i>dest-mac-mask</i> any host <i>dest-mac-addr</i> }	Specifies use of a filter for MAC. To remove the filter, use the no form of this command.
[no] monitor capture mycap match {ipv4 ipv6} [<i>src-prefix/length</i> any host <i>src-ip-addr</i>] [<i>dest-prefix/length</i> any host <i>dest-ip-addr</i>]	Specifies a filter for IPv4/IPv6, use one of the formats. To remove the filters, use the no form of this command.
[no] monitor capture mycap match {ipv4 ipv6} proto { tcp udp } [<i>src-prefix/length</i> any host <i>src-ip-addr</i>] [eq gt lt neq <0-65535>] [<i>dest-prefix/length</i> any host <i>dest-ip-addr</i>] [eq gt lt neq <0-65535>]	

To define a capture point, use the following commands:

Command	Purpose
monitor capture name [{ interface <i>name</i> vlan <i>num</i> control-plane } { in out both }	Specifies one or more attachment points with direction. To remove the attachment point, use the no form of this command.
monitor capture name [[file location <i>filename</i> [buffer-size <1-100>] [ring <2-10>] [size <1-100>]] [buffer [circular] size <1-100>]]	Specifies the capture destination. To remove the details, use the no form of this command.
[no] monitor capture name limit { duration <i>seconds</i> } [packet-length <i>size</i>] [packets <i>num</i>]	Specifies capture limits. To remove the limits, use the no form of this command.

To clear the buffer contents, use the following command

Command	Purpose
monitor capture [clear export <i>filename</i>]	Clears capture buffer contents or stores the packets to a file.

To start and stop a capture point, use the following command:

Command	Purpose
monitor capture <i>name</i> start [capture-filter <i>filter-string</i>] [display [display-filter <i>filter-string</i>]] [brief detailed dump stop]	To start or stop a capture point, use the monitor capture command.

Examples

Associating or Disassociating a Capture File

```
Switch# monitor capture point mycap file location bootdisk:mycap.pcap

Switch# no monitor capture mycap file
```

Specifying a Memory Buffer Size for Packet Burst Handling

```
Switch# monitor capture mycap buffer-size 1000000
```

Defining an Explicit Core System Filter to Match Both IPv4 and IPv6 TCP Traffic

```
Switch# monitor capture mycap match any protocol tcp
```

Defining a Core System Filter Using an Existing ACL or Class Map

```
Switch# monitor capture mycap match access-list myacl

Switch# monitor capture mycap match class-map mycm
```

Activating and Deactivating a Capture Point

A capture point cannot be activated unless an attachment point and a core system filter have been defined and the associated filename (if any) does not already exist. A capture point with no associated filename can only be activated to display. If no capture or display filters are specified, all of the packets captured by the core system filter are displayed. The default display mode is **brief**.

To activate or deactivate a capture point, perform these tasks:

Command	Purpose
monitor capture <i>name</i> start [capture-filter <i>filter-string</i>] [display [display-filter <i>filter-string</i>]] [brief detailed dump]	Activates a capture point.
monitor capture <i>name</i> stop	Deactivates a capture point.
Example: Switch# monitor capture mycap start capture-filter "net 10.1.1.0 0.0.0.255 and port 80" Switch# monitor capture mycap start display display-filter "net 10.1.1.0 0.0.0.255 and port 80"	

Configuring Wireshark on VSS

To configure Wireshark in the VSS active switch, perform the following step:

```
Switch_VSS# monitor capture mycap match any interface gi2/1/1 in file location
bootflash:text.pcap
```

Monitoring Wireshark

The commands in the following table are used to monitor Wireshark.

Table 71-2 Wireshark Monitoring Commands

Command	Purpose
<code>show monitor capture point name</code>	Displays the capture point state so that you can see what capture points are defined, what their attributes are, and whether they are active. When capture point <i>name</i> is specified, it displays specific capture point's details.
<code>show monitor capture file name</code> <code>[display-filter filter-string] [brief detailed dump]</code>	Activates Wireshark using an existing .pcap file as the source for packets. If no display filter is specified, all of the packets in the file are displayed. The default display mode is brief .

Configuration Examples for Wireshark

Example: Displaying a Brief Output from a .pcap File

You can display the output from a .pcap file by entering:

```
Switch# show monitor capture file bootflash:mycap.pcap
 1  0.000000  10.1.1.140 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
 2  1.000000  10.1.1.141 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
 3  2.000000  10.1.1.142 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
 4  3.000000  10.1.1.143 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
 5  4.000000  10.1.1.144 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
 6  5.000000  10.1.1.145 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
 7  6.000000  10.1.1.146 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
 8  7.000000  10.1.1.147 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
 9  8.000000  10.1.1.148 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
10  9.000000  10.1.1.149 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
11 10.000000  10.1.1.150 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
12 11.000000  10.1.1.151 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
13 12.000000  10.1.1.152 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
14 13.000000  10.1.1.153 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
15 14.000000  10.1.1.154 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
16 15.000000  10.1.1.155 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
17 16.000000  10.1.1.156 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
18 17.000000  10.1.1.157 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
19 18.000000  10.1.1.158 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
20 19.000000  10.1.1.159 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
21 20.000000  10.1.1.160 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
22 21.000000  10.1.1.161 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
23 22.000000  10.1.1.162 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
24 23.000000  10.1.1.163 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
25 24.000000  10.1.1.164 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
26 25.000000  10.1.1.165 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
27 26.000000  10.1.1.166 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
28 27.000000  10.1.1.167 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
29 28.000000  10.1.1.168 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
```

```

30 29.000000 10.1.1.169 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
31 30.000000 10.1.1.170 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
32 31.000000 10.1.1.171 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
33 32.000000 10.1.1.172 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
34 33.000000 10.1.1.173 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
35 34.000000 10.1.1.174 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
36 35.000000 10.1.1.175 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
37 36.000000 10.1.1.176 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
38 37.000000 10.1.1.177 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
39 38.000000 10.1.1.178 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
40 39.000000 10.1.1.179 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
41 40.000000 10.1.1.180 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
42 41.000000 10.1.1.181 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
43 42.000000 10.1.1.182 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
44 43.000000 10.1.1.183 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
45 44.000000 10.1.1.184 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
46 45.000000 10.1.1.185 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
47 46.000000 10.1.1.186 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
48 47.000000 10.1.1.187 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
49 48.000000 10.1.1.188 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
50 49.000000 10.1.1.189 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
51 50.000000 10.1.1.190 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
52 51.000000 10.1.1.191 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
53 52.000000 10.1.1.192 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
54 53.000000 10.1.1.193 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
55 54.000000 10.1.1.194 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
56 55.000000 10.1.1.195 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
57 56.000000 10.1.1.196 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
58 57.000000 10.1.1.197 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
59 58.000000 10.1.1.198 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002

```

Example: Displaying Detailed Output from a .pcap File

You can display the detailed .pcap file output by entering:

```

Switch# show monitor capture file bootflash:mycap.pcap detailed
Frame 1: 256 bytes on wire (2048 bits), 256 bytes captured (2048 bits)
  Arrival Time: Mar 21, 2012 14:35:09.111993000 PDT
  Epoch Time: 1332365709.111993000 seconds
  [Time delta from previous captured frame: 0.000000000 seconds]
  [Time delta from previous displayed frame: 0.000000000 seconds]
  [Time since reference or first frame: 0.000000000 seconds]
  Frame Number: 1
  Frame Length: 256 bytes (2048 bits)
  Capture Length: 256 bytes (2048 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  [Protocols in frame: eth:ip:udp:data]
Ethernet II, Src: 00:00:00:00:03:01 (00:00:00:00:03:01), Dst: 54:75:d0:3a:85:3f
(54:75:d0:3a:85:3f)
  Destination: 54:75:d0:3a:85:3f (54:75:d0:3a:85:3f)
  Address: 54:75:d0:3a:85:3f (54:75:d0:3a:85:3f)
  ....0. .... = IG bit: Individual address (unicast)
  ....0. .... = LG bit: Globally unique address (factory default)
  Source: 00:00:00:00:03:01 (00:00:00:00:03:01)
  Address: 00:00:00:00:03:01 (00:00:00:00:03:01)
  ....0. .... = IG bit: Individual address (unicast)
  ....0. .... = LG bit: Globally unique address (factory default)
  Type: IP (0x0800)
  Frame check sequence: 0x03b07f42 [incorrect, should be 0x08fcee78]
Internet Protocol, Src: 10.1.1.140 (10.1.1.140), Dst: 20.1.1.2 (20.1.1.2)
  Version: 4
  Header length: 20 bytes
  Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    0000 00.. = Differentiated Services Codepoint: Default (0x00)
    ....0. = ECN-Capable Transport (ECT): 0
    ....0. = ECN-CE: 0
  Total Length: 238

```

```

Identification: 0x0000 (0)
Flags: 0x00
    0... .... = Reserved bit: Not set
    .0... .... = Don't fragment: Not set
    ..0. .... = More fragments: Not set
Fragment offset: 0
Time to live: 64
Protocol: UDP (17)
Header checksum: 0x5970 [correct]
    [Good: True]
    [Bad: False]
Source: 10.1.1.140 (10.1.1.140)
Destination: 20.1.1.2 (20.1.1.2)
User Datagram Protocol, Src Port: 20001 (20001), Dst Port: 20002 (20002)
Source port: 20001 (20001)
Destination port: 20002 (20002)
Length: 218
Checksum: 0x6e2b [validation disabled]
    [Good Checksum: False]
    [Bad Checksum: False]
Data (210 bytes)

0000  00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f  .....
0010  10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f  .....
0020  20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f  !"#%&'()*+,-./
0030  30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f  0123456789:;<=>?
0040  40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f  @ABCDEFGHJKLMNO
0050  50 51 52 53 54 55 56 57 58 59 5a 5b 5c 5d 5e 5f  PQRSTUVWXYZ[\]^_
0060  60 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f  `abcdefghijklmnopqrstuvwxyz
0070  70 71 72 73 74 75 76 77 78 79 7a 7b 7c 7d 7e 7f  pqrstuvwxyz{|}~.
0080  80 81 82 83 84 85 86 87 88 89 8a 8b 8c 8d 8e 8f  .....
0090  90 91 92 93 94 95 96 97 98 99 9a 9b 9c 9d 9e 9f  .....
00a0  a0 a1 a2 a3 a4 a5 a6 a7 a8 a9 aa ab ac ad ae af  .....
00b0  b0 b1 b2 b3 b4 b5 b6 b7 b8 b9 ba bb bc bd be bf  .....
00c0  c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 ca cb cc cd ce cf  .....
00d0  d0 d1  .....
      Data: 000102030405060708090a0b0c0d0e0f1011121314151617...
      [Length: 210]

Frame 2: 256 bytes on wire (2048 bits), 256 bytes captured (2048 bits)
Arrival Time: Mar 21, 2012 14:35:10.111993000 PDT

```

Example: Displaying a Hexadecimal Dump Output from a .pcap File

You can display the hexadecimal dump output by entering:

```

Switch# show monitor capture file bootflash:mycap.pcap dump
  1  0.000000  10.1.1.140 -> 20.1.1.2      UDP Source port: 20001  Destination port:
20002

0000  54 75 d0 3a 85 3f 00 00 00 00 03 01 08 00 45 00  Tu...?.....E.
0010  00 ee 00 00 00 00 40 11 59 70 0a 01 01 8c 14 01  .....@.Yp.....
0020  01 02 4e 21 4e 22 00 da 6e 2b 00 01 02 03 04 05  ..N!N"..n+.....
0030  06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15  .....
0040  16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25  ..... !"#%
0050  26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35  &'()*+,-./012345
0060  36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45  6789:;<=>?@ABCDE
0070  46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55  FGHIJKLMNOPQRSTU
0080  56 57 58 59 5a 5b 5c 5d 5e 5f 60 61 62 63 64 65  VWXYZ[\]^_`abcde
0090  66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75  fghijklmnopqrstu
00a0  76 77 78 79 7a 7b 7c 7d 7e 7f 80 81 82 83 84 85  vwxyz{|}~.....
00b0  86 87 88 89 8a 8b 8c 8d 8e 8f 90 91 92 93 94 95  .....

```

```

00c0 96 97 98 99 9a 9b 9c 9d 9e 9f a0 a1 a2 a3 a4 a5 .....
00d0 a6 a7 a8 a9 aa ab ac ad ae af b0 b1 b2 b3 b4 b5 .....
00e0 b6 b7 b8 b9 ba bb bc bd be bf c0 c1 c2 c3 c4 c5 .....
00f0 c6 c7 c8 c9 ca cb cc cd ce cf d0 d1 03 b0 7f 42 .....B

      2      1.000000      10.1.1.141 -> 20.1.1.2      UDP Source port: 20001 Destination port:
20002

0000 54 75 d0 3a 85 3f 00 00 00 00 03 01 08 00 45 00 Tu...?.....E.
0010 00 ee 00 00 00 00 40 11 59 6f 0a 01 01 8d 14 01 .....@.Yo.....
0020 01 02 4e 21 4e 22 00 da 6e 2a 00 01 02 03 04 05 ..N!N".n*.....
0030 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 .....
0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 ..... !"#$$%
0050 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 &'()*+,-./012345
0060 36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45 6789:;<=>?@ABCDE
0070 46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 FGHIJKLMNOPQRSTU
0080 56 57 58 59 5a 5b 5c 5d 5e 5f 60 61 62 63 64 65 VWXYZ[\]^_`abcde
0090 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 fghijklmnopqrstu
00a0 76 77 78 79 7a 7b 7c 7d 7e 7f 80 81 82 83 84 85 vwxyz{|}~.....
00b0 86 87 88 89 8a 8b 8c 8d 8e 8f 90 91 92 93 94 95 .....
00c0 96 97 98 99 9a 9b 9c 9d 9e 9f a0 a1 a2 a3 a4 a5 .....
00d0 a6 a7 a8 a9 aa ab ac ad ae af b0 b1 b2 b3 b4 b5 .....
00e0 b6 b7 b8 b9 ba bb bc bd be bf c0 c1 c2 c3 c4 c5 .....
00f0 c6 c7 c8 c9 ca cb cc cd ce cf d0 d1 95 2c c3 3f .....,,.?

      3      2.000000      10.1.1.142 -> 20.1.1.2      UDP Source port: 20001 Destination port:
20002

0000 54 75 d0 3a 85 3f 00 00 00 00 03 01 08 00 45 00 Tu...?.....E.
0010 00 ee 00 00 00 00 40 11 59 6e 0a 01 01 8e 14 01 .....@.Yn.....
0020 01 02 4e 21 4e 22 00 da 6e 29 00 01 02 03 04 05 ..N!N".n).....
0030 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 .....
0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 ..... !"#$$%
0050 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 &'()*+,-./012345
0060 36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45 6789:;<=>?@ABCDE
0070 46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 FGHIJKLMNOPQRSTU
0080 56 57 58 59 5a 5b 5c 5d 5e 5f 60 61 62 63 64 65 VWXYZ[\]^_`abcde
0090 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 fghijklmnopqrstu
00a0 76 77 78 79 7a 7b 7c 7d 7e 7f 80 81 82 83 84 85 vwxyz{|}~.....
00b0 86 87 88 89 8a 8b 8c 8d 8e 8f 90 91 92 93 94 95 .....
00c0 96 97 98 99 9a 9b 9c 9d 9e 9f a0 a1 a2 a3 a4 a5 .....
00d0 a6 a7 a8 a9 aa ab ac ad ae af b0 b1 b2 b3 b4 b5 .....
00e0 b6 b7 b8 b9 ba bb bc bd be bf c0 c1 c2 c3 c4 c5 .....
00f0 c6 c7 c8 c9 ca cb cc cd ce cf d0 d1 6c f8 dc 14 .....l...

      4      3.000000      10.1.1.143 -> 20.1.1.2      UDP Source port: 20001 Destination port:
20002

0000 54 75 d0 3a 85 3f 00 00 00 00 03 01 08 00 45 00 Tu...?.....E.
0010 00 ee 00 00 00 00 40 11 59 6d 0a 01 01 8f 14 01 .....@.Ym.....
0020 01 02 4e 21 4e 22 00 da 6e 28 00 01 02 03 04 05 ..N!N".n(.....
0030 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 .....
0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 ..... !"#$$%
0050 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 &'()*+,-./012345

```

Example: Displaying Packets from a .pcap File with a Display Filter

You can display the .pcap file packets output by entering:

```
Switch# show monitor capture file bootflash:mycap.pcap display-filter "ip.src ==
10.1.1.140" dump
```

```

1      0.000000    10.1.1.140 -> 20.1.1.2      UDP Source port: 20001  Destination port:
20002

0000  54 75 d0 3a 85 3f 00 00 00 00 03 01 08 00 45 00  Tu...?.....E.
0010  00 ee 00 00 00 00 40 11 59 70 0a 01 01 8c 14 01  .....@.Yp.....
0020  01 02 4e 21 4e 22 00 da 6e 2b 00 01 02 03 04 05  ..N!N"...n+.....
0030  06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15  .....
0040  16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25  ..... !"#$$%
0050  26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35  &'()*+,-./012345
0060  36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45  6789:;<=>?@ABCDE
0070  46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55  FGHIJKLMNOPQRSTU
0080  56 57 58 59 5a 5b 5c 5d 5e 5f 60 61 62 63 64 65  VWXYZ[\]^_`abcde
0090  66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75  fghijklmnopqrstu
00a0  76 77 78 79 7a 7b 7c 7d 7e 7f 80 81 82 83 84 85  vwxyz{|}~.....
00b0  86 87 88 89 8a 8b 8c 8d 8e 8f 90 91 92 93 94 95  .....
00c0  96 97 98 99 9a 9b 9c 9d 9e 9f a0 a1 a2 a3 a4 a5  .....
00d0  a6 a7 a8 a9 aa ab ac ad ae af b0 b1 b2 b3 b4 b5  .....
00e0  b6 b7 b8 b9 ba bb bc bd be bf c0 c1 c2 c3 c4 c5  .....
00f0  c6 c7 c8 c9 ca cb cc cd ce cf d0 d1 03 b0 7f 42  .....B

```

Usage Examples for Wireshark

Example: Simple Capture and Display

This example shows how to monitor traffic in the Layer 3 interface Gigabit 3/1:

Step 1 Define a capture point to match on the relevant traffic by entering:

```

Switch# monitor capture mycap interface gi 3/1 in match ipv4 any any
Switch# monitor capture mycap limit duration 60 packets 100

```



Note To avoid high CPU utilization, a low packet count and duration as limits has been set.

Step 2 Confirm that the capture point has been correctly defined by entering:

```

Switch# show monitor capture mycap parameter
      monitor capture mycap interface GigabitEthernet3/1 in
      monitor capture mycap match ipv4 any any
      monitor capture mycap limit packets 100 duration 60
Switch# show monitor capture mycap
Status Information for Capture mycap
Target Type:
  Interface: GigabitEthernet3/1, Direction: in
Status : Inactive
Filter Details:
  IPv4
    Source IP: any
    Destination IP: any
    Protocol: any
File Details:
  File not associated
Buffer Details:
  Buffer Type: LINEAR (default)
Limit Details:
  Number of Packets to capture: 100
  Packet Capture duration: 60

```


Step 3 Start the capture process and display the results.

```
Switch# monitor capture mycap start display
0.000000 10.1.1.30 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
1.000000 10.1.1.31 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
2.000000 10.1.1.32 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
3.000000 10.1.1.33 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
4.000000 10.1.1.34 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
5.000000 10.1.1.35 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
6.000000 10.1.1.36 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
7.000000 10.1.1.37 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
8.000000 10.1.1.38 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
9.000000 10.1.1.39 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
```

Step 4 Delete the capture point by entering:

```
Switch# no monitor capture mycap
```

Example: Simple Capture and Store

This example shows how to capture packets to a filter.

Step 1 Define a capture point to match on the relevant traffic and associate it to a file by entering:

```
Switch# monitor capture mycap interface gi 3/1 in match ipv4 any any
Switch# monitor capture mycap limit duration 60 packets 100
Switch# monitor cap mycap file location bootflash:mycap.pcap
```

Step 2 Confirm that the capture point has been correctly defined by entering:

```
Switch# show monitor capture mycap parameter
monitor capture mycap interface GigabitEthernet3/1 in
monitor capture mycap match ipv4 any any
monitor capture mycap file location bootflash:mycap.pcap
monitor capture mycap limit packets 100 duration 60
Switch# show monitor capture mycap
Target Type:
Interface: GigabitEthernet3/1, Direction: in
Status : Inactive
Filter Details:
IPv4
Source IP: any
Destination IP: any
Protocol: any
File Details:
Associated file name: bootflash:mycap.pcap
Buffer Details:
Buffer Type: LINEAR (default)
Limit Details:
Number of Packets to capture: 100
Packet Capture duration: 60
```

Step 3 Launch packet capture by entering:

```
Switch# monitor capture mycap start
```

Step 4 After sufficient time has passed, stop the capture by entering:

```
Switch# monitor capture mycap stop
```

**Note**

Alternatively, you could let the capture operation stop automatically after the time has elapsed or the packet count has been met.

The **mycap.pcap** file now contains the captured packets.

Step 5 Display the packets by entering:

```
Switch# show monitor capture file bootflash:mycap.pcap
0.000000 10.1.1.30 -> 20.1.1.2      UDP Source port: 20001 Destination port: 20002
1.000000 10.1.1.31 -> 20.1.1.2      UDP Source port: 20001 Destination port: 20002
2.000000 10.1.1.32 -> 20.1.1.2      UDP Source port: 20001 Destination port: 20002
3.000000 10.1.1.33 -> 20.1.1.2      UDP Source port: 20001 Destination port: 20002
4.000000 10.1.1.34 -> 20.1.1.2      UDP Source port: 20001 Destination port: 20002
5.000000 10.1.1.35 -> 20.1.1.2      UDP Source port: 20001 Destination port: 20002
6.000000 10.1.1.36 -> 20.1.1.2      UDP Source port: 20001 Destination port: 20002
7.000000 10.1.1.37 -> 20.1.1.2      UDP Source port: 20001 Destination port: 20002
8.000000 10.1.1.38 -> 20.1.1.2      UDP Source port: 20001 Destination port: 20002
9.000000 10.1.1.39 -> 20.1.1.2      UDP Source port: 20001 Destination port: 20002
```

Step 6 Delete the capture point by entering:

```
Switch# no monitor capture mycap
```

Example: Using Buffer Capture

This example shows how to use buffer capture:

Step 1 Launch a capture session with the buffer capture option by entering:

```
Switch# monitor capture mycap interface gi 3/1 in
Switch# monitor capture mycap match ipv4 any any
Switch# monitor capture mycap buffer circular size 1
Switch# monitor capture mycap start
```

Step 2 Determine whether the capture is active by entering:

```
Switch# show monitor capture mycap

Status Information for Capture mycap
Target Type:
  Interface: GigabitEthernet3/1, Direction: in
Status : Active
Filter Details:
  IPv4
    Source IP: any
    Destination IP: any
  Protocol: any
File Details:
  File not associated
Buffer Details:
  Buffer Type: CIRCULAR
  Buffer Size (in MB): 1
Limit Details:
  limit not set
```

Step 3 Display the packets in the buffer by entering:

```
Switch# show monitor capture mycap buffer brief
0.000000 10.1.1.215 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
1.000000 10.1.1.216 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
2.000000 10.1.1.217 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
3.000000 10.1.1.218 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
4.000000 10.1.1.219 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
5.000000 10.1.1.220 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
6.000000 10.1.1.221 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
7.000000 10.1.1.222 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
8.000000 10.1.1.223 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
9.000000 10.1.1.224 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
10.000000 10.1.1.225 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
11.000000 10.1.1.226 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
12.000000 10.1.1.227 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
13.000000 10.1.1.228 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
14.000000 10.1.1.229 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
15.000000 10.1.1.230 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
16.000000 10.1.1.231 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
17.000000 10.1.1.232 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
18.000000 10.1.1.233 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
19.000000 10.1.1.234 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
20.000000 10.1.1.235 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
21.000000 10.1.1.236 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
```

Notice that the packets have been buffered.

Step 4 Display the packets in other display modes.

```
Switch# show monitor capture mycap buffer detailed
Frame 1: 256 bytes on wire (2048 bits), 256 bytes captured (2048 bits)
  Arrival Time: Apr 15, 2012 15:50:02.398966000 PDT
  Epoch Time: 1334530202.398966000 seconds
  [Time delta from previous captured frame: 0.000000000 seconds]
  [Time delta from previous displayed frame: 0.000000000 seconds]
  [Time since reference or first frame: 0.000000000 seconds]
  Frame Number: 1
  Frame Length: 256 bytes (2048 bits)
  Capture Length: 256 bytes (2048 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  [Protocols in frame: eth:ip:udp:data]
Ethernet II, Src: 00:00:00:00:03:01 (00:00:00:00:03:01), Dst: 54:75:d0:3a:85:3f
(54:75:d0:3a:85:3f)
  Destination: 54:75:d0:3a:85:3f (54:75:d0:3a:85:3f)
    Address: 54:75:d0:3a:85:3f (54:75:d0:3a:85:3f)
      .... 0 .... = IG bit: Individual address (unicast)
      .... 0 .... = LG bit: Globally unique address (factory default)
  Source: 00:00:00:00:03:01 (00:00:00:00:03:01)
    Address: 00:00:00:00:03:01 (00:00:00:00:03:01)
      .... 0 .... = IG bit: Individual address (unicast)
      .... 0 .... = LG bit: Globally unique address (factory default)
...
Switch# show monitor capture mycap buffer dump
0.000000 10.1.1.215 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002

0000 54 75 d0 3a 85 3f 00 00 00 00 03 01 08 00 45 00 Tu...?.....E.
0010 00 ee 00 00 00 00 40 11 59 25 0a 01 01 d7 14 01 .....@.Y%.....
0020 01 02 4e 21 4e 22 00 da 6d e0 00 01 02 03 04 05 ..N!N".m.....
0030 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 .....
0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 ..... !"#$$%
0050 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 &'()*+,-./012345
0060 36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45 6789:;<=>?@ABCDE
0070 46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 FGHIJKLMNOPQRSTU
```

```

0080 56 57 58 59 5a 5b 5c 5d 5e 5f 60 61 62 63 64 65  VWXYZ[\]^_`abcde
0090 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75  fghijklmnopqrstu
00a0 76 77 78 79 7a 7b 7c 7d 7e 7f 80 81 82 83 84 85  vwxyz{|}~.....
00b0 86 87 88 89 8a 8b 8c 8d 8e 8f 90 91 92 93 94 95  .....
00c0 96 97 98 99 9a 9b 9c 9d 9e 9f a0 a1 a2 a3 a4 a5  .....
00d0 a6 a7 a8 a9 aa ab ac ad ae af b0 b1 b2 b3 b4 b5  .....
00e0 b6 b7 b8 b9 ba bb bc bd be bf c0 c1 c2 c3 c4 c5  .....
00f0 c6 c7 c8 c9 ca cb cc cd ce cf d0 d1 03 3e d0 33  .....>.3

```

Step 5 Clear the buffer once, wait for 10 seconds, then stop the traffic by entering:

```
Switch# monitor capture mycap clear
```

Wait for 10 seconds and stop the traffic.

Confirm that the same set of packets are displayed after this time gap by entering:

```

Switch# show monitor capture mycap buffer brief
0.000000 10.1.1.2 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
1.000000 10.1.1.3 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
2.000000 10.1.1.4 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
3.000000 10.1.1.5 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
4.000000 10.1.1.6 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
5.000000 10.1.1.7 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
6.000000 10.1.1.8 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
7.000000 10.1.1.9 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
8.000000 10.1.1.10 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
9.000000 10.1.1.11 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002

```

[Wait for about 10 secs]

```

Switch# show monitor capture mycap buffer brief
0.000000 10.1.1.2 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
1.000000 10.1.1.3 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
2.000000 10.1.1.4 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
3.000000 10.1.1.5 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
4.000000 10.1.1.6 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
5.000000 10.1.1.7 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
6.000000 10.1.1.8 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
7.000000 10.1.1.9 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
8.000000 10.1.1.10 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
9.000000 10.1.1.11 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002

```

[Wait for about 10 secs]

```

Switch# show monitor capture mycap buffer brief
0.000000 10.1.1.2 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
1.000000 10.1.1.3 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
2.000000 10.1.1.4 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
3.000000 10.1.1.5 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
4.000000 10.1.1.6 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
5.000000 10.1.1.7 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
6.000000 10.1.1.8 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
7.000000 10.1.1.9 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
8.000000 10.1.1.10 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
9.000000 10.1.1.11 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002

```

Step 6 Clear the packets from the buffer by entering:

```
Switch# monitor capture mycap clear
```

Step 7 Confirm that the buffer is now empty by entering:

```
Switch# show monitor capture mycap buffer brief
```

Wait about 10 seconds.

Step 8 Display the buffer contents.

```
Switch# show monitor capture mycap buffer brief
```

Step 9 Restart the traffic, wait about 10 seconds, then display buffer contents by entering:

```
Switch# show monitor capture mycap buffer brief
0.000000 10.1.1.2 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
1.000000 10.1.1.3 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
2.000000 10.1.1.4 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
3.000000 10.1.1.5 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
4.000000 10.1.1.6 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
5.000000 10.1.1.7 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
6.000000 10.1.1.8 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
7.000000 10.1.1.9 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
8.000000 10.1.1.10 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
9.000000 10.1.1.11 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
10.000000 10.1.1.12 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
```

Step 10 Store the buffer contents to the mycap1.pcap file in the internal bootflash: storage device.

```
Switch# monitor capture mycap export bootflash:mycap1.pcap
Exported Successfully
```

Step 11 Check that the file has been created and that it contains the packets by entering:

```
Switch# dir bootflash:mycap1.pcap
Directory of bootflash:/mycap1.pcap

14758  -rw-          20152  Apr 15 2012 16:00:28 -07:00  mycap1.pcap

831541248 bytes total (831340544 bytes free)
Switch# show monitor capture file bootflash:mycap1.pcap brief
 1  0.000000 10.1.1.2 -> 20.1.1.2 UDP Source port: 20001 Destination port:
20002
 2  1.000000 10.1.1.3 -> 20.1.1.2 UDP Source port: 20001 Destination port:
20002
 3  2.000000 10.1.1.4 -> 20.1.1.2 UDP Source port: 20001 Destination port:
20002
 4  3.000000 10.1.1.5 -> 20.1.1.2 UDP Source port: 20001 Destination port:
20002
 5  4.000000 10.1.1.6 -> 20.1.1.2 UDP Source port: 20001 Destination port:
20002
 6  5.000000 10.1.1.7 -> 20.1.1.2 UDP Source port: 20001 Destination port:
20002
 7  6.000000 10.1.1.8 -> 20.1.1.2 UDP Source port: 20001 Destination port:
20002
 8  7.000000 10.1.1.9 -> 20.1.1.2 UDP Source port: 20001 Destination port:
20002
 9  8.000000 10.1.1.10 -> 20.1.1.2 UDP Source port: 20001 Destination port:
20002
10  9.000000 10.1.1.11 -> 20.1.1.2 UDP Source port: 20001 Destination port:
20002
11 10.000000 10.1.1.12 -> 20.1.1.2 UDP Source port: 20001 Destination port:
20002
12 11.000000 10.1.1.13 -> 20.1.1.2 UDP Source port: 20001 Destination port:
20002
13 12.000000 10.1.1.14 -> 20.1.1.2 UDP Source port: 20001 Destination port:
20002
14 13.000000 10.1.1.15 -> 20.1.1.2 UDP Source port: 20001 Destination port:
20002
15 14.000000 10.1.1.16 -> 20.1.1.2 UDP Source port: 20001 Destination port:
20002
```

```
16 15.000000 10.1.1.17 -> 20.1.1.2 UDP Source port: 20001 Destination port:
20002
```

Step 12 Stop the packet capture and display the buffer contents by entering:

```
Switch# monitor capture mycap stop
Switch# show monitor capture mycap buffer brief
0.000000 10.1.1.2 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
1.000000 10.1.1.3 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
2.000000 10.1.1.4 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
3.000000 10.1.1.5 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
4.000000 10.1.1.6 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
5.000000 10.1.1.7 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
6.000000 10.1.1.8 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
7.000000 10.1.1.9 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
8.000000 10.1.1.10 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
9.000000 10.1.1.11 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
10.000000 10.1.1.12 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
11.000000 10.1.1.13 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
```

Step 13 Clear the buffer and then try to display packets from the buffer by entering:

```
Switch# monitor capture mycap clear
Switch# show monitor capture mycap buffer brief
```

Step 14 Delete the capture point by entering:

```
Switch# no monitor capture mycap
```

Example: Capture Sessions

The following examples shows how to start or stop a capture session in various modes:

```
Switch# monitor capture mycap int gi 3/1 in match ipv4 any any
Switch# monitor capture mycap file location bootflash:mycap.pcap
Switch# monitor capture mycap limit packets 100 duration 60

Switch# monitor capture mycap start
Switch#
Switch# monitor capture mycap stop
Switch# monitor capture mycap start capture-filter "udp.port == 20001"
Switch# monitor capture mycap stop
Switch# monitor capture mycap start capture-filter "udp.port == 20001" display
A file by the same capture file name already exists, overwrite?[confirm]

0.000000 10.1.1.9 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
0.000000 10.1.1.10 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
0.000000 10.1.1.11 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
0.000000 10.1.1.12 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
0.000000 10.1.1.13 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
0.000000 10.1.1.14 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
0.000000 10.1.1.15 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
0.000000 10.1.1.16 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
0.000000 10.1.1.17 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
0.000000 10.1.1.18 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
0.000000 10.1.1.19 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
0.000000 10.1.1.20 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
0.000000 10.1.1.21 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
0.000000 10.1.1.22 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
0.000000 10.1.1.23 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
0.000000 10.1.1.24 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
```

```

0.000000 10.1.1.25 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
0.000000 10.1.1.26 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
0.000000 10.1.1.27 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
0.000000 10.1.1.28 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
0.000000 10.1.1.29 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
0.000000 10.1.1.30 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002

```

```

Switch# monitor capture mycap start capture-filter "udp.port == 20001" display
display-filter "udp.port == 20002"
%Display-filter cannot be specified when capture is associated to a file. Ignoring
display filter%
A file by the same capture file name already exists, overwrite?[confirm]

```

```

0.000000 10.1.1.96 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
0.000000 10.1.1.97 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
0.000000 10.1.1.98 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
0.000000 10.1.1.99 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
0.000000 10.1.1.100 -> 20.1.1.2   UDP Source port: 20001 Destination port: 20002
0.000000 10.1.1.101 -> 20.1.1.2   UDP Source port: 20001 Destination port: 20002
0.000000 10.1.1.102 -> 20.1.1.2   UDP Source port: 20001 Destination port: 20002
0.000000 10.1.1.103 -> 20.1.1.2   UDP Source port: 20001 Destination port: 20002
0.000000 10.1.1.104 -> 20.1.1.2   UDP Source port: 20001 Destination port: 20002
0.000000 10.1.1.105 -> 20.1.1.2   UDP Source port: 20001 Destination port: 20002
0.000000 10.1.1.106 -> 20.1.1.2   UDP Source port: 20001 Destination port: 20002
0.000000 10.1.1.107 -> 20.1.1.2   UDP Source port: 20001 Destination port: 20002
0.000000 10.1.1.108 -> 20.1.1.2   UDP Source port: 20001 Destination port: 20002
0.000000 10.1.1.109 -> 20.1.1.2   UDP Source port: 20001 Destination port: 20002

```

```

Switch# monitor capture mycap start capture-filter "udp.port == 20001" display
display-filter "udp.port == 20002" detailed
%Display-filter cannot be specified when capture is associated to a file. Ignoring
display filter%
A file by the same capture file name already exists, overwrite?[confirm]

```

```

Frame 1: 256 bytes on wire (2048 bits), 256 bytes captured (2048 bits)
  Arrival Time: Dec 31, 1969 17:00:00.000000000 PDT
  Epoch Time: 0.000000000 seconds
  [Time delta from previous captured frame: 0.000000000 seconds]
  [Time delta from previous displayed frame: 0.000000000 seconds]
  [Time since reference or first frame: 0.000000000 seconds]
  Frame Number: 1
  Frame Length: 256 bytes (2048 bits)
  Capture Length: 256 bytes (2048 bits)
  [Frame is marked: False]
  [Frame is ignored: False]
  [Protocols in frame: eth:ip:udp:data]
Ethernet II, Src: 00:00:00:00:03:01 (00:00:00:00:03:01), Dst: 54:75:d0:3a:85:3f
(54:75:d0:3a:85:3f)
  Destination: 54:75:d0:3a:85:3f (54:75:d0:3a:85:3f)
    Address: 54:75:d0:3a:85:3f (54:75:d0:3a:85:3f)
      ....0 .... = IG bit: Individual address (unicast)
      ....0. .... = LG bit: Globally unique address (factory default)
    Source: 00:00:00:00:03:01 (00:00:00:00:03:01)
      Address: 00:00:00:00:03:01 (00:00:00:00:03:01)
        ....0 .... = IG bit: Individual address (unicast)
        ....0. .... = LG bit: Globally unique address (factory default)

```

```

Switch# monitor capture mycap start capture-filter "udp.port == 20001" display dump
A file by the same capture file name already exists, overwrite?[confirm]

```

```

0.000000 10.1.1.6 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002

0000  54 75 d0 3a 85 3f 00 00 00 00 03 01 08 00 45 00   Tu...?.....E.
0010  00 ee 00 00 00 00 40 11 59 f6 0a 01 01 06 14 01   .....@.Y.....

```

```

0020 01 02 4e 21 4e 22 00 da 6e b1 00 01 02 03 04 05  ..N!N"..n.....
0030 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15  .....
0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25  ..... !"#$$
0050 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35  &'()*+,-./012345
0060 36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45  6789:;<=>?@ABCDE
0070 46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55  FGHIJKLMNOPQRSTU
0080 56 57 58 59 5a 5b 5c 5d 5e 5f 60 61 62 63 64 65  VWXYZ[\]^_`abcde
0090 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75  fghijklmnopqrstu
00a0 76 77 78 79 7a 7b 7c 7d 7e 7f 80 81 82 83 84 85  vwxyz{|}~.....
00b0 86 87 88 89 8a 8b 8c 8d 8e 8f 90 91 92 93 94 95  .....
00c0 96 97 98 99 9a 9b 9c 9d 9e 9f a0 a1 a2 a3 a4 a5  .....
00d0 a6 a7 a8 a9 aa ab ac ad ae af b0 b1 b2 b3 b4 b5  .....
00e0 b6 b7 b8 b9 ba bb bc bd be bf c0 c1 c2 c3 c4 c5  .....
00f0 c6 c7 c8 c9 ca cb cc cd ce cf d0 d1 ac 69 6e fd  .....in.

```

```
0.000000 10.1.1.7 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
```

```

Switch# monitor capture mycap start display display-filter "udp.port == 20002"
%Display-filter cannot be specified when capture is associated to a file. Ignoring
display filter%
A file by the same capture file name already exists, overwrite?[confirm]

```

```

0.000000 10.1.1.41 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
1.000000 10.1.1.42 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
2.000000 10.1.1.43 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
3.000000 10.1.1.44 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
4.000000 10.1.1.45 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
5.000000 10.1.1.46 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
5.998993 10.1.1.47 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
6.998993 10.1.1.48 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
7.998993 10.1.1.49 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
8.998993 10.1.1.50 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
9.998993 10.1.1.51 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
10.998993 10.1.1.52 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002

```

```

Switch# monitor capture mycap start display display-filter "udp.port == 20002" dump
%Display-filter cannot be specified when capture is associated to a file. Ignoring
display filter%
A file by the same capture file name already exists, overwrite?[confirm]

```

```

0.000000 10.1.1.117 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002

0000 54 75 d0 3a 85 3f 00 00 00 00 03 01 08 00 45 00  Tu.:.?.....E.
0010 00 ee 00 00 00 00 40 11 59 87 0a 01 01 75 14 01  .....@.Y....u..
0020 01 02 4e 21 4e 22 00 da 6e 42 00 01 02 03 04 05  ..N!N"..nB.....
0030 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15  .....
0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25  ..... !"#$$
0050 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35  &'()*+,-./012345
0060 36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45  6789:;<=>?@ABCDE
0070 46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55  FGHIJKLMNOPQRSTU
0080 56 57 58 59 5a 5b 5c 5d 5e 5f 60 61 62 63 64 65  VWXYZ[\]^_`abcde
0090 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75  fghijklmnopqrstu
00a0 76 77 78 79 7a 7b 7c 7d 7e 7f 80 81 82 83 84 85  vwxyz{|}~.....
00b0 86 87 88 89 8a 8b 8c 8d 8e 8f 90 91 92 93 94 95  .....
00c0 96 97 98 99 9a 9b 9c 9d 9e 9f a0 a1 a2 a3 a4 a5  .....
00d0 a6 a7 a8 a9 aa ab ac ad ae af b0 b1 b2 b3 b4 b5  .....
00e0 b6 b7 b8 b9 ba bb bc bd be bf c0 c1 c2 c3 c4 c5  .....
00f0 c6 c7 c8 c9 ca cb cc cd ce cf d0 d1 41 0c b4 5d  .....A..]

```

```
1.000000 10.1.1.118 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
```

```
Switch# no monitor capture mycap file
```



```
Switch# monitor capture mycap start display display-filter "udp.port == 20002" dump

0.000000 10.1.1.160 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
```

```
0000 54 75 d0 3a 85 3f 00 00 00 00 03 01 08 00 45 00 Tu...?.....E.
0010 00 ee 00 00 00 00 40 11 59 5c 0a 01 01 a0 14 01 .....@.Y\.....
0020 01 02 4e 21 4e 22 00 da 6e 17 00 01 02 03 04 05 ..N!N"..n.....
0030 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 .....
0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 ..... !"#$$
0050 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 &'()*+,-./012345
0060 36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45 6789:;<=>?@ABCDE
0070 46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 FGHIJKLMNOPQRSTU
0080 56 57 58 59 5a 5b 5c 5d 5e 5f 60 61 62 63 64 65 VWXYZ[\]^_`abcde
0090 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 fghijklmnopqrstu
00a0 76 77 78 79 7a 7b 7c 7d 7e 7f 80 81 82 83 84 85 vwxyz{|}~.....
00b0 86 87 88 89 8a 8b 8c 8d 8e 8f 90 91 92 93 94 95 .....
00c0 96 97 98 99 9a 9b 9c 9d 9e 9f a0 a1 a2 a3 a4 a5 .....
00d0 a6 a7 a8 a9 aa ab ac ad ae af b0 b1 b2 b3 b4 b5 .....
00e0 b6 b7 b8 b9 ba bb bc bd be bf c0 c1 c2 c3 c4 c5 .....
00f0 c6 c7 c8 c9 ca cb cc cd ce cf d0 d1 9f 20 8a e5 .....
```

```
1.000000 10.1.1.161 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
```

```
Switch# monitor capture mycap start display display-filter "udp.port == 20002"
```

```
0.000000 10.1.1.173 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
1.000000 10.1.1.174 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
2.000000 10.1.1.175 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
3.000000 10.1.1.176 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
4.000000 10.1.1.177 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
5.000000 10.1.1.178 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
6.000000 10.1.1.179 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
7.000000 10.1.1.180 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
8.000000 10.1.1.181 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
9.000000 10.1.1.182 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
10.000000 10.1.1.183 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
11.000000 10.1.1.184 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
12.000000 10.1.1.185 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
```

```
Switch# monitor capture mycap start display detailed
```

```
Frame 1: 256 bytes on wire (2048 bits), 256 bytes captured (2048 bits)
Arrival Time: Apr 12, 2012 11:46:54.245974000 PDT
Epoch Time: 1334256414.245974000 seconds
[Time delta from previous captured frame: 0.000000000 seconds]
[Time delta from previous displayed frame: 0.000000000 seconds]
[Time since reference or first frame: 0.000000000 seconds]
Frame Number: 1
Frame Length: 256 bytes (2048 bits)
Capture Length: 256 bytes (2048 bits)
[Frame is marked: False]
[Frame is ignored: False]
[Protocols in frame: eth:ip:udp:data]
Ethernet II, Src: 00:00:00:00:03:01 (00:00:00:00:03:01), Dst: 54:75:d0:3a:85:3f
(54:75:d0:3a:85:3f)
Destination: 54:75:d0:3a:85:3f (54:75:d0:3a:85:3f)
Address: 54:75:d0:3a:85:3f (54:75:d0:3a:85:3f)
.... 0 .... = IG bit: Individual address (unicast)
.... 0 .... = LG bit: Globally unique address (factory default)
```

```

Source: 00:00:00:00:03:01 (00:00:00:00:03:01)
Address: 00:00:00:00:03:01 (00:00:00:00:03:01)
....0 .... = IG bit: Individual address (unicast)
....0. .... = LG bit: Globally unique address (factory default)

```

Switch#

Example: Capture and Store in Lock-step Mode

This example captures live traffic and stores the packets in lock-step mode to achieve a high capture rate.



Note

The capture rate might be slow for the first 15 seconds. If possible and needed, start the traffic 15 seconds after the capture session has started.

Step 1 Define a capture point to match on the relevant traffic and associate it to a file by entering:

```

Switch# monitor capture mycap interface gi 3/1 in match ipv4 any any
Switch# monitor capture mycap limit duration 60 packets 100
Switch# monitor cap mycap file location bootflash:mycap.pcap buffer-size 64

```

Step 2 Confirm that the capture point has been correctly defined by entering:

```

Switch# show monitor capture mycap parameter
  monitor capture mycap interface GigabitEthernet3/1 in
  monitor capture mycap match ipv4 any any
  monitor capture mycap file location bootflash:mycap.pcap buffer-size 64
  monitor capture mycap limit packets 100000 duration 60
Switch# show monitor capture mycap
Target Type:
  Interface: GigabitEthernet3/1, Direction: in
Status : Inactive
Filter Details:
  IPv4
  Source IP: any
  Destination IP: any
  Protocol: any
File Details:
  Associated file name: bootflash:mycap.pcap
Buffer Details:
  Buffer Type: LINEAR (default)
Limit Details:
  Number of Packets to capture: 100
  Packet Capture duration: 60

```

Step 3 Launch packet capture by entering:

```
Switch# monitor capture mycap start
```

Let the capture operation stop automatically after the time has elapsed or the packet count has been met.

The **mycap.pcap** file now contains the captured packets.

Step 4 Display the packets by entering:

```

Switch# show monitor capture file bootflash:mycap.pcap
0.000000 10.1.1.30 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
1.000000 10.1.1.31 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
2.000000 10.1.1.32 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
3.000000 10.1.1.33 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002

```

```

4.000000  10.1.1.34 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
5.000000  10.1.1.35 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
6.000000  10.1.1.36 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
7.000000  10.1.1.37 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
8.000000  10.1.1.38 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002
9.000000  10.1.1.39 -> 20.1.1.2    UDP Source port: 20001 Destination port: 20002

```

Step 5 Delete the capture point by entering:

```
Switch# no monitor capture mycap
```

Example: Simple Capture and Store in Lock-step with High-speed Mode

This example shows how to capture live traffic and store the packets in lock-step with high-speed mode to achieve high capture rate.



Note

The capture rate might be slow for the first 15 seconds. If possible and necessary, start the traffic 15 seconds after the capture session starts.

Step 1 Define a capture point to match on the relevant traffic and associate it to a file by entering:

```

Switch# monitor capture mycap interface gi 3/1 in match ipv4 any any
Switch# monitor capture mycap limit duration 60 packets 100
Switch# monitor cap mycap file location bootflash:mycap.pcap buffer-size 90

```

Step 2 Confirm that the capture point has been correctly defined by entering:

```

Switch# show monitor capture mycap parameter
  monitor capture mycap interface GigabitEthernet3/1 in
  monitor capture mycap match ipv4 any any
  monitor capture mycap file location bootflash:mycap.pcap buffer-size 90
  monitor capture mycap limit packets 100000 duration 60
Switch# show monitor capture mycap
Target Type:
  Interface: GigabitEthernet3/1, Direction: in
Status : Inactive
Filter Details:
  IPv4
    Source IP: any
    Destination IP: any
    Protocol: any
File Details:
  Associated file name: bootflash:mycap.pcap
Buffer Details:
  Buffer Type: LINEAR (default)
Limit Details:
  Number of Packets to capture: 100
  Packet Capture duration: 60

```

Step 3 Launch packet capture by entering:

```
Switch# monitor capture mycap start
```

Let the capture operation stop automatically after the time has elapsed or the packet count has been met.

The **mycap.pcap** file now contains the captured packets.

Step 4 Display the packets by entering:

```
Switch# show monitor capture file bootflash:mycap.pcap
0.000000 10.1.1.30 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
1.000000 10.1.1.31 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
2.000000 10.1.1.32 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
3.000000 10.1.1.33 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
4.000000 10.1.1.34 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
5.000000 10.1.1.35 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
6.000000 10.1.1.36 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
7.000000 10.1.1.37 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
8.000000 10.1.1.38 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
9.000000 10.1.1.39 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
```

Step 5 Delete the capture point by entering:

```
Switch# no monitor capture mycap
```

Example: Simple Capture and Store of Packets in Egress Direction

This example shows how to capture live traffic and store the packets in egress direction using lock-step with high-speed mode.

Step 1 Define a capture point to match on the relevant traffic and associate it to a file by entering:

```
Switch# monitor capture mycap interface gi 3/1 out match ipv4 any any
Switch# monitor capture mycap limit duration 60 packets 100
Switch# monitor cap mycap file location bootflash:mycap.pcap buffer-size 90
```

Step 2 Confirm that the capture point has been correctly defined by entering:

```
Switch# show monitor capture mycap parameter
monitor capture mycap interface GigabitEthernet3/1 out
monitor capture mycap match ipv4 any any
monitor capture mycap file location bootflash:mycap.pcap buffer-size 90
monitor capture mycap limit packets 100000 duration 60
Switch# show monitor capture mycap
Target Type:
Interface: GigabitEthernet3/1, Direction: out
Status : Inactive
Filter Details:
IPv4
Source IP: any
Destination IP: any
Protocol: any
File Details:
Associated file name: bootflash:mycap.pcap
Buffer Details:
Buffer Type: LINEAR (default)
Limit Details:
Number of Packets to capture: 100
Packet Capture duration: 60
```

Step 3 Launch packet capture by entering:

```
Switch# monitor capture mycap start
```



Note

Let the capture operation stop automatically after the time has elapsed or the packet count has been met.

The mycap.pcap file now contains the captured packets.

Step 4 Display the packets by entering:

```
Switch# show monitor capture file bootflash:mycap.pcap
0.000000 10.1.1.30 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
1.000000 10.1.1.31 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
2.000000 10.1.1.32 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
3.000000 10.1.1.33 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
4.000000 10.1.1.34 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
5.000000 10.1.1.35 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
6.000000 10.1.1.36 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
7.000000 10.1.1.37 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
8.000000 10.1.1.38 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
9.000000 10.1.1.39 -> 20.1.1.2 UDP Source port: 20001 Destination port: 20002
```

Step 5 Delete the capture point by entering:

```
Switch# no monitor capture mycap
```

VSS Specific Examples

Example: Capturing and Storing in a file

This example shows how to do a simple capture and store operation after the attachment point is issued on the VSS active switch:

Step 1 Launch a capture session by entering the following commands in VSS active switch:

```
vss_dut1# monitor capture mycap interface gi 1/1/1 in
vss_dut1# monitor capture mycap match ipv4 any any
vss_dut1# monitor capture mycap file location bootflash:mycap.pcap
vss_dut1# monitor capture mycap limit packets 10 duration 10
vss_dut1# monitor capture mycap start
```

Step 2 After the capture session concludes, check that the capture file has stored the packets:

```
*Nov 15 00:04:08.337 PDT: %BUFCAP-6-ENABLE: Capture Point mycap enabled.
*Nov 15 00:04:08.339 PDT: Policy name = mycap, Instance ID = 4
vss_dut1#
*Nov 15 00:04:13.736 PDT: %BUFCAP-6-DISABLE_ASYNC: Capture Point mycap disabled. Reason :
Wireshark Session Ended
vss_dut1# dir bootflash:mycap.pcap
Directory of bootflash:/mycap.pcap
72971 -rw-      824 Nov 15 2012 00:04:13 -07:00 mycap.pcap
822910976 bytes total (304648192 bytes free)
```

Example: Capturing and Storing in a File with Display

This example shows how to perform a basic capture and store operation with the display option in brief mode after the attachment point is issued on the VSS active switch:

Step 1 Prepare a capture session by entering the following commands in VSS active switch:

```
vss_dut1# monitor capture mycap interface gi 1/1/1 in
vss_dut1# monitor capture mycap match ipv4 any any
```

```
vss_dut1# monitor capture mycap file location bootflash:mycap.pcap
vss_dut1# monitor capture mycap limit packets 60 duration 60
```

Step 2 Start the capture session with display option in brief mode:

```
vss_dut1# monitor capture mycap start display

*Nov 14 23:43:20.506 PDT: Policy name = mycap, Instance ID = 3
 0.000000 20.1.1.2 -> 30.1.1.2 TCP [TCP ZeroWindow] 0 > 0 [<None>] Seq=1 Win=0
Len=6
 0.595022 20.1.1.2 -> 30.1.1.2 TCP [TCP ZeroWindow] [TCP Retransmission] 0 > 0
[<None>] Seq=1 Win=0 Len=6
 1.012008 20.1.1.2 -> 30.1.1.2 TCP [TCP ZeroWindow] [TCP Retransmission] 0 > 0
[<None>] Seq=1 Win=0 Len=6
 1.500026 20.1.1.2 -> 30.1.1.2 TCP [TCP ZeroWindow] [TCP Retransmission] 0 > 0
[<None>] Seq=1 Win=0 Len=6
 2.005005 20.1.1.2 -> 30.1.1.2 TCP [TCP ZeroWindow] [TCP Retransmission] 0 > 0
[<None>] Seq=1 Win=0 Len=6
 2.500026 20.1.1.2 -> 30.1.1.2 TCP [TCP ZeroWindow] [TCP Retransmission] 0 > 0
[<None>] Seq=1 Win=0 Len=6
 3.000000 20.1.1.2 -> 30.1.1.2 TCP [TCP ZeroWindow] [TCP Retransmission] 0 > 0
[<None>] Seq=1 Win=0 Len=6
...
```

Example: Circular Buffer Usage

This example shows how to do a simple capture and store operation with display option after the attachment point is issued on the VSS active switch:

Step 1 Prepare the capture session by entering the following commands in VSS active switch:

```
vss_dut1# monitor capture mycap interface gi 2/1/1 in
vss_dut1# monitor capture mycap match ipv4 any any
vss_dut1# monitor capture mycap buffer size 1 circular
vss_dut1# monitor capture mycap limit packets 10
```

Step 2 Start the session in VSS active switch. Periodically, check that the packets are stored in the capture file.

```
vss_dut1# monitor capture mycap start
vss_dut1# show monitor capture mycap buffer
0.000000 10.10.10.20 -> 10.10.10.10 TCP [TCP ZeroWindow] 2000 > 3000 [<None>] Seq=1
Win=0 Len=6[Malformed Packet]
 0.000000 10.10.10.20 -> 10.10.10.10 TCP [TCP ZeroWindow] [TCP Out-Of-Order] 2000 >
3000 [<None>] Seq=1 Win=0 Len=6[Malformed Packet]
 0.000000 10.10.10.20 -> 10.10.10.10 TCP [TCP ZeroWindow] [TCP Out-Of-Order] 2000 >
3000 [<None>] Seq=1 Win=0 Len=6[Malformed Packet]
 0.000000 10.10.10.20 -> 10.10.10.10 TCP [TCP ZeroWindow] [TCP Out-Of-Order] 2000 >
3000 [<None>] Seq=1 Win=0 Len=6[Malformed Packet]

vss_dut1# show monitor capture mycap buffer detailed
Frame 1: 64 bytes on wire (512 bits), 64 bytes captured (512 bits)
Arrival Time: Oct 15, 2018 17:12:27.182989000 UTC
Epoch Time: 1539623547.182989000 seconds
[Time delta from previous captured frame: 0.000000000 seconds]
[Time delta from previous displayed frame: 0.000000000 seconds]
[Time since reference or first frame: 0.000000000 seconds]
Frame Number: 1
Frame Length: 64 bytes (512 bits)
Capture Length: 64 bytes (512 bits)
[Frame is marked: False]
```

```

[Frame is ignored: False]
[Protocols in frame: eth:ip:tcp:skinny]
Ethernet II, Src: 00:00:0c:00:02:00 (00:00:0c:00:02:00), Dst: 00:00:0c:00:03:00
(00:00:0c:00:03:00)
  Destination: 00:00:0c:00:03:00 (00:00:0c:00:03:00)
    Address: 00:00:0c:00:03:00 (00:00:0c:00:03:00)
      .... 0 .... = IG bit: Individual address (unicast)
      .... 0. .... = LG bit: Globally unique address (factory default)
  Source: 00:00:0c:00:02:00 (00:00:0c:00:02:00)
    Address: 00:00:0c:00:02:00 (00:00:0c:00:02:00)
      .... 0 .... = IG bit: Individual address (unicast)
      .... 0. .... = LG bit: Globally unique address (factory default)
  Type: IP (0x0800)
  Frame check sequence: 0x48987da0 [incorrect, should be 0x77ee4fab]
Internet Protocol, Src: 10.10.10.20 (10.10.10.20), Dst: 10.10.10.10 (10.10.10.10)
  Version: 4
  Header length: 20 bytes
  Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    0000 00.. = Differentiated Services Codepoint: Default (0x00)
    .... 0. = ECN-Capable Transport (ECT): 0
    .... 0 = ECN-CE: 0...
Output truncated
!
!
!

vss_dut1# monitor capture mycap export bootflash:mycap_exp.pcap
...
vss_dut1# show monitor capture mycap buffer

```

Step 3 Once the capture session is over, delete the capture point if it is no longer needed.

```

vss_dut1# monitor capture mycap stop
*Nov 15 01:08:58.627 PDT: %BUFCAP-6-DISABLE: Capture Point mycap disabled
vss_dut1# no monitor capture mycap

```




Configuring Enhanced Object Tracking

This chapter describes how to configure enhanced object tracking on the Catalyst 4500 series switch. This feature provides a more complete alternative to the Hot Standby Routing Protocol (HSRP) tracking mechanism, which allows you to track the line-protocol state of an interface. If the line protocol state of an interface goes down, the HSRP priority of the interface is reduced and another HSRP device with a higher priority becomes active. The enhanced object tracking feature separates the tracking mechanism from HSRP and creates a separate, standalone tracking process that can be used by processes other than HSRP. This allows tracking other objects in addition to the interface line-protocol state. A client process, such as HSRP, can register an interest in tracking objects and request notification when the tracked object changes state. This feature increases the availability and speed of recovery of a routing system and decreases outages and outage duration.

Unless otherwise noted, the term *switch* refers to a Catalyst 4500 series switch.

The chapter includes these sections:

- [Understanding Enhanced Object Tracking, page 72-1](#)
- [Configuring Enhanced Object Tracking Features, page 72-2](#)
- [Monitoring Enhanced Object Tracking, page 72-12](#)



Note

For more details on enhanced object tracking, refer to this URL:

http://www.cisco.com/en/US/docs/ios/ios_xe/ipapp/configuration/guide/ipapp_eot_xe.html

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

Understanding Enhanced Object Tracking

Each tracked object has a unique number that is specified in the tracking command-line interface (CLI). Client processes use this number to track a specific object. The tracking process periodically polls the tracked object for value changes and sends any changes (as up or down values) to interested client processes, either immediately or after a specified delay. Several clients can track the same object, and can take different actions when the object changes state.

You can also track a combination of objects in a list by using either a weight threshold or a percentage threshold to measure the state of the list. You can combine objects using Boolean logic. A tracked list with a Boolean “AND” function requires that each object in the list be in an up state for the tracked object to be up. A tracked list with a Boolean “OR” function needs only one object in the list to be in the up state for the tracked object to be up.

Configuring Enhanced Object Tracking Features

- [Default Configuration, page 72-2](#)
- [Tracking Interface Line-Protocol or IP Routing State, page 72-2](#)
- [Configuring a Tracked List, page 72-3](#)
- [Configuring HSRP Object Tracking, page 72-7](#)
- [Configuring Other Tracking Characteristics, page 72-8](#)
- [Configuring IP SLAs Object Tracking, page 72-8](#)
- [Configuring Static Routing Support, page 72-10](#)

Default Configuration

No type of object tracking is configured.

Tracking Interface Line-Protocol or IP Routing State

You can track either the interface line protocol state or the interface IP routing state. When you track the IP routing state, these three conditions are required for the object to be up:

- IP routing must be enabled and active on the interface.
- The interface line-protocol state must be up.
- The interface IP address must be known.

If all three of these conditions are not met, the IP routing state is down.

To track the line-protocol state or IP routing state of an interface, perform the following task:

	Command	Purpose
Step 1	configure terminal	Enters global configuration mode.
Step 2	track <i>object-number</i> interface <i>interface-id</i> line-protocol	(Optional) Creates a tracking list to track the line-protocol state of an interface and enter tracking configuration mode. <ul style="list-style-type: none"> • The <i>object-number</i> identifies the tracked object and can be from 1 to 500. • The interface <i>interface-id</i> is the interface being tracked.
Step 3	delay { up <i>seconds</i> [down <i>seconds</i>] [up <i>seconds</i>] down <i>seconds</i> }	(Optional) Specifies a period of time in seconds to delay communicating state changes of a tracked object. The range is from 1 to 180 seconds.
Step 4	exit	Returns to global configuration mode.

	Command	Purpose
Step 5	track <i>object-number</i> interface <i>interface-id</i> ip routing	(Optional) Creates a tracking list to track the IP routing state of an interface, and enter tracking configuration mode. IP-route tracking tracks an IP route in the routing table and the ability of an interface to route IP packets. <ul style="list-style-type: none"> The <i>object-number</i> identifies the tracked object and can be from 1 to 500. The interface <i>interface-id</i> is the interface being tracked.
Step 6	delay { up <i>seconds</i> [down <i>seconds</i>] [up <i>seconds</i>] down <i>seconds</i> }	(Optional) Specifies a period of time in seconds to delay communicating state changes of a tracked object. The range is from 1 to 180 seconds.
Step 7	end	Returns to privileged EXEC mode.
Step 8	show track <i>object-number</i>	Verifies that the specified objects are being tracked.
Step 9	copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example configures the tracking of an interface line-protocol state and verifies the configuration:

```
Switch(config)# track 33 interface gigabitethernet 1/0/1 line-protocol
Switch(config-track)# end
Switch# show track 33
Track 33
  Interface GigabitEthernet1/0/1 line-protocol
  Line protocol is Down (hw down)
    1 change, last change 00:18:28
```

Configuring a Tracked List

You can configure a tracked list of objects with a Boolean expression, a weight threshold, or a percentage threshold. A tracked list contains one or more objects. An object must exist before it can be added to the tracked list.

- You configure a Boolean expression to specify calculation by using either “AND” or “OR” operators.
- When you measure the tracked list state by a weight threshold, you assign a weight number to each object in the tracked list. The state of the tracked list is determined by whether or not the threshold was met. The state of each object is determined by comparing the total weight of all objects against a threshold weight for each object.
- When you measure the tracked list by a percentage threshold, you assign a percentage threshold to all objects in the tracked list. The state of each object is determined by comparing the assigned percentages of each object to the list.

Configuring a Tracked List with a Boolean Expression

Configuring a tracked list with a Boolean expression enables calculation by using either “AND” or “OR” operators. For example, when tracking two interfaces using the “AND” operator, *up* means that both interfaces are up, and *down* means that either interface is down.

To configure a tracked list of objects with a Boolean expression, perform this task:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>track track-number list boolean {and or}</code>	Configures a tracked list object, and enter tracking configuration mode. The <i>track-number</i> can be from 1 to 500. <ul style="list-style-type: none"> boolean—Specify the state of the tracked list based on a Boolean calculation. and—Specify that the list is up if all objects are up or down if one or more objects are down. or—Specify that the list is up if one object is up or down if all objects are down.
Step 3	<code>object object-number [not]</code>	Specifies the object to be tracked. The range is from 1 to 500. The keyword not negates the state of the object, which means that when the object is up, the tracked list detects the object as down. Note An object must exist before you can add it to a tracked list.
Step 4	<code>delay {up seconds [down seconds] [up seconds] down seconds}</code>	(Optional) Specifies a period of time in seconds to delay communicating state changes of a tracked object. The range is from 1 to 180 seconds.
Step 5	<code>end</code>	Returns to privileged EXEC mode.
Step 6	<code>show track object-number</code>	Verifies that the specified objects are being tracked.
Step 7	<code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

Use the **no track track-number** global configuration command to delete the tracked list.

This example configures track list 4 with a Boolean AND expression that contains two objects with one object state negated. If the list is up, the list detects that object 2 is down:

```
Switch(config)# track 4 list boolean and
Switch(config-track)# object 1
Switch(config-track)# object 2 not
Switch(config-track)# exit
```

Configuring a Tracked List with a Weight Threshold

To track by weight threshold, configure a tracked list of objects, specify that weight is used as the threshold, and configure a weight for each of its objects. The state of each object is determined by comparing the total weight of all objects that are up against a threshold weight for each object.

You cannot use the Boolean “NOT” operator in a weight threshold list.

Beginning in privileged EXEC mode, follow these steps to configure a tracked list of objects by using a weight threshold and to configure a weight for each object:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>track track-number list threshold weight</code>	Configures a tracked list object and enter tracking configuration mode. The <i>track-number</i> can be from 1 to 500. <ul style="list-style-type: none"> threshold—Specify the state of the tracked list based on a threshold. weight—Specify that the threshold is based on weight.
Step 3	<code>object object-number [weight weight-number]</code>	Specifies the object to be tracked. The range is from 1 to 500. The optional weight weight-number specifies a threshold weight for the object. The range is from 1 to 255. Note An object must exist before you can add it to a tracked list.
Step 4	<code>threshold weight {up number / [down number]}</code>	Specifies the threshold weight. <ul style="list-style-type: none"> up number—The valid range is from 1 to 255. down number—(Optional) The range depends on the number selected for the up number. If you configure the up number as 25, the range shown for the down number is 0 to 24.
Step 5	<code>delay {up seconds [down seconds] [up seconds] down seconds}</code>	(Optional) Specifies a period of time in seconds to delay communicating state changes of a tracked object. The range is from 1 to 180 seconds.
Step 6	<code>end</code>	Returns to privileged EXEC mode.
Step 7	<code>show track object-number</code>	Verifies that the specified objects are being tracked.
Step 8	<code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

Use the **no track track-number** global configuration command to delete the tracked list.

The example configures track list 4 to track by weight threshold. If object 1 and object 2 are down, then track list 4 is up because object 3 satisfies the up threshold value of up 30. But if object 3 is down, both objects 1 and 2 must be up in order to satisfy the threshold weight.

```
Switch(config)# track 4 list threshold weight
Switch(config-track)# object 1 weight 15
Switch(config-track)# object 2 weight 20
Switch(config-track)# object 3 weight 30
Switch(config-track)# threshold weight up 30 down 10
Switch(config-track)# exit
```

This configuration can be useful if object 1 and object 2 represent two small bandwidth connections and object 3 represents one large bandwidth connection. The configured **down 10** value means that once the tracked object is up, it will not go down until the threshold value is equal to or lower than 10, which in this example means that all connections are down.

Configuring a Tracked List with a Percentage Threshold

To track by percentage threshold, configure a tracked list of objects, specify that a percentage will be used as the threshold, and specify a percentage for all objects in the list. The state of the list is determined by comparing the assigned percentage of each object to the list.

You cannot use the Boolean “NOT” operator in a percentage threshold list.

To configure a tracked list of objects by using a percentage threshold, perform this task:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>track track-number list threshold percentage</code>	Configures a tracked list object and enter tracking configuration mode. The <i>track-number</i> can be from 1 to 500. <ul style="list-style-type: none"> threshold—Specify the state of the tracked list based on a threshold. percentage—Specify that the threshold is based on percentage.
Step 3	<code>object object-number</code>	Specifies the object to be tracked. The range is from 1 to 500. Note An object must exist before you can add it to a tracked list.
Step 4	<code>threshold percentage {up number / [down number]}</code>	Specifies the threshold percentage. <ul style="list-style-type: none"> up number—The valid range is from 1 to 100. down number—(Optional) The range depends on the number selected for the up number. If you configure the up number as 25, the range shown for the down number is 0 to 24.
Step 5	<code>delay {up seconds [down seconds] [up seconds] down seconds}</code>	(Optional) Specifies a period of time in seconds to delay communicating state changes of a tracked object. The range is from 1 to 180 seconds.
Step 6	<code>end</code>	Returns to privileged EXEC mode.
Step 7	<code>show track object-number</code>	Verifies that the specified objects are being tracked.
Step 8	<code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

Use the **no track track-number** global configuration command to delete the tracked list.

This example configures tracked list 4 with three objects and a specified percentages to measure the state of the list:

```
Switch(config)# track 4 list threshold percentage
Switch(config-track)# object 1
Switch(config-track)# object 2
Switch(config-track)# object 3
Switch(config-track)# threshold percentage up 51 down 10
Switch(config-track)# exit
```

Configuring HSRP Object Tracking

To configure a standby HSRP group to track an object and change the HSRP priority based on the object state, perform this task:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<pre>track object-number {interface interface-id {line-protocol ip routing} ip route ip-address/prefix-length {metric threshold reachability} list {boolean {and or}} {threshold {weight percentage}}}</pre>	<p>(Optional) Creates a tracking list to track the configured state and enter tracking configuration mode.</p> <ul style="list-style-type: none"> The <i>object-number</i> range is from 1 to 500. Enter interface <i>interface-id</i> to select an interface to track. Enter line-protocol to track the interface line protocol state or enter ip routing to track the interface IP routing state. Enter ip route <i>ip-address/prefix-length</i> to track the state of an IP route. Enter metric threshold to track the threshold metric or enter reachability to track if the route is reachable. <p>The default up threshold is 254 and the default down threshold is 255.</p> <ul style="list-style-type: none"> Enter list to track objects grouped in a list. Configure the list as described on the previous pages. <ul style="list-style-type: none"> For boolean, see the “Configuring a Tracked List with a Boolean Expression” section on page 72-4. For threshold weight, see the “Configuring a Tracked List with a Weight Threshold” section on page 72-5. For threshold percentage, see the “Configuring a Tracked List with a Percentage Threshold” section on page 72-6. <p>Note Repeat this step for each interface to be tracked.</p>
Step 3	<code>exit</code>	Returns to global configuration mode.
Step 4	<code>interface interface-id</code>	Enters interface configuration mode.
Step 5	<code>standby [group-number] ip [ip-address [secondary]]</code>	<p>Creates (or enable) the HSRP group by using its number and virtual IP address.</p> <ul style="list-style-type: none"> (Optional) <i>group-number</i>—Enter a group number on the interface for which HSRP is being enabled. The range is 0 to 255; the default is 0. If there is only one HSRP group, you do not need to enter a group number. (Optional on all but one interface) <i>ip-address</i>—Specify the virtual IP address of the hot standby router interface. You must enter the virtual IP address for at least one of the interfaces; it can be learned on the other interfaces. (Optional) secondary—Specify that the IP address is a secondary hot standby router interface. If this keyword is omitted, the configured address is the primary IP address.

	Command	Purpose
Step 6	standby [<i>group-number</i>] track <i>object-number</i> [decrement <i>priority-decrement</i>]	Configures HSRP to track an object and change the hot standby priority based on the state of the object. <ul style="list-style-type: none"> (Optional) <i>group-number</i>—Enter the group number to which the tracking applies. <i>object-number</i>—Enter a number representing the object to be tracked. The range is from 1 to 500; the default is 1. (Optional) decrement <i>priority-decrement</i>—Specify the amount by which the hot standby priority for the router is decremented (or incremented) when the tracked object goes down (or comes back up). The range is from 1 to 255; the default is 10.
Step 7	end	Returns to privileged EXEC mode.
Step 8	show standby	Verifies the standby router IP address and tracking states.
Step 9	copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuring Other Tracking Characteristics

You can also use the enhanced object tracking for tracking other characteristics.

- You can track the reachability of an IP route by using the **track ip route reachability** global configuration command.
- You can use the **track ip route metric threshold** global configuration command to determine if a route is above or below threshold.
- You can use the **track resolution** global configuration command to change the metric resolution default values for routing protocols.
- You can use the **track timer** tracking configuration command to configure the tracking process to periodically poll tracked objects.

Use the **show track** privileged EXEC command to verify enhanced object tracking configuration.

Configuring IP SLAs Object Tracking

Cisco IOS IP Service Level Agreements (IP SLAs) is a network performance measurement and diagnostics tool that uses active monitoring by generating traffic to measure network performance. Cisco IP SLAs operations collects real-time metrics that you can use for network troubleshooting, design, and analysis.

For more information about Cisco IP SLAs on the switch, see [Chapter 80, “Configuring Cisco IOS IP SLA Operations.”](#) For IP SLAs command information see the *Cisco IOS IP SLAs Command Reference, Release 12.4T*.

Object tracking of IP SLAs operations allows clients to track the output from IP SLAs objects and use this information to trigger an action. Every IP SLAs operation maintains an SNMP operation return-code value, such as *OK* or *OverThreshold*, that can be interpreted by the tracking process. You can track two

aspects of IP SLAs operation: state and reachability. For state, if the return code is OK, the track state is up; if the return code is not OK, the track state is down. For reachability, if the return code is OK or OverThreshold, reachability is up; if not OK, reachability is down.

To track the state of an IP SLAs operation or the reachability of an IP SLAs IP host, perform this task:

	Command	Purpose
Step 1	configure terminal	Enters global configuration mode.
Step 2	track object-number ip sla operation-number state	Enters tracking configuration mode to track the state of an IP SLAs operation. <ul style="list-style-type: none"> The <i>object-number</i> range is from 1 to 500. The <i>operation-number</i> range is from 1 to 2147483647.
Step 3	delay {up seconds [down seconds] [up seconds] down seconds}	(Optional) Specifies a period of time in seconds to delay communicating state changes of a tracked object. The range is from 1 to 180 seconds.
Step 4	exit	Returns to global configuration mode.
Step 5	track object-number ip sla operation-number reachability	Enters tracking configuration mode to track the reachability of an IP SLAs IP host. <ul style="list-style-type: none"> The <i>object-number</i> range is from 1 to 500. The <i>operation-number</i> range is from 1 to 2147483647.
Step 6	delay {up seconds [down seconds] [up seconds] down seconds}	(Optional) Specifies a period of time in seconds to delay communicating state changes of a tracked object. The range is from 1 to 180 seconds.
Step 7	end	Returns to privileged EXEC mode.
Step 8	show track object-number	Displays tracking information to verify the configuration.
Step 9	copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to configure and display IP SLAs state tracking:

```
Switch(config)# track 2 200 state
Switch(config)# end
Switch# show track 2
Track 2
  Response Time Reporter 1 state
  State is Down
    1 change, last change 00:00:47
  Latest operation return code: over threshold
  Latest RTT (milliseconds) 4
  Tracked by:
    HSRP Ethernet0/1 3
```

This example output shows whether a route is reachable:

```
Switch(config)# track 3 500 reachability
Switch(config)# end
Switch# show track 3
Track 3
  Response Time Reporter 1 reachability
  Reachability is Up
    1 change, last change 00:00:47
  Latest operation return code: over threshold
  Latest RTT (milliseconds) 4
  Tracked by:
    HSRP Ethernet0/1 3
```

Configuring Static Routing Support

Static routing support using enhanced object tracking provides the ability for the switch to use ICMP pings to identify when a preconfigured static route or a DHCP route goes down. When tracking is enabled, the system tracks the state of the route and informs the client when that state changes. Static route object tracking uses Cisco IP SLAs to generate ICMP pings to monitor the state of the connection to the primary gateway.

For more information about Cisco IP SLAs support on the switch, see [Chapter 80, “Configuring Cisco IOS IP SLA Operations.”](#)

- For more information about static route object tracking, see:
http://www.cisco.com/en/US/docs/ios/dial/configuration/guide/dia_rel_stc_rtg_bckup_support_TS_D_Island_of_Content_Chapter.html

You use this process to configure static route object tracking:

-
- | | |
|---------------|--|
| Step 1 | Configure a primary interface for static routing or for DHCP.ex |
| Step 2 | Configure an IP SLAs agent to ping an IP address using a primary interface and a track object to monitor the state of the agent. |
| Step 3 | Configure a default static default route using a secondary interface. This route is used only if the primary route is removed. |
-

Configuring a Primary Interface

To configure a primary interface for static routing, perform this task:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>interface interface-id</code>	Selects a primary or secondary interface and enter interface configuration mode.
Step 3	<code>description string</code>	Adds a description to the interface.
Step 4	<code>ip address ip-address mask [secondary]</code>	Sets the primary or secondary IP address for the interface.
Step 5	<code>exit</code>	Returns to global configuration mode.

To configure a primary interface for DHCP, perform this task:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>interface interface-id</code>	Selects a primary or secondary interface and enter interface configuration mode.
Step 3	<code>description string</code>	Adds a description to the interface.
Step 4	<code>ip dhcp client route track number</code>	Configures the DCHP client to associate any added routes with the specified track number. Valid numbers are from 1 to 500.

	Command	Purpose
Step 5	<code>ip address dhcp</code>	Acquires an IP address on an Ethernet interface from DHCP.
Step 6	<code>exit</code>	Returns to global configuration mode.

Configuring a Cisco IP SLAs Monitoring Agent and Track Object

To configure network monitoring with Cisco IP SLAs, perform this task:

Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>ip sla operation-number</code>	Begins configuring a Cisco IP SLAs operation and enter IP SLA configuration mode.
Step 3	<code>icmp-echo {destination-ip-address / destination hostname [source- ipaddr {ip-address / hostname source-interface interface-id]}</code>	Configures a Cisco IP SLAs end-to-end ICMP echo response time operation and enter IP SLAs ICMP echo configuration mode.
Step 4	<code>timeout milliseconds</code>	Sets the amount of time for which the operation waits for a response from its request packet.
Step 5	<code>frequency seconds</code>	Sets the rate at which the operation is sent into the network.
Step 6	<code>threshold milliseconds</code>	Sets the rising threshold (hysteresis) that generates a reaction event and stores history information for the operation.
Step 7	<code>exit</code>	Exits IP SLAs ICMP echo configuration mode.
Step 8	<code>ip sla schedule operation-number [life {forever seconds}] start-time time pending now after time] [ageout seconds] [recurring]</code>	Configures the scheduling parameters for a single IP SLAs operation.
Step 9	<code>track object-number ip sla operation-number {state reachability}</code>	Tracks the state of a Cisco IOS IP SLAs operation and enter tracking configuration mode.
Step 10	<code>end</code>	Returns to privileged EXEC mode.
Step 11	<code>show track object-number</code>	Displays tracking information to verify the configuration.
Step 12	<code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

Configuring a Routing Policy and Default Route

- To configure a routing policy for backup static routing by using object tracking, perform the following task. For more details about the commands in the procedure, see: http://www.cisco.com/en/US/docs/ios/dial/configuration/guide/dia_rel_stc_rtg_bckup_support_TS_D_Island_of_Content_Chapter.html

Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>access-list access-list-number</code>	Defines an extended IP access list. Configure any optional characteristics.
Step 3	<code>route-map map-tag [permit deny] [sequence-number]</code>	Enters route-map configuration mode and define conditions for redistributing routes from one routing protocol to another.

Step 4	<code>match ip address {access-list number / access-list name}</code>	Distributes any routes that have a destination network number address that is permitted by a standard or extended access list or performs policy routing on packets. You can enter multiple numbers or names.
Step 5	<code>set ip next-hop dynamic dhcp</code>	For DHCP networks only. Sets the next hop to the gateway that was most recently learned by the DHCP client.
Step 6	<code>set interface interface-id</code>	For static routing networks only. Indicates where to send output packets that pass a match clause of a route map for policy routing.
Step 7	<code>exit</code>	Exits route-map configuration mode.
Step 8	<code>ip local policy route-map map-tag</code>	Identifies a route map to use for local policy routing.
Step 9	<code>ip route prefix mask {ip-address / interface-id [ip-address]} [distance] [name] [permanent track track-number] [tag tag]</code>	For static routing networks only. Establishes static routes. Entering track track-number specifies that the static route is installed only if the configured track object is up.
Step 10	<code>end</code>	Returns to privileged EXEC mode.
Step 11	<code>show ip route track table</code>	Displays information about the IP route track table.
Step 12	<code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

For configuration examples, see:

http://www.cisco.com/en/US/tech/tk801/tk133/technologies_configuration_example09186a0080093f7e.shtml

Monitoring Enhanced Object Tracking

Use the privileged EXEC or user EXEC commands in [Table 72-1](#) to display enhanced object tracking information.

Table 72-1 Commands for Displaying Tracking Information

Command	Purpose
<code>show ip route track table</code>	Displays information about the IP route track table.
<code>show track [object-number]</code>	Displays information about the all tracking lists or the specified list.
<code>show track brief</code>	Displays a single line of tracking information output.
<code>show track interface [brief]</code>	Displays information about tracked interface objects.
<code>show track ip [object-number] [brief] route</code>	Displays information about tracked IP-route objects.
<code>show track resolution</code>	Displays the resolution of tracked parameters.
<code>show track timers</code>	Displays tracked polling interval timers.



Configuring System Message Logging

This chapter describes how to configure system message logging on the Catalyst 4500 series switch.

This chapter consists of these sections:

- [About System Message Logging, page 73-1](#)
- [Configuring System Message Logging, page 73-2](#)
- [Displaying the Logging Configuration, page 73-12](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About System Message Logging

By default, a switch sends the output from system messages and **debug** privileged EXEC commands to a logging process. The logging process controls the distribution of logging messages to various destinations, such as the logging buffer, terminal lines, or a UNIX syslog server, depending on your configuration. The process also sends messages to the console.



Note

The syslog format is compatible with 4.3 BSD UNIX.

When the logging process is disabled, messages are sent only to the console. The messages are sent as they are generated, so message and debug output are interspersed with prompts or output from other commands. Messages are displayed on the console after the process that generated them has finished.

You can set the severity level of the messages to control the type of messages displayed on the consoles and each of the destinations. You can time-stamp log messages or set the syslog source address to enhance real-time debugging and management. For information on possible messages, see the system message guide for this release.

You can access logged system messages by using the switch command-line interface (CLI) or by saving them to a properly configured syslog server. The switch software saves syslog messages in an internal buffer on the switch. If the switch fails, the log is lost unless you had saved it to flash memory.

You can remotely monitor system messages by viewing the logs on a syslog server or by accessing the switch through Telnet or using the console port.

Configuring System Message Logging

These sections describe how to configure system message logging:

- [System Log Message Format, page 73-2](#)
- [Default System Message Logging Configuration, page 73-3](#)
- [Disabling Message Logging, page 73-3](#)
- [Setting the Message Display Destination Device, page 73-4](#)
- [Synchronizing Log Messages, page 73-5](#)
- [Enabling and Disabling Timestamps on Log Messages, page 73-6](#)
- [Enabling and Disabling Sequence Numbers in Log Messages \(Optional\), page 73-7](#)
- [Defining the Message Severity Level \(Optional\), page 73-8](#)
- [Limiting Syslog Messages Sent to the History Table and to SNMP \(Optional\), page 73-9](#)
- [Configuring UNIX Syslog Servers, page 73-10](#)

System Log Message Format

System log messages can contain up to 80 characters and a percent sign (%), which follows the optional sequence number or time-stamp information, if configured. Messages are displayed in this format:

seq no:timestamp: %facility-severity-MNEMONIC:description

The part of the message preceding the percent sign depends on the setting of the **service sequence-numbers**, **service timestamps log datetime**, **service timestamps log datetime [localtime] [msec] [show-timezone]** command or the **service timestamps log uptime** global configuration command.

[Table 73-1](#) describes the elements of syslog messages.

Table 73-1 **System Log Message Elements**

Element	Description
<i>seq no:</i>	Stamps log messages with a sequence number only if the service sequence-numbers global configuration command is configured. For more information, see the “ Enabling and Disabling Sequence Numbers in Log Messages (Optional) ” section on page 73-7 .
<i>timestamp</i> formats: <i>mm/dd hh:mm:ss</i> or <i>hh:mm:ss</i> (short uptime) or <i>d h</i> (long uptime)	Date and time of the message or event. This information appears only if the service timestamps log [datetime log] global configuration command is configured. For more information, see the “ Enabling and Disabling Timestamps on Log Messages ” section on page 73-6 .

Table 73-1 System Log Message Elements (continued)

Element	Description
<i>facility</i>	The facility to which the message refers (for example, SNMP, SYS, and so forth). For a list of supported facilities, see Table 73-4 on page 73-12 .
<i>severity</i>	Single-digit code from 0 to 7 that is the severity of the message. For a description of the severity levels, see Table 73-3 on page 73-8 .
<i>MNEMONIC</i>	Text string that uniquely describes the message.
<i>description</i>	Text string containing detailed information about the event being reported.

This example shows a partial switch system message:

```
00:00:46: %LINK-3-UPDOWN: Interface Port-channel1, changed state to up
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet1/0/1, changed state to up
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet1/0/2, changed state to up
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to down
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0/1, changed
state to down 2
*Mar  1 18:46:11: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
18:47:02: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
*Mar  1 18:48:50.483 UTC: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
```

Default System Message Logging Configuration

[Table 73-2](#) shows the default system message logging configuration.

Table 73-2 Default System Message Logging Configuration

Feature	Default Setting
System message logging to the console	Enabled.
Console severity	Debugging (and numerically lower levels; see Table 73-3 on page 73-8).
Logging file configuration	No filename specified.
Logging buffer size	4096 bytes.
Logging history size	1 message.
Timestamps	Disabled.
Synchronous logging	Disabled.
Logging server	Disabled.
Syslog server IP address	None configured.
Server facility	Local7 (see Table 73-4 on page 73-12).
Server severity	Informational (and numerically lower levels; see Table 73-3 on page 73-8).

Disabling Message Logging

Message logging is enabled by default. It must be enabled to send messages to any destination other than the console. When enabled, log messages are sent to a logging process, which logs messages to designated locations asynchronously to the processes that generated the messages.

To disable message logging, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# no logging on	Disables message logging.
Step 3	Switch(config)# end	Returns to privileged EXEC mode.
Step 4	Switch# show running-config or show logging	Verifies your entries.
Step 5	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Disabling the logging process can slow down the switch because a process must wait until the messages are written to the console before continuing. When the logging process is disabled, messages are displayed on the console as soon as they are produced, often appearing in the middle of command output.

The **logging synchronous** global configuration command also affects the display of messages to the console. When this command is enabled, messages appear only after you press Return. For more information, see the [“Synchronizing Log Messages”](#) section on page 73-5.

To reenable message logging after it has been disabled, use the **logging on** global configuration command.

Setting the Message Display Destination Device

If message logging is enabled, you can send messages to specific locations in addition to the console.

To specify the locations that receive messages, perform this task, beginning in privileged EXEC mode:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# logging buffered [size]	<p>Logs messages to an internal buffer on the switch. The default buffer size is 4096. The range is 4096 to 2147483647 bytes.</p> <p>If the switch, the log file is lost unless you previously saved it to flash memory. See Step 4.</p> <p>Note Do not make the buffer size too large because the switch could run out of memory for other tasks. Use the show memory privileged EXEC command to view the free processor memory on the switch. However, this value is the maximum available, and the buffer size should <i>not</i> be set to this amount.</p>

	Command	Purpose
Step 3	Switch(config)# logging <i>host</i>	Logs messages to a UNIX syslog server host. For <i>host</i> , specify the name or IP address of the host to be used as the syslog server. To build a list of syslog servers that receive logging messages, enter this command more than once. For complete syslog server configuration steps, see the “Configuring UNIX Syslog Servers” section on page 73-10.
Step 4	Switch(config)# end	Returns to privileged EXEC mode.
Step 5	Switch# terminal monitor	Logs messages to a nonconsole terminal during the current session. Terminal parameter-setting commands are set locally and do not remain in effect after the session has ended. You must perform this step for each session to see the debugging messages.
Step 6	Switch# show running-config	Verifies your entries.
Step 7	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

The **logging buffered** global configuration command copies logging messages to an internal buffer. The buffer is circular, so newer messages overwrite older messages after the buffer is full. To display the messages that are logged in the buffer, use the **show logging** privileged EXEC command. The first message displayed is the oldest message in the buffer. To clear the contents of the buffer, use the **clear logging** privileged EXEC command.

To disable logging to the console, use the **no logging console** global configuration command. To disable logging to a file, use the **no logging file** [*severity-level-number* | *type*] global configuration command.

Synchronizing Log Messages

You can synchronize unsolicited messages and **debug** privileged EXEC command output with solicited device output and prompts for a specific console port line or virtual terminal line. You can identify the types of messages to be output asynchronously based on the level of severity. You can also configure the maximum number of buffers for storing asynchronous messages for the terminal after which messages are dropped.

When synchronous logging of unsolicited messages and **debug** command output is enabled, unsolicited device output appears on the console or printed after solicited device output appears or printed. Unsolicited messages and **debug** command output appears on the console after the prompt for user input is returned. Unsolicited messages and **debug** command output are not interspersed with solicited device output and prompts. After the unsolicited messages are displayed, the console again displays the user prompt.

To configure synchronous logging, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# line [console vty] <i>line-number</i> [<i>ending-line-number</i>]	Specifies the line to be configured for synchronous logging of messages. <ul style="list-style-type: none"> Use the console keyword for configurations that occur using the switch console port. Use the line vty line-number command to specify which vty lines are to have synchronous logging enabled. You use a vty connection for configurations that occur through a Telnet session. The range of line numbers is from 0 to 15. <p>You can change the setting of all 16 vty lines at once by entering: line vty 0 15</p> <p>Or you can change the setting of the single vty line being used for your current connection. For example, to change the setting for vty line 2, enter: line vty 2</p> <p>When you enter this command, the mode changes to line configuration.</p>
Step 3	Switch(config)# logging synchronous [level [<i>severity-level</i> all] limit <i>number-of-buffers</i>]	Enables synchronous logging of messages. <ul style="list-style-type: none"> (Optional) For level severity-level, specify the message severity level. Messages with a severity level equal to or higher than this value are printed asynchronously. Low numbers mean greater severity and high numbers mean lesser severity. The default is 2. (Optional) Specifying level all means that all messages are printed asynchronously regardless of the severity level. (Optional) For limit number-of-buffers, specify the number of buffers to be queued for the terminal after which new messages are dropped. The range is 0 to 2147483647. The default is 20.
Step 4	Switch(config)# end	Returns to privileged EXEC mode.
Step 5	Switch# show running-config	Verifies your entries.
Step 6	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To disable synchronization of unsolicited messages and debug output, use the **no logging synchronous** [**level severity-level** | **all**] [**limit number-of-buffers**] line configuration command.

Enabling and Disabling Timestamps on Log Messages



Note

By default, log messages are not time-stamped.

To enable time-stamping of log messages, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# service timestamps log uptime or Switch(config)# service timestamps log datetime [msec] [localtime] [show-timezone]	Enables log time-stamps. The first command enables time-stamps on log messages, showing the time since the system was rebooted. The second command enables time-stamps on log messages. Depending on the options selected, the timestamp can include the date, time in milliseconds relative to the local time zone, and the time zone name.
Step 3	Switch(config)# end	Returns to privileged EXEC mode.
Step 4	Switch# show running-config	Verifies your entries.
Step 5	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To disable time-stamps for both debug and log messages, use the **no service timestamps** global configuration command.

This example shows part of a logging display with the **service timestamps log datetime** global configuration command enabled:

```
*Mar 1 18:46:11: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
```

This example shows part of a logging display with the **service timestamps log uptime** global configuration command enabled:

```
00:00:46: %LINK-3-UPDOWN: Interface Port-channel1, changed state to up
```

Enabling and Disabling Sequence Numbers in Log Messages (Optional)

Because more than one log message can have the same timestamp, you can display messages with sequence numbers so that you can unambiguously refer to a single message. By default, sequence numbers in log messages are not displayed.

To enable sequence numbers in log messages, perform this task, which is optional.

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# service sequence-numbers	Enables sequence numbers.
Step 3	Switch(config)# end	Returns to privileged EXEC mode.
Step 4	Switch# show running-config	Verifies your entries.
Step 5	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To disable sequence numbers, use the **no service sequence-numbers** global configuration command.

This example shows part of a logging display with sequence numbers enabled:

```
000019: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
```

Defining the Message Severity Level (Optional)

You can limit messages displayed to the selected device by specifying the severity level of the message, which are described in [Table 73-3](#).

To define the message severity level, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# logging console level	Limits messages logged to the console. By default, the console receives debugging messages and numerically lower levels (see Table 73-3 on page 73-8).
Step 3	Switch(config)# logging monitor level	Limits messages logged to the terminal lines. By default, the terminal receives debugging messages and numerically lower levels (see Table 73-3 on page 73-8).
Step 4	Switch(config)# logging trap level	Limits messages logged to the syslog servers. By default, syslog servers receive informational messages and numerically lower levels (see Table 73-3 on page 73-8). For complete syslog server configuration steps, see the “Configuring UNIX Syslog Servers” section on page 73-10 .
Step 5	Switch(config)# end	Returns to privileged EXEC mode.
Step 6	Switch# show running-config or Switch# show logging	Verifies your entries.
Step 7	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.



Note

Specifying a *level* causes messages at that level and numerically lower levels to be displayed at the destination.

To disable logging to the console, use the **no logging console** global configuration command. To disable logging to a terminal other than the console, use the **no logging monitor** global configuration command. To disable logging to syslog servers, use the **no logging trap** global configuration command.

[Table 73-3](#) describes the *level* keywords. It also lists the corresponding UNIX syslog definitions from the most severe level to the least severe level.

Table 73-3 Message Logging Level Keywords

Level Keyword	Level	Description	Syslog Definition
emergencies	0	System unstable	LOG_EMERG
alerts	1	Immediate action needed	LOG_ALERT
critical	2	Critical conditions	LOG_CRIT
errors	3	Error conditions	LOG_ERR
warnings	4	Warning conditions	LOG_WARNING
notifications	5	Normal but significant condition	LOG_NOTICE

Table 73-3 Message Logging Level Keywords (continued)

Level Keyword	Level	Description	Syslog Definition
informational	6	Informational messages only	LOG_INFO
debugging	7	Debugging messages	LOG_DEBUG

The software generates four other categories of messages:

- Error messages about software or hardware malfunctions, displayed at levels **warnings** through **emergencies**. These types of messages mean that the functionality of the switch is affected. For information on how to recover from these malfunctions, see the system message guide for this release.
- Output from the **debug** commands, displayed at the **debugging** level. Debug commands are typically used only by the Technical Assistance Center.
- Interface up or down transitions and system restart messages, displayed at the **notifications** level. This message is only for information; switch functionality is not affected.
- Reload requests and low-process stack messages, displayed at the **informational** level. This message is only for information; switch functionality is not affected.

Limiting Syslog Messages Sent to the History Table and to SNMP (Optional)

If you enabled syslog message traps to be sent to an SNMP network management station by using the **snmp-server enable trap** global configuration command, you can change the level of messages sent and stored in the switch history table. You also can change the number of messages that are stored in the history table.

Messages are stored in the history table because SNMP traps are not guaranteed to reach their destination. By default, one message of the level **warning** and numerically lower levels (see [Table 73-3 on page 73-8](#)) are stored in the history table even if syslog traps are not enabled.

To change the level and history table size defaults, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# logging history level ¹	Changes the default level of syslog messages stored in the history file and sent to the SNMP server. See Table 73-3 on page 73-8 for a list of <i>level</i> keywords. By default, warnings, errors, critical, alerts, and emergencies messages are sent.
Step 3	Switch(config)# logging history size number	Specifies the number of syslog messages that can be stored in the history table. The default is to store one message. The range is 0 to 500 messages.
Step 4	Switch(config)# end	Returns to privileged EXEC mode.
Step 5	Switch# show running-config	Verifies your entries.
Step 6	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

1. [Table 73-3](#) lists the level keywords and severity level. For SNMP usage, the severity level values increase by 1. For example, emergencies equal 1, not 0, and critical equals 3, not 2.

When the history table is full (it contains the maximum number of message entries specified with the **logging history size** global configuration command), the oldest message entry is deleted from the table to allow the new message entry to be stored.

To return the logging of syslog messages to the default level, use the **no logging history** global configuration command. To return the number of messages in the history table to the default value, use the **no logging history size** global configuration command.

Configuring UNIX Syslog Servers

The next sections describe how to configure the UNIX server syslog daemon and how to define the UNIX system logging facility.

Logging Messages to a UNIX Syslog Daemon

Before you can send system log messages to a UNIX syslog server, you must configure the syslog daemon on a UNIX server. This procedure is optional.



Note

Some recent versions of UNIX syslog daemons no longer accept by default syslog packets from the network. If applies to your system, use the UNIX **man syslogd** command to decide what options must be added to or removed from the syslog command line to enable logging of remote syslog messages.

Log in as root, and perform these steps:

Step 1 Add a line such as the following to the file `/etc/syslog.conf`:

```
local7.debug /usr/adm/logs/cisco.log
```

The **local7** keyword specifies the logging facility to be used; see [Table 73-4 on page 73-12](#) for information on the facilities. The **debug** keyword specifies the syslog level; see [Table 73-3 on page 73-8](#) for information on the severity levels. The syslog daemon sends messages at this level or at a more severe level to the file specified in the next field. The file must already exist, and the syslog daemon must have permission to write to it.

Step 2 Create the log file by entering these commands at the UNIX shell prompt:

```
$ touch /var/log/cisco.log
$ chmod 666 /var/log/cisco.log
```

Step 3 Ensure that the syslog daemon reads the new changes:

```
$ kill -HUP `cat /etc/syslog.pid`
```

For more information, see the **man syslog.conf** and **man syslogd** commands on your UNIX system.

Configuring the UNIX System Logging Facility

When sending system log messages to an external device, you can cause the switch to identify its messages as originating from any of the UNIX syslog facilities.

To configure UNIX system facility message logging, perform this task (which is optional):

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# logging host	Logs messages to a UNIX syslog server host by entering its IP address. To build a list of syslog servers that receive logging messages, enter this command more than once.
Step 3	Switch(config)# logging trap level	Limits messages logged to the syslog servers. By default, syslog servers receive informational messages and lower. See Table 73-3 on page 73-8 for <i>level</i> keywords.
Step 4	Switch(config)# logging facility facility-type	Configures the syslog facility. See Table 73-4 on page 73-12 for <i>facility-type</i> keywords. The default is local7 .
Step 5	Switch(config)# end	Returns to privileged EXEC mode.
Step 6	Switch# show running-config	Verifies your entries.
Step 7	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To remove a syslog server, use the **no logging host** global configuration command, and specify the syslog server IP address. To disable logging to syslog servers, enter the **no logging trap** global configuration command.

Table 73-4 lists the UNIX system facilities supported by the software. For more information about these facilities, consult the operator's manual for your UNIX operating system.

Table 73-4 Logging Facility-Type Keywords

Facility Type Keyword	Description
auth	Authorization system
cron	Cron facility
daemon	System daemon
kern	Kernel
local0-7	Locally defined messages
lpr	Line printer system
mail	Mail system
news	USENET news
sys9-14	System use
syslog	System log
user	User process
uucp	UNIX-to-UNIX copy system

Displaying the Logging Configuration

To display the logging configuration and the contents of the log buffer, use the **show logging** privileged EXEC command. For information about the fields in this display, see the *Cisco IOS Configuration Fundamentals Command Reference, Release 12.3*.



Onboard Failure Logging (OBFL)

This chapter includes the following major sections:

- [Prerequisites for OBFL, page 74-1](#)
- [Restrictions for OBFL, page 74-2](#)
- [Information About OBFL, page 74-2](#)
- [Default Settings for OBFL, page 74-8](#)
- [Enabling OBFL, page 74-8](#)
- [Configuration Examples for OBFL, page 74-9](#)



Note

For more information about Onboard Failure Logging, see:

http://www.cisco.com/en/US/docs/ios/12_0s/feature/guide/12sobfl.html

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

Prerequisites for OBFL

None.

Restrictions for OBFL

They include:

- **Software Restrictions**—If a device (router or switch) intends to use *linear* flash memory as its OBFL storage media, Cisco IOS software must reserve a minimum of two physical sectors (or physical blocks) for the OBFL feature. Because an erase operation for a linear flash device is done on per-sector (or per-block) basis, one extra physical sector is needed. Otherwise, the minimum amount of space reserved for the OBFL feature on any device must be at least 8 KB.
- **Firmware Restrictions**—If a line card or port adapter runs an operating system or firmware that is different from the Cisco IOS operating system, the line card or port adapter must provide device driver level support or an interprocess communications (IPC) layer that allows the OBFL file system to communicate to the line card or port adapter. This requirement is enforced to allow OBFL data to be recorded on a storage device attached to the line card or port adapter.
- **Hardware Restrictions**—To support the OBFL feature, a device must have at least 8 KB of nonvolatile memory space reserved for OBFL data logging.

Information About OBFL

- [Overview of OBFL, page 74-2](#)
- [Information about Data Collected by OBFL, page 74-2](#)

Overview of OBFL

The Onboard Failure Logging (OBFL) feature collects data such as operating temperatures, hardware uptime, interrupts, and other important events and messages from system hardware installed in a Cisco router or switch. The data is stored in nonvolatile memory and helps technical personnel diagnose hardware problems.

Information about Data Collected by OBFL

- [OBFL Data Overview, page 74-2](#)
- [Temperature, page 74-3](#)
- [Operational Uptime, page 74-4](#)
- [Interrupts, page 74-6](#)
- [Message Logging, page 74-7](#)

OBFL Data Overview

The OBFL feature records operating temperatures, hardware uptime, interrupts, and other important events and messages that can assist with diagnosing problems with hardware cards (or *modules*) installed in a Cisco router or switch. Data is logged to files stored in nonvolatile memory. When the onboard hardware is started up, a first record is made for each area monitored and becomes a base value for subsequent records. The OBFL feature provides a circular updating scheme for collecting continuous records and archiving older (historical) records, ensuring accurate data about the system. Data is

recorded in one of two formats: continuous information that displays a snapshot of measurements and samples in a continuous file, and summary information that provides details about the data being collected. The data is displayed using the **show logging onboard** command. The message “No historical data to display” is seen when historical data is not available.

Temperature

Temperatures surrounding hardware modules can exceed recommended safe operating ranges and cause system problems such as packet drops. Higher than recommended operating temperatures can also accelerate component degradation and affect device reliability. Monitoring temperatures is important for maintaining environmental control and system reliability. Once a temperature sample is logged, the sample becomes the base value for the next record. From that point on, temperatures are recorded either when there are changes from the previous record or if the maximum storage time is exceeded. Temperatures are measured and recorded in degrees Celsius.

Temperature Example

```
Switch# sh logging onboard temperature
```

```
-----
TEMPERATURE SUMMARY INFORMATION
-----
```

```
Number of sensors      : 7
Sampling frequency    : 1 minutes
Maximum time of storage : 10 minutes
-----
```

Sensor	ID	Maximum Temperature 0C
Stub A	0	43
Stub B	1	37
XPP	2	51
VFE	3	61
NFE	4	50
CPU	5	55
FPGA	6	44

```
-----
Temp      Sensor ID
0C         1      2      3      4      5      6      7
-----
1         9y      9y      9y      9y      9y      9y      9y
15        0m      71h      0m      0m      0m      0m      0m
16        0m      183h     0m      0m      0m      0m      0m
-----
```

```
Temp      Sensor ID
0C         1      2      3      4      5      6      7
-----
17        0m      142m     0m      0m      0m      0m      0m
18        0m      190m     0m      0m      0m      0m      0m
19        0m      30m      0m      0m      0m      0m      0m
20        113h     0m      0m      0m      0m      0m      0m
21        37h      0m      0m      0m      0m      0m      101h
22        107h     0m      0m      0m      0m      0m      106h
23        110m     12m      0m      0m      0m      0m      47h
24        10m      122m     0m      0m      0m      0m      182m
25        0m      0m      0m      0m      0m      0m      120m
26        0m      56h      0m      0m      0m      0m      30m
27        0m      368h     0m      0m      0m      0m      0m
28        0m      8y       0m      0m      0m      0m      0m
29        134m     8y       0m      0m      139h     0m      0m
30        0m      682h     83h     0m      116h     0m      0m
31        90m      738h     31h     0m      200m     0m      95m
-----
```

```

32  209h 935h 138h 0m 120m 141h 258h
33  331h 934h 192m 0m 0m 113h 316h
34  579h 8y 190m 0m 0m 182m 432h
35  17y 149h 80m 0m 0m 150m 8y
36  914h 20m 0m 0m 0m 10m 8y
37  838h 270m 0m 140h 26m 0m 8y
38  47d 0m 0m 102h 108m 0m 790h
39  8y 0m 0m 948m 20m 0m 421h
-----
Temp                               Sensor ID
0C      1      2      3      4      5      6      7
-----
40  414h 0m 2m 100m 78h 0m 288h
41  74h 0m 113h 40m 340h 134m 113h
42  10m 0m 380h 0m 198h 0m 446m
43  270m 0m 8y 0m 373h 0m 0m
44  0m 0m 8y 0m 683h 45h 10m
45  0m 0m 8y 2m 17y 274h 0m
46  0m 0m 897h 105m 64d 257h 0m
47  0m 0m 785h 27m 8y 169h 0m
48  0m 0m 639h 4m 319h 666h 0m
49  0m 0m 379h 92h 786m 17y 0m
50  0m 0m 94h 330h 270m 61d 0m
51  0m 0m 106m 192h 0m 48d 0m
52  0m 0m 0m 190h 0m 8y 0m
53  0m 0m 0m 573h 0m 227h 0m
54  0m 0m 0m 736h 0m 180m 0m
55  0m 0m 0m 716h 0m 260m 0m
56  0m 0m 0m 902h 0m 0m 0m
57  0m 0m 0m 8y 0m 0m 0m
58  0m 0m 0m 8y 0m 0m 0m
59  0m 0m 0m 8y 0m 0m 0m
60  0m 0m 0m 226h 0m 0m 0m
61  0m 0m 0m 629m 0m 0m 0m

```

Switch#

To interpret this data:

- Number of sensors is the total number of temperature sensors that will be recorded. A column for each sensor is displayed with temperatures listed under the number of each sensor, as available.
- Sampling frequency is the time between measurements.
- Maximum time of storage determines the maximum amount of time, in minutes, that can pass when the temperature remains unchanged and the data is not saved to storage media. After this time, a temperature record will be saved even if the temperature has not changed.
- The Sensor column lists the name of the sensor.
- The ID column lists an assigned identifier for the sensor.
- Maximum Temperature 0C shows the highest recorded temperature per sensor.
- Temp indicates a recorded temperature in degrees Celsius in the historical record. Columns following show the total time each sensor has recorded that temperature.
- Sensor ID is an assigned number, so that temperatures for the same sensor can be stored together.

Operational Uptime

The operational uptime tracking begins when the module is powered on, and information is retained for the life of the module.

Operational Uptime Example

Switch# **sh logging onboard uptime detail**

----- UPTIME SUMMARY INFORMATION -----

First customer power on : 04/13/2010 19:45:08
 Total uptime : 1 years 34 weeks 3 days 12 hours 50 minutes
 Total downtime : 1 years 7 weeks 3 days 18 hours 12 minutes
 Number of resets : 1409
 Number of slot changes : 19
 Current reset reason : 0x0
 Current reset timestamp : 01/29/2013 21:56:43
 Current slot : 5
 Current subslot : 0
 Current uptime : 0 years 0 weeks 0 days 0 hours 20 minutes

 Reset | |
 Reason | Count |

No historical data to display

----- UPTIME CONTINUOUS INFORMATION -----

Time Stamp	Reset	Uptime
MM/DD/YYYY HH:MM:SS	Reason	years weeks days hours minutes
04/13/2010 19:45:08	0x0	0 0 0 0 0
04/13/2010 22:26:50	0x9	0 0 0 2 0
04/14/2010 18:54:42	0x9	0 0 0 20 0
04/14/2010 21:31:00	0x9	0 0 0 2 0
04/14/2010 22:04:15	0x9	0 0 0 0 25
04/14/2010 22:22:20	0x9	0 0 0 0 5
04/14/2010 23:05:58	0x9	0 0 0 0 5
04/15/2010 19:03:11	0x9	0 0 0 19 0
04/15/2010 21:29:22	0x9	0 0 0 2 0
04/15/2010 21:49:49	0x8	0 0 0 0 10
04/16/2010 18:46:03	0x9	0 0 0 20 0
04/16/2010 19:25:37	0x9	0 0 0 0 25
04/16/2010 19:34:59	0x9	0 0 0 0 0
04/16/2010 19:46:06	0x9	0 0 0 0 0
04/16/2010 19:57:16	0x9	0 0 0 0 5
04/16/2010 20:17:55	0x9	0 0 0 0 0

Time Stamp	Reset	Uptime
MM/DD/YYYY HH:MM:SS	Reason	years weeks days hours minutes
04/16/2010 20:31:28	0x9	0 0 0 0 0
04/16/2010 20:50:07	0x9	0 0 0 0 10
04/16/2010 22:45:15	0x9	0 0 0 0 0
04/18/2010 19:55:25	0x9	0 0 0 0 0
04/18/2010 20:01:52	0x9	0 0 0 0 0
04/19/2010 00:21:42	0x9	0 0 0 0 0
04/19/2010 01:20:33	0x0	0 0 0 0 30
04/19/2010 19:25:04	0x9	0 0 0 15 0
04/19/2010 20:05:04	0x9	0 0 0 0 15
04/19/2010 20:55:43	0x9	0 0 0 0 0
04/19/2010 21:11:52	0x9	0 0 0 0 0
04/19/2010 21:20:35	0x9	0 0 0 0 0
04/19/2010 21:39:45	0x9	0 0 0 0 10
04/19/2010 21:54:50	0x9	0 0 0 0 5
04/19/2010 22:11:48	0x9	0 0 0 0 5

```

04/19/2010 22:35:38 0x9 0 0 0 0 5
04/19/2010 22:49:41 0x9 0 0 0 0 0
04/19/2010 23:12:58 0x9 0 0 0 0 10
04/20/2010 00:36:04 0x9 0 0 0 1 0
04/20/2010 00:49:19 0x9 0 0 0 0 5
04/20/2010 00:58:29 0x9 0 0 0 0 0
04/20/2010 16:51:06 0x9 0 0 0 15 0

```

Switch#

The operational uptime application tracks the following events:

- Date and time the customer first powered on a component.
- Total uptime and downtime for the component in years, weeks, days, hours, and minutes.
- Total number of component resets.
- Total number of slot (module) changes.
- Current reset timestamp to include the date and time.
- Current slot (module) number of the component.
- Current uptime in years, weeks, days, hours, and minutes.

Interrupts

Interrupts are generated by system components that require attention from the CPU such as ASICs and NMIs. Interrupts are generally related to hardware limit conditions or errors that need to be corrected.

The continuous format records each time a component is interrupted, and this record is stored and used as base information for subsequent records. Each time the list is saved, a timestamp is added. Time differences from the previous interrupt are counted, so that technical personnel can gain a complete record of the component's operational history when an error occurs.

Interrupts Example

Switch# **sh logging onboard interrupt detail**

----- INTERRUPT SUMMARY INFORMATION

Name	ID	Offset	Bit	Count
dropped	2	0x0004	0	27323
ipp	6	0x5A00	0	983763
ipp high	10	0x700A	0	34105
ipp low	11	0x9000	0	30211

----- CONTINUOUS INTERRUPT INFORMATION

MM/DD/YYYY HH:MM:SS mmm	Name	ID	Offset	Bit
12/12/2011 16:06:43 0	ipp high	10	0x7AEA	7
12/12/2011 16:06:43 0	dropped	2	0x0006	0
12/12/2011 16:06:46 0	ipp high	10	0x7AEC	0
12/12/2011 16:06:46 0	ipp high	10	0x7AEC	1
12/12/2011 16:06:46 0	ipp high	10	0x7AEC	4
12/12/2011 16:06:46 0	ipp high	10	0x7AEC	5
12/12/2011 16:06:46 0	ipp low	11	0xC000	0
12/12/2011 16:06:46 0	ipp low	11	0xC000	1
12/12/2011 16:06:46 0	ipp low	11	0xC000	4

```

12/12/2011 16:06:46 0 ipp low 11 0xC000 5
12/12/2011 16:06:46 0 ipp high 10 0x7AEA 0
12/12/2011 16:06:46 0 ipp high 10 0x7AEA 2
12/12/2011 16:06:46 0 ipp high 10 0x7AEA 3
12/12/2011 16:06:46 0 ipp high 10 0x7AEA 4
12/12/2011 16:06:46 0 ipp high 10 0x7AEA 6
12/12/2011 16:06:46 0 ipp high 10 0x7AEA 7
12/12/2011 16:06:46 0 dropped 2 0x0006 0
12/12/2011 16:06:49 0 ipp high 10 0x7AEC 0
12/12/2011 16:06:49 0 ipp high 10 0x7AEC 1
-----

```

Switch#

To interpret this data:

- Name is a description of the component including its position in the device.
- ID is an assigned field for data storage.
- Offset is the register offset from a component register's base address.
- Bit is the interrupt bit number recorded from the component's internal register.
- The timestamp shows the date and time that an interrupt occurred down to the millisecond.

Message Logging

The OBFL feature logs standard system messages. Instead of displaying the message to a terminal, the message is written to and stored in a file, so the message can be accessed and read at a later time. System messages range from level 1 alerts to level 7 debug messages, and these levels can be specified in the **hw module logging onboard** command.

Error Message Log Example

Switch# **sh logging onboard message det**

ERROR MESSAGE SUMMARY INFORMATION

```

-----
Facility-Sev-Name      | Count | Persistence Flag
MM/DD/YYYY HH:MM:SS
-----
%CAT4K-3-DIAGNOSTICS_PASSED :    22    LAST
11/24/2010 15:46:20 module passed diagnostics
%CAT4K-2-DIAGNOSTIC_STATUS :    22    LAST
11/24/2010 15:46:20 diagnostic Packet memory Skipped
-----

```

ERROR MESSAGE CONTINUOUS INFORMATION

```

-----
MM/DD/YYYY HH:MM:SS Facility-Sev-Name
-----
12/15/2010 11:32:39 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
12/15/2010 11:32:39 %CAT4K-2-DIAGNOSTIC_STATUS : diagnostic Packet memory Skipped
12/15/2010 13:03:41 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
12/15/2010 13:03:41 %CAT4K-2-DIAGNOSTIC_STATUS : diagnostic Packet memory Skipped
12/15/2010 13:25:02 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
12/15/2010 13:25:02 %CAT4K-2-DIAGNOSTIC_STATUS : diagnostic Packet memory Skipped
12/15/2010 13:45:34 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
12/15/2010 13:45:34 %CAT4K-2-DIAGNOSTIC_STATUS : diagnostic Packet memory Skipped
12/15/2010 14:05:01 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
12/15/2010 14:05:01 %CAT4K-2-DIAGNOSTIC_STATUS : diagnostic Packet memory Skipped
-----

```

```
12/15/2010 14:35:51 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
12/15/2010 14:35:51 %CAT4K-2-DIAGNOSTIC_STATUS : diagnostic Packet memory Skipped
-----
```

Switch#

To interpret this data:

- A timestamp shows the date and time the message was logged.
- Facility-Sev-Name is a coded naming scheme for a system message, as follows:
 - The Facility code consists of two or more uppercase letters that indicate the hardware device (facility) to which the message refers.
 - Sev is a single-digit code from 1 to 7 that reflects the severity of the message.
 - Name is one or two code names separated by a hyphen that describe the part of the system from where the message is coming.
- The error message follows the Facility-Sev-Name codes. For more information about system messages, see the [Cisco IOS System and Error Messages](#) guide.
- Count indicates the number of instances of this message that is allowed in the history file. Once that number of instances has been recorded, the oldest instance will be removed from the history file to make room for new ones.
- The Persistence Flag gives a message priority over others that do not have the flag set.

Default Settings for OBFL

The OBFL feature is enabled by default. Because of the valuable information this feature offers technical personnel, it should not be disabled.

Enabling OBFL

To enable OBFL, perform this task:

	Command or Action	Purpose
Step 1	Router> enable	Enables privileged EXEC mode (enter your password if prompted).
Step 2	Router# configure terminal	Enters global configuration mode.
Step 3	Router(config)# hw-module module module-number logging onboard [message level {1-7}]	Enables OBFL on the specified hardware module. Note By default, all system messages sent to a device are logged by the OBFL feature. You can define a specific message level (only level 1 messages, as an example) to be logged using the message level keywords.
Step 4	Router(config)# end	Ends global configuration mode.

Configuration Examples for OBFL

The important OBFL feature is the information that is displayed by the **show logging onboard module** privileged EXEC command. This section provides the following examples of how to enable and display OBFL records.

- [Enabling OBFL Message Logging: Example](#)
- [OBFL Message Log: Example](#)
- [OBFL Component Uptime Report: Example](#)
- [OBFL Report for a Specific Time: Example](#)

Enabling OBFL Message Logging: Example

The following example shows how to configure OBFL message logging at level 3:

```
Router(config)# hw-module module 1 logging onboard message level 3
```

OBFL Message Log: Example

The following example shows how to display the system messages that are being logged for module 2:

```
Switch# show logging onboard module 2 message continuous
-----
---
ERROR MESSAGE CONTINUOUS INFORMATION
-----
---
MM/DD/YYYY HH:MM:SS Facility-Sev-Name
-----
---
12/13/2012 18:12:32 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
12/14/2012 17:50:55 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
12/20/2012 17:45:55 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
12/20/2012 19:55:27 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
12/20/2012 20:37:27 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
12/21/2012 16:09:15 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
01/07/2013 02:43:06 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
01/07/2013 04:59:38 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
01/16/2013 15:36:34 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
01/17/2013 12:41:44 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
01/18/2013 14:03:24 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
01/18/2013 14:16:09 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
01/18/2013 14:21:59 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
01/18/2013 15:23:04 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
01/18/2013 15:41:29 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
01/22/2013 14:59:10 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
01/24/2013 11:47:27 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
01/24/2013 16:40:58 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
-----
Switch#
```

OBFL Component Uptime Report: Example

The following example shows how to display a summary report for component uptimes for module 2:

```
Switch# show logging onboard module 2 uptime
-----
---
UPTIME SUMMARY INFORMATION
-----
---
First customer power on : 12/13/2012 18:12:53
Total uptime           :  0 years  0 weeks  4 days 15 hours 55 minutes
Total downtime         :  0 years  6 weeks  0 days 12 hours 18 minutes
Number of resets        : 20
Number of slot changes  : 1
Current reset reason     : 0x0
Current reset timestamp : 01/29/2013 21:56:18
Current slot            : 2
Current subslot          : 0
Current uptime           :  0 years  0 weeks  0 days  0 hours 30 minutes
-----
---
Reset |      |
Reason| Count|
-----
---
No historical data to display
-----
Switch#
```

OBFL Report for a Specific Time: Example

The following example shows how to display continuous reports for all components during a specific time period:

```
Switch# show logging onboard module 2 continuous start 18:12:32 13 Dec 2012 end 16:40:58
24 Jan 2013
PID: WS-C4510R+E      , VID: 6  , SN: FOX1503GL5V
```

```
-----
UPTIME CONTINUOUS INFORMATION
-----
Time Stamp           | Reset | Uptime
MM/DD/YYYY HH:MM:SS | Reason| years weeks days hours minutes
-----
12/13/2012 18:12:53  0x0      0    0    0    0    0
12/14/2012 17:51:14  0x0      0    0    0   23    0
12/20/2012 17:45:52  0x0      0    0    0    1    0
12/20/2012 19:55:22  0x0      0    0    0    2    0
12/20/2012 20:37:26  0x0      0    0    0    0   40
12/21/2012 16:09:14  0x0      0    0    0    0   10
01/07/2013 02:43:04  0x0      0    0    0    2    0
01/07/2013 04:59:35  0x0      0    0    0    2    0
01/16/2013 15:36:32  0x0      0    0    1   17    0
01/17/2013 12:41:42  0x0      0    0    0    3    0
01/18/2013 14:03:21  0x0      0    0    1    1    0
01/18/2013 14:16:08  0x0      0    0    0    0   10
01/18/2013 14:21:58  0x0      0    0    0    0    0
01/18/2013 15:23:02  0x0      0    0    0    1    0
01/18/2013 15:41:25  0x0      0    0    0    0   15
01/22/2013 14:59:05  0x0      0    0    0    3    0
```

Time Stamp	Reset	Uptime					
MM/DD/YYYY HH:MM:SS	Reason	years weeks days hours minutes					
01/24/2013 11:47:25	0x0	0 0 0 3 0					
01/24/2013 16:40:56	0x0	0 0 0 3 0					

ENVIRONMENT CONTINUOUS INFORMATION

MM/DD/YYYY HH:MM:SS	Device	Name	IOS Version	F/W Ver	RAM(KB)	Event
		VID PID	TAN	Serial No		
12/13/2012 18:12:57	slot-2:	NA	NA	0 Inserted		
	Cis WS-C4510R+E	NA	FOX1503GL5V			
12/14/2012 17:50:55	slot-2:	NA	NA	0 Inserted		
	Cis WS-C4510R+E	NA	FOX1503GL5V			
12/20/2012 17:45:55	slot-2:	NA	NA	0 Inserted		
	Cis WS-C4510R+E	NA	FOX1503GL5V			
12/20/2012 19:55:27	slot-2:	NA	NA	0 Inserted		
	Cis WS-C4510R+E	NA	FOX1503GL5V			
12/20/2012 20:37:27	slot-2:	NA	NA	0 Inserted		
	Cis WS-C4510R+E	NA	FOX1503GL5V			
12/21/2012 16:09:15	slot-2:	NA	NA	0 Inserted		
	Cis WS-C4510R+E	NA	FOX1503GL5V			
01/07/2013 02:43:06	slot-2:	NA	NA	0 Inserted		
	Cis WS-C4510R+E	NA	FOX1503GL5V			
01/07/2013 04:59:38	slot-2:	NA	NA	0 Inserted		

ENVIRONMENT CONTINUOUS INFORMATION

MM/DD/YYYY HH:MM:SS	Device	Name	IOS Version	F/W Ver	RAM(KB)	Event
		VID PID	TAN	Serial No		
	Cis WS-C4510R+E	NA	FOX1503GL5V			
01/16/2013 15:36:34	slot-2:	NA	NA	0 Inserted		
	Cis WS-C4510R+E	NA	FOX1503GL5V			
01/17/2013 12:41:44	slot-2:	NA	NA	0 Inserted		
	Cis WS-C4510R+E	NA	FOX1503GL5V			
01/18/2013 14:03:24	slot-2:	NA	NA	0 Inserted		
	Cis WS-C4510R+E	NA	FOX1503GL5V			
01/18/2013 14:16:09	slot-2:	NA	NA	0 Inserted		
	Cis WS-C4510R+E	NA	FOX1503GL5V			
01/18/2013 14:21:59	slot-2:	NA	NA	0 Inserted		
	Cis WS-C4510R+E	NA	FOX1503GL5V			
01/18/2013 15:23:04	slot-2:	NA	NA	0 Inserted		
	Cis WS-C4510R+E	NA	FOX1503GL5V			
01/18/2013 15:41:29	slot-2:	NA	NA	0 Inserted		
	Cis WS-C4510R+E	NA	FOX1503GL5V			
01/22/2013 14:59:10	slot-2:	NA	NA	0 Inserted		
	Cis WS-C4510R+E	NA	FOX1503GL5V			
01/24/2013 11:47:27	slot-2:	NA	NA	0 Inserted		
	Cis WS-C4510R+E	NA	FOX1503GL5V			
01/24/2013 16:40:58	slot-2:	NA	NA	0 Inserted		
	Cis WS-C4510R+E	NA	FOX1503GL5V			

TEMPERATURE CONTINUOUS INFORMATION

Sensor	ID
--------	----

```

Air inlet                                0
Air inlet remote                        1
Air outlet                             2
Air outlet remote                      3

```

```

-----
Time Stamp | Sensor Temperature 0C
MM/DD/YYYY HH:MM:SS | 0 1 2 3
-----
01/18/2013 12:18:59 32 23 33 27
01/18/2013 12:28:59 32 23 33 27
01/18/2013 12:38:59 32 23 33 27
01/18/2013 12:48:00 32 23 33 27
01/18/2013 12:58:00 32 23 33 27
01/18/2013 13:08:00 32 23 33 27
01/18/2013 13:18:00 32 23 33 27
01/18/2013 13:28:00 32 23 33 27
01/18/2013 13:38:00 32 23 33 27
01/18/2013 13:48:00 32 23 33 27
01/18/2013 13:58:00 32 23 33 27
01/18/2013 14:03:21 30 23 31 27
01/18/2013 14:12:22 32 23 33 27
01/18/2013 14:16:08 30 23 31 27
01/18/2013 14:21:58 30 23 31 26
01/18/2013 14:31:58 32 23 33 28
01/18/2013 14:40:59 32 23 33 28

```

```

-----
Time Stamp | Sensor Temperature 0C
MM/DD/YYYY HH:MM:SS | 0 1 2 3
-----
01/18/2013 14:47:04 26 23 26 25
01/18/2013 14:57:04 24 22 24 23
01/18/2013 15:07:04 24 22 24 23
01/18/2013 15:17:04 24 22 24 23
01/18/2013 15:23:03 25 22 26 23
01/18/2013 15:25:03 30 22 31 25
01/18/2013 15:35:03 32 23 33 27
01/18/2013 15:41:25 30 23 31 26
01/18/2013 15:51:25 32 23 33 27
01/18/2013 16:00:27 32 23 33 27
01/18/2013 16:10:27 32 23 33 27
01/18/2013 16:20:27 32 23 33 28
01/18/2013 16:30:27 32 23 33 27
01/18/2013 16:40:27 32 23 33 27
01/18/2013 16:50:27 32 23 33 27
01/18/2013 17:00:27 31 23 33 27
01/18/2013 17:10:27 32 23 33 27
01/18/2013 17:20:27 32 23 33 27
01/18/2013 17:30:27 31 23 33 27
01/18/2013 17:40:27 31 23 33 27
01/18/2013 17:50:27 31 23 33 28
01/18/2013 18:00:27 32 24 34 30
01/18/2013 18:09:28 32 24 34 31

```

```

-----
Time Stamp | Sensor Temperature 0C
MM/DD/YYYY HH:MM:SS | 0 1 2 3
-----
01/18/2013 18:19:28 33 24 35 32
01/18/2013 18:29:28 33 25 35 33
01/18/2013 18:39:28 32 25 36 34
01/18/2013 18:49:28 32 25 36 34
01/18/2013 18:54:28 34 26 39 39

```

```

01/18/2013 19:04:28 35 27 42 44
01/18/2013 19:14:28 35 27 42 44
01/22/2013 14:59:05 25 22 26 22
01/22/2013 15:02:05 30 23 31 26
01/22/2013 15:09:06 32 24 34 31
01/22/2013 15:12:06 33 26 37 36
01/22/2013 15:15:06 34 27 40 41
01/22/2013 15:25:06 36 28 43 46
01/22/2013 15:35:06 36 28 43 46
01/22/2013 15:45:06 36 28 43 46
01/22/2013 15:55:06 36 28 44 46
01/22/2013 16:05:06 36 28 43 46
01/22/2013 16:14:07 36 28 43 46
01/22/2013 16:24:07 35 28 43 46
01/22/2013 16:34:07 36 28 44 46
01/22/2013 16:44:07 36 28 43 46
01/22/2013 16:54:07 36 28 43 46
01/22/2013 16:58:07 34 26 41 41

```

```

-----
Time Stamp | Sensor Temperature 0C
MM/DD/YYYY HH:MM:SS | 0 1 2 3
-----

```

```

01/22/2013 17:00:07 32 24 37 34
01/22/2013 17:10:07 31 24 34 32
01/22/2013 17:20:07 31 23 34 30
01/22/2013 17:30:07 32 24 35 33
01/22/2013 17:40:07 32 24 35 33
01/22/2013 17:49:08 32 24 35 33
01/22/2013 17:59:08 32 24 35 33
01/24/2013 11:47:25 26 22 26 23
01/24/2013 11:49:25 30 24 31 28
01/24/2013 11:56:25 33 25 35 33
01/24/2013 12:06:25 32 25 35 33
01/24/2013 12:16:25 33 25 35 33
01/24/2013 12:26:25 33 25 35 33
01/24/2013 12:36:25 33 25 36 33
01/24/2013 12:46:25 33 25 36 33
01/24/2013 12:56:25 34 27 39 38
01/24/2013 13:01:25 35 28 42 43
01/24/2013 13:11:25 36 29 44 46
01/24/2013 13:21:25 37 29 45 47
01/24/2013 13:30:26 37 29 45 47
01/24/2013 13:40:26 37 29 45 47
01/24/2013 13:50:26 37 29 45 47
01/24/2013 14:00:26 36 29 45 47

```

```

-----
Time Stamp | Sensor Temperature 0C
MM/DD/YYYY HH:MM:SS | 0 1 2 3
-----

```

```

01/24/2013 14:10:26 37 29 45 47
01/24/2013 14:20:26 37 29 45 47
01/24/2013 14:30:26 36 28 43 45
01/24/2013 14:32:26 34 26 39 39
01/24/2013 14:38:26 33 25 36 33
01/24/2013 14:48:26 34 26 37 36
01/24/2013 14:58:26 34 26 38 36
01/24/2013 15:08:26 34 26 38 37
01/24/2013 15:17:27 34 26 38 37
01/24/2013 15:27:27 34 26 38 37
01/24/2013 16:40:56 26 22 27 24
-----

```

```
-----
ERROR MESSAGE CONTINUOUS INFORMATION
-----
```

```
MM/DD/YYYY HH:MM:SS Facility-Sev-Name
-----
```

```
12/13/2012 18:12:32 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
12/14/2012 17:50:55 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
12/20/2012 17:45:55 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
12/20/2012 19:55:27 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
12/20/2012 20:37:27 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
12/21/2012 16:09:15 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
01/07/2013 02:43:06 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
01/07/2013 04:59:38 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
01/16/2013 15:36:34 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
01/17/2013 12:41:44 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
01/18/2013 14:03:24 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
01/18/2013 14:16:09 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
01/18/2013 14:21:59 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
01/18/2013 15:23:04 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
01/18/2013 15:41:29 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
01/22/2013 14:59:10 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
01/24/2013 11:47:27 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
-----
```

```
ERROR MESSAGE CONTINUOUS INFORMATION
-----
```

```
MM/DD/YYYY HH:MM:SS Facility-Sev-Name
-----
```

```
01/24/2013 16:40:58 %CAT4K-3-DIAGNOSTICS_PASSED : module passed diagnostics
-----
```

```
Switch#
```



Configuring SNMP

This chapter describes how to configure the Simple Network Management Protocol (SNMP) on the Catalyst 4500 series switch.

This chapter consists of these sections:

- [About SNMP, page 75-1](#)
- [Configuring SNMP, page 75-5](#)
- [Displaying SNMP Status, page 75-16](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About SNMP

SNMP is an application-layer protocol that provides a message format for communication between managers and agents. The SNMP system consists of an SNMP manager, an SNMP agent, and a MIB. The SNMP manager can be part of a network management system (NMS) such as CiscoWorks. The agent and MIB reside on the switch. To configure SNMP on the switch, you define the relationship between the manager and the agent.

The SNMP agent contains MIB variables whose values the SNMP manager can request or change. A manager can get a value from an agent or store a value into the agent. The agent gathers data from the MIB, the repository for information about device parameters and network data. The agent can also respond to a manager's requests to get or set data.

An agent can send unsolicited traps to the manager. Traps are messages alerting the SNMP manager to a condition on the network. Traps can mean improper user authentication, restarts, link status (up or down), MAC address tracking, closing of a Transmission Control Protocol (TCP) connection, loss of connection to a neighbor, or other significant events.

This section includes information about these topics:

- [SNMP Versions, page 75-2](#)
- [SNMP Manager Functions, page 75-3](#)
- [SNMP Agent Functions, page 75-4](#)
- [SNMP Community Strings, page 75-4](#)
- [Using SNMP to Access MIB Variables, page 75-4](#)
- [SNMP Notifications, page 75-5](#)

SNMP Versions

The Catalyst 4500 series switch supports these SNMP versions:

- **SNMPv1**—The Simple Network Management Protocol, a Full Internet Standard, defined in RFC 1157.
- **SNMPv2C** replaces the Party-based Administrative and Security Framework of SNMPv2Classic with the community-string-based Administrative Framework of SNMPv2C while retaining the bulk retrieval and improved error handling of SNMPv2Classic. It has these features:
 - **SNMPv2**—Version 2 of the Simple Network Management Protocol, a Draft Internet Standard, defined in RFCs 1902 through 1907.
 - **SNMPv2C**—The community-string-based Administrative Framework for SNMPv2, an Experimental Internet Protocol defined in RFC 1901.
- **SNMPv3**—Version 3 of the SNMP is an interoperable standards-based protocol defined in RFCs 2273 to 2275. SNMPv3 provides secure access to devices by authenticating and encrypting packets over the network and includes these security features:
 - **Message integrity**—Ensures that a packet was not tampered with in transit
 - **Authentication**—Determines that the message is from a valid source
 - **Encryption**—Mixes the contents of a package to prevent it from being read by an unauthorized source



Note

To select encryption, enter the **priv** keyword. This keyword is available only when the crypto (encrypted) software image is installed.

Both SNMPv1 and SNMPv2C use a community-based form of security. The community of managers able to access the agent's MIB is defined by an IP address access control list and password.

SNMPv2C includes a bulk retrieval mechanism and more detailed error message reporting to management stations. The bulk retrieval mechanism retrieves tables and large quantities of information, minimizing the number of round-trips required. The SNMPv2C improved error-handling includes expanded error codes that distinguish different kinds of error conditions; these conditions are reported through a single error code in SNMPv1. Error return codes in SNMPv2C report the error type.

SNMPv3 provides for both security models and security levels. A security model is an authentication strategy set up for a user and the group within which you reside. A security level is the permitted level of security within a security model. A combination of the security level and the security model determine which security mechanism is used when handling an SNMP packet. Available security models are SNMPv1, SNMPv2C, and SNMPv3.

The following table identifies the characteristics of the different combinations of security models and levels.

Model	Level	Authentication	Encryption	Result
SNMPv1	noAuthNoPriv	Community string	No	Uses a community string match for authentication.
SNMPv2C	noAuthNoPriv	Community string	No	Uses a community string match for authentication.
SNMPv3	noAuthNoPriv	Username	No	Uses a username match for authentication.
SNMPv3	authNoPriv	MD5 or SHA	No	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms.
SNMPv3	authPriv (requires the cryptographic software image)	MD5 or SHA	DES	Provides authentication based on the HMAC-MD5 or HMAC-SHA algorithms. Provides DES 56-bit encryption in addition to authentication based on the CBC-DES (DES-56) standard.

You must configure the SNMP agent to use the SNMP version supported by the management station. Because an agent can communicate with multiple managers, you can configure the software to support communications using SNMPv1, and SNMPv2C, and SNMPv3 protocols.

SNMP Manager Functions

The SNMP manager uses information in the MIB to perform the operations described in [Table 75-1](#).

Table 75-1 *SNMP Operations*

Operation	Description
get-request	Retrieves a value from a specific variable.
get-next-request	Retrieves a value from a variable within a table. ¹
get-bulk-request ²	Retrieves large blocks of data, such as multiple rows in a table, that would otherwise require the transmission of many small blocks of data.
get-response	Replies to a get-request, get-next-request, and set-request sent by an NMS.
set-request	Stores a value in a specific variable.
trap	An unsolicited message sent by an SNMP agent to an SNMP manager when some event has occurred.

1. With this operation, an SNMP manager does not need to know the exact variable name. A sequential search is performed to find the needed variable from within a table.
2. The **get-bulk** command only works with SNMPv2 or later.

SNMP Agent Functions

The SNMP agent responds to SNMP manager requests as follows:

- **Get a MIB variable**—The SNMP agent begins this function in response to a request from the NMS. The agent retrieves the value of the requested MIB variable and responds to the NMS with that value.
- **Set a MIB variable**—The SNMP agent begins this function in response to a message from the NMS. The SNMP agent changes the value of the MIB variable to the value requested by the NMS.

The SNMP agent also sends unsolicited trap messages to notify an NMS that a significant event has occurred on the agent. Examples of trap conditions include, but are not limited to, when a port or module goes up or down, when spanning-tree topology changes occur, and when authentication failures occur.

SNMP Community Strings

SNMP community strings authenticate access to MIB objects and function as embedded passwords. In order for the NMS to access the switch, the community string definitions on the NMS must match at least one of the three community string definitions on the switch.

A community string can have one of these attributes:

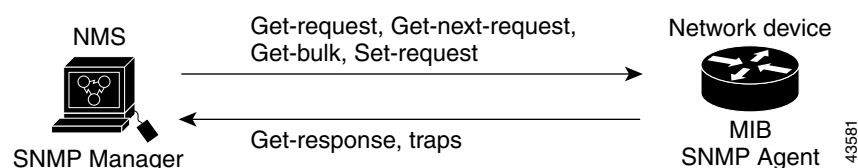
- **Read-only (RO)**—Gives read access to authorized management stations to all objects in the MIB except the community strings, but does not allow write access
- **Read-write (RW)**—Gives read and write access to authorized management stations to all objects in the MIB, but does not allow access to the community strings
- **Read-write-all**—Gives read and write access to authorized management stations to all objects in the MIB, including the community strings

Using SNMP to Access MIB Variables

An example of an NMS is the CiscoWorks network management software. CiscoWorks 2000 software uses the switch MIB variables to set device variables and to poll devices on the network for specific information. The results of a poll can be displayed as a graph and analyzed to troubleshoot internetworking problems, increase network performance, verify the configuration of devices, monitor traffic loads, and more.

As shown in [Figure 75-1](#), the SNMP agent gathers data from the MIB. The agent can send traps, or notification of certain events, to the SNMP manager, which receives and processes the traps. Traps alert the SNMP manager to a condition on the network such as improper user authentication, restarts, link status (up or down), MAC address tracking, and so forth. The SNMP agent also responds to MIB-related queries sent by the SNMP manager in *get-request*, *get-next-request*, and *set-request* format.

Figure 75-1 SNMP Network



SNMP Notifications

SNMP allows the switch to send notifications to SNMP managers when particular events occur. SNMP notifications can be sent as traps or inform requests. In command syntax, unless there is an option in the command to select either traps or informs, the keyword *traps* refers to either traps or informs, or both. Use the **snmp-server host** command to specify whether to send SNMP notifications as traps or informs.

**Note**

SNMPv1 does not support informs.

Traps are unreliable because the receiver does not send an acknowledgment when it receives a trap, and the sender cannot determine if the trap was received. When an SNMP manager receives an inform request, it acknowledges the message with an SNMP response protocol data unit (PDU). If the sender does not receive a response, the inform request can be sent again. Because they can be resent, informs are more likely than traps to reach their intended destination.

The characteristics that make informs more reliable than traps also consume more resources in the switch and in the network. Unlike a trap, which is discarded as soon as it is sent, an inform request is held in memory until a response is received or the request times out. Traps are sent only once, but an inform might be resent or retried several times. The retries increase traffic and contribute to a higher overhead on the network. Traps and informs require a trade-off between reliability and resources. If it is important that the SNMP manager receive every notification, use inform requests. If traffic on the network or memory in the switch is a concern and notification is not required, use traps.

Configuring SNMP

This section describes how to configure SNMP on your switch. It contains this configuration information:

- [Default SNMP Configuration, page 75-5](#)
- [SNMP Configuration Guidelines, page 75-6](#)
- [Disabling the SNMP Agent, page 75-7](#)
- [Configuring Community Strings, page 75-7](#)
- [Configuring SNMP Groups and Users, page 75-9](#)
- [Configuring SNMP Notifications, page 75-11](#)
- [Setting the Agent Contact and Location Information, page 75-14](#)
- [Limiting TFTP Servers Used Through SNMP, page 75-15](#)
- [SNMP Examples, page 75-15](#)

Default SNMP Configuration

[Table 75-2](#) shows the default SNMP configuration.

Table 75-2 **Default SNMP Configuration**

Feature	Default Setting
SNMP agent	Enabled
SNMP trap receiver	None configured
SNMP traps	None enabled except the trap for TCP connections (tty)
SNMP version	If no version keyword is present, the default is Version 1.
SNMPv3 authentication	If no keyword is entered, the default is the noauth (noAuthNoPriv) security level.
SNMP notification type	If no type is specified, all notifications are sent.

SNMP Configuration Guidelines

An SNMP *group* is a table that maps SNMP users to SNMP views. An SNMP *user* is a member of an SNMP group. An SNMP *host* is the recipient of an SNMP trap operation. An SNMP *engine ID* is a name for the local or remote SNMP engine.

When configuring SNMP, follow these guidelines:

- When configuring an SNMP group, do not specify a notify view. The **snmp-server host** global configuration command autogenerates a notify view for you and then adds it to the group associated with that user. Modifying the group's notify view affects all users associated with that group. For information about when you should configure notify views, see the *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2*.
- To configure a remote user, specify the IP address or port number for the remote SNMP agent of the device where you reside.
- Before you configure remote users for a particular agent, configure the SNMP engine ID, using the **snmp-server engineID** global configuration with the **remote** option. The remote agent's SNMP engine ID and user password are used to compute the authentication and privacy digests. If you do not configure the remote engine ID first, the configuration command fails.
- When configuring SNMP informs, you need to configure the SNMP engine ID for the remote agent in the SNMP database before you can send proxy requests or informs to it.
- If a local user is not associated with a remote host, the switch does not send informs for the **auth** (authNoPriv) and the **priv** (authPriv) authentication levels.
- Changing the value of the SNMP engine ID has important side effects. A user's password (entered on the command line) is converted to an MD5 or SHA security digest based on the password and the local engine ID. The command-line password is then destroyed, as required by RFC 2274. Because of this deletion, if the value of the engine ID changes, the security digests of SNMPv3 users become invalid, and you need to reconfigure SNMP users by using the **snmp-server user username** global configuration command. Similar restrictions require the reconfiguration of community strings when the engine ID changes.

Disabling the SNMP Agent

To disable the SNMP agent, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# no snmp-server	Disables the SNMP agent operation.
Step 3	Switch(config)# end	Returns to privileged EXEC mode.
Step 4	Switch# show running-config	Verifies your entries.
Step 5	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

The **no snmp-server** global configuration command disables all running versions (Version 1, Version 2C, and Version 3) on the device. No specific Cisco IOS command exists to enable SNMP. The first **snmp-server** global configuration command that you enter enables all versions of SNMP.

Configuring Community Strings

You use the SNMP community string to define the relationship between the SNMP manager and the agent. The community string acts like a password to permit access to the agent on the switch. Optionally, you can specify one or more of these characteristics associated with the string:

- An access list of IP addresses of the SNMP managers that are permitted to use the community string to gain access to the agent
- A MIB view, which defines the subset of all MIB objects accessible to the given community
- Read and write or read-only permission for the MIB objects accessible to the community

To configure a community string on the switch, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# [no] snmp-server community string [view view-name] [ro rw] [access-list-number]	<p>Configures the community string.</p> <ul style="list-style-type: none"> For <i>string</i>, specify a string that acts like a password and permits access to the SNMP protocol. You can configure one or more community strings up to 117 characters. (Optional) For view, specify the view record accessible to the community. (Optional) Specify either read-only (ro) if you want authorized management stations to retrieve MIB objects, or specify read-write (rw) if you want authorized management stations to retrieve and modify MIB objects. By default, the community string permits read-only access to all objects. (Optional) For <i>access-list-number</i>, enter an IP standard access list numbered from 1 to 99 and 1300 to 1999. <p>To remove a specific community string, use the no snmp-server community string global configuration command.</p>
Step 3	Switch(config)# access-list access-list-number {deny permit} source [source-wildcard]	<p>(Optional) If you specified an IP standard access list number in Step 2, create the list, repeating the command as many times as necessary.</p> <ul style="list-style-type: none"> For <i>access-list-number</i>, enter the access list number specified in Step 2. The deny keyword denies access if the conditions are matched. The permit keyword permits access if the conditions are matched. For <i>source</i>, enter the IP address of the SNMP managers that are permitted to use the community string to gain access to the agent. (Optional) For <i>source-wildcard</i>, enter the wildcard bits in dotted decimal notation to be applied to the source. Place ones in the bit positions that you want to ignore. <p>Recall that the access list is always terminated by an implicit deny statement for everything.</p>
Step 4	Switch(config)# end	Returns to privileged EXEC mode.
Step 5	Switch# show running-config	Verifies your entries.
Step 6	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.



Note

To disable access for an SNMP community, set the community string for that community to the null string (do not enter a value for the community string).

**Note**

You cannot use the **snmp-server enable informs** command. To enable the sending of SNMP inform notifications, use the **snmp-server enable traps** command combined with the **snmp-server host host-addr informs** command.

This example shows how to assign the string *comaccess* to SNMP, to allow read-only access, and to specify that IP access list 4 can use the community string to gain access to the switch SNMP agent:

```
Switch(config)# snmp-server community comaccess ro 4
```

Configuring SNMP Groups and Users

You can specify an identification name (engine ID) for the local or remote SNMP server engine on the switch. You can configure an SNMP server group that maps SNMP users to SNMP views, and you can add new users to the SNMP group.

To configure SNMP on the switch, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# snmp-server engineID { local <i>engineid-string</i> remote <i>ip-address</i> [udp-port <i>port-number</i>] <i>engineid-string</i> }	Configures a name for either the local or remote copy of SNMP. <ul style="list-style-type: none"> The <i>engineid-string</i> is a 24-character ID string with the name of the copy of SNMP. You need not specify the entire 24-character engine ID if it has trailing zeros. Specify only the portion of the engine ID up to the point where only zeros remain in the value. For example, to configure an engine ID of 123400000000000000000000, you can enter this: snmp-server engineID local 1234 If you select remote, specify the <i>ip-address</i> of the device that contains the remote copy of SNMP and the optional UDP port on the remote device. The default is 162.

Step 3

Command	Purpose
<pre>Switch(config)# snmp-server group groupname {v1 v2c v3 [auth noauth priv]} [read readview] [write writeview] [notify notifyview] [access access-list]</pre>	<p>Configures a new SNMP group on the remote device.</p> <ul style="list-style-type: none"> For <i>groupname</i>, specify the name of the group. Specify a security model: <ul style="list-style-type: none"> v1 is the least secure of the possible security models. v2c is the second least secure model. It allows transmission of informs and integers twice the normal width. v3, the most secure, requires you to select an authentication level: <ul style="list-style-type: none"> auth—Enables the Message Digest 5 (MD5) and the Secure Hash Algorithm (SHA) packet authentication. noauth—The noAuthNoPriv security level. It is the default if no keyword is specified. priv—Enables Data Encryption Standard (DES) packet encryption (also called <i>privacy</i>). <p>Note The priv keyword is available only when the crypto software image is installed.</p> <ul style="list-style-type: none"> (Optional) Enter read <i>readview</i> with a string (not to exceed 64 characters) that is the name of the view in which you can only view the contents of the agent. (Optional) Enter write <i>writeview</i> with a string (not to exceed 64 characters) that is the name of the view in which you enter data and configure the contents of the agent. (Optional) Enter notify <i>notifyview</i> with a string (not to exceed 64 characters) that is the name of the view in which you specify a notify, inform, or trap. (Optional) Enter access <i>access-list</i> with a string (not to exceed 64 characters) that is the name of the access list.

	Command	Purpose
Step 4	Switch(config)# snmp-server user <i>username groupname [remote host</i> <i>[udp-port port]] {v1 / v2c v3 [auth</i> <i>{md5 sha} auth-password]} [encrypted]</i> <i>[access access-list]</i>	Configures a new user to an SNMP group. <ul style="list-style-type: none"> • The <i>username</i> is your name on the host that connects to the agent. • The <i>groupname</i> is the name of the group to which you are associated. • (Optional) Enter remote to specify a remote SNMP entity to which you belong and the hostname or IP address of that entity with the optional UDP port number. The default is 162. • Enter the SNMP version number (v1, or v2c, or v3). If you enter v3, you have these additional options: <ul style="list-style-type: none"> – auth is an authentication level setting session, which can be either the HMAC-MD5-96 or the HMAC-SHA-96 authentication level, and requires a password string (not to exceed 64 characters). – encrypted specifies that the password appears in encrypted format. • (Optional) Enter access access-list with a string (not to exceed 64 characters) that is the name of the access list.
Step 5	Switch(config)# end	Returns to privileged EXEC mode.
Step 6	Switch# show running-config	Verifies your entries.
Step 7	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Configuring SNMP Notifications

A trap manager is a management station that receives and processes traps. Traps are system alerts that the switch generates when certain events occur. By default, no trap manager is defined, and no traps are sent. Switches running Cisco IOS Release 12.2(31)SG and later releases can have an unlimited number of trap managers.



Note

Many commands use the word *traps* in the command syntax. Unless there is an option in the command to select either traps or informs, the keyword *traps* refers to either traps, informs, or both. Use the **snmp-server host** command to specify whether to send SNMP notifications as traps or informs.

Table 75-3 describes the supported switch traps (notification types). You can enable any or all of these traps and configure a trap manager to receive them.

Table 75-3 Switch Notification Types

Notification Type Keyword	Description
bgp	Generates BGP state change traps. Note This option is only available when the enhanced multilayer image is installed.
bridge	Generates STP bridge MIB traps.

Table 75-3 **Switch Notification Types (continued)**

Notification Type Keyword	Description
config	Generates a trap for SNMP configuration changes.
config-copy	Generates a trap for SNMP copy configuration changes.
cpu	Allows cpu-related traps.
eigrp	Enable EIGRP traps. Note This option is only available when the enhanced multilayer image is installed.
entity	Generates a trap for SNMP entity changes.
envmon	Generates environmental monitor traps. You can enable any or all of these environmental traps: fan, shutdown, supply, temperature.
flash	Generates SNMP FLASH notifications.
fru-ctrl	Enable SNMP entity FRU control traps.
hsrp	Generates a trap for Hot Standby Router Protocol (HSRP) changes.
ipmulticast	Generates a trap for IP multicast routing changes.
isis	Enable IS-IS traps. Note This option is only available when the enhanced multilayer image is installed.
mac-notification	Generates a trap for MAC address notifications.
msdp	Generates a trap for Multicast Source Discovery Protocol (MSDP) changes. Note This option is only available when the enhanced multilayer image is installed.
ospf	Generates a trap for Open Shortest Path First (OSPF) changes. You can enable any or all of these traps: Cisco specific, errors, link-state advertisement, rate limit, retransmit, and state changes. Note This option is only available when the enhanced multilayer image is installed.
pim	Generates a trap for Protocol-Independent Multicast (PIM) changes. You can enable any or all of these traps: invalid PIM messages, neighbor changes, and rendezvous point (RP)-mapping changes.
port-security	Generates SNMP port security traps. You can also set a maximum trap rate per second. The range is from 0 to 1000; the default is 0, which means that there is no rate limit.
rf	Enable all SNMP traps defined in Cisco-RF-MIB.
snmp	Generates a trap for SNMP-type notifications for authentication, cold start, warm start, link up or link down.
storm-control	Generates a trap for SNMP storm-control. You can also set a maximum trap rate per second. The range is from 0 to 1000; the default is 0 (no limit is imposed; a trap is sent at every occurrence).
stp	Generates SNMP STP Extended MIB traps.
syslog	Generates SNMP syslog traps.

Table 75-3 Switch Notification Types (continued)

Notification Type Keyword	Description
tty	Generates a trap for TCP connections. This trap is enabled by default.
vlan-membership	Generates a trap for SNMP VLAN membership changes.
vlancreate	Generates SNMP VLAN created traps.
vlandelete	Generates SNMP VLAN deleted traps.
vtp	Generates a trap for VLAN Trunking Protocol (VTP) changes.

You can use the **snmp-server host** global configuration command to a specific host to receive the notification types listed in [Table 75-3](#).

To configure the switch to send traps or informs to a host, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# snmp-server engineID remote ip-address engineid-string	Specifies the engine ID for the remote host.
Step 3	Switch(config)# snmp-server user username groupname remote host [udp-port port] {v1 / v2c v3 [auth {md5 sha} auth-password]} [encrypted] [access access-list]	Configures an SNMP user to be associated with the remote host created in Step 2. Note You cannot configure a remote user for an address without first configuring the engine ID for the remote host. If you try to configure the user before configuring the remote engine ID, you receive an error message, and the command is not executed.
Step 4	Switch(config)# snmp-server host host-addr [traps informs] [version {1 / 2c 3 [auth noauth priv]]} community-string [udp-port port] [notification-type]	Specifies the recipient of an SNMP trap operation. <ul style="list-style-type: none"> For <i>host-addr</i>, specify the name or Internet address of the host (the targeted recipient). (Optional) Enter traps (the default) to send SNMP traps to the host. (Optional) Enter informs to send SNMP informs to the host. (Optional) Specify the SNMP version (1, 2c, or 3). SNMPv1 does not support informs. (Optional) For Version 3, select authentication level auth, noauth, or priv. Note The priv keyword is available only when the crypto software image is installed. <ul style="list-style-type: none"> For <i>community-string</i>, enter the password-like community string sent with the notification operation. (Optional) For udp-port port, enter the remote device UDP port. (Optional) For <i>notification-type</i>, use the keywords listed in Table 75-3 on page 75-11. If no type is specified, all notifications are sent.

	Command	Purpose
Step 5	Switch(config)# snmp-server enable traps <i>notification-types</i>	Enables the switch to send traps or informs and specify the type of notifications to be sent. For a list of notification types, see Table 75-3 on page 75-11 , or enter this: snmp-server enable traps ? To enable multiple types of traps, you must enter a separate snmp-server enable traps command for each trap type.
Step 6	Switch(config)# snmp-server trap-source <i>interface-id</i>	(Optional) Specifies the source interface, which provides the IP address for the trap message. This command also sets the source IP address for informs.
Step 7	Switch(config)# snmp-server queue-length <i>length</i>	(Optional) Establishes the message queue length for each trap host. The range is 1 to 1000; the default is 10.
Step 8	Switch(config)# snmp-server trap-timeout <i>seconds</i>	(Optional) Defines how often to resend trap messages. The range is 1 to 1000; the default is 30 seconds.
Step 9	Switch(config)# end	Returns to privileged EXEC mode.
Step 10	Switch# show running-config	Verifies your entries.
Step 11	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

The **snmp-server host** command specifies which hosts receive the notifications. The **snmp-server enable trap** command globally enables the mechanism for the specified notification (for traps and informs). To enable a host to receive an inform, you must configure an **snmp-server host informs** command for the host and globally enable informs by using the **snmp-server enable traps** command.

To remove the specified host from receiving traps, use the **no snmp-server host** *host* global configuration command. The **no snmp-server host** command with no keywords disables traps, but not informs, to the host. To disable informs, use the **no snmp-server host informs** global configuration command. To disable a specific trap type, use the **no snmp-server enable traps** *notification-types* global configuration command.

Setting the Agent Contact and Location Information

To set the system contact and location of the SNMP agent so that these descriptions can be accessed using the configuration file, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# snmp-server contact <i>text</i>	Sets the system contact string. For example: snmp-server contact Dial System Operator at beeper 21555.
Step 3	Switch(config)# snmp-server location <i>text</i>	Sets the system location string. For example: snmp-server location Building 3/Room 222
Step 4	Switch(config)# end	Returns to privileged EXEC mode.

	Command	Purpose
Step 5	Switch# show running-config	Verifies your entries.
Step 6	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Limiting TFTP Servers Used Through SNMP

To limit the TFTP servers used for saving and loading configuration files through SNMP to the servers specified in an access list, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# snmp-server tftp-server-list access-list-number	Limits TFTP servers used for configuration file copies through SNMP to the servers in the access list. For <i>access-list-number</i> , enter an IP standard access list numbered from 1 to 99 and 1300 to 1999.
Step 3	Switch(config)# access-list access-list-number {deny / permit} source [source-wildcard]	Creates a standard access list, repeating the command as many times as necessary. <ul style="list-style-type: none"> For <i>access-list-number</i>, enter the access list number specified in Step 2. The deny keyword denies access if the conditions are matched. The permit keyword permits access if the conditions are matched. For <i>source</i>, enter the IP address of the TFTP servers that can access the switch. (Optional) For <i>source-wildcard</i>, enter the wildcard bits, in dotted decimal notation, to be applied to the source. Place ones in the bit positions that you want to ignore. <p>Note The access list is always terminated by an implicit deny statement for everything.</p>
Step 4	Switch(config)# end	Returns to privileged EXEC mode.
Step 5	Switch# show running-config	Verifies your entries.
Step 6	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

SNMP Examples

This example shows how to enable all versions of SNMP. The configuration permits any SNMP manager to access all objects with read-only permissions using the community string public. This configuration does not cause the switch to send any traps.

```
Switch(config)# snmp-server community public
```

This example shows how to permit any SNMP manager to access all objects with read-only permission using the community string public. The switch also sends VTP traps to the hosts 192.180.1.111 and 192.180.1.33 using SNMPv1 and to the host 192.180.1.27 using SNMPv2C. The community string public is sent with the traps.

```
Switch(config)# snmp-server community public
Switch(config)# snmp-server enable traps vtp
Switch(config)# snmp-server host 192.180.1.27 version 2c public
Switch(config)# snmp-server host 192.180.1.111 version 1 public
Switch(config)# snmp-server host 192.180.1.33 public
```

This example shows how to allow read-only access for all objects to members of access list 4 that use the comaccess community string. No other SNMP managers have access to any objects. SNMP Authentication Failure traps are sent by SNMPv2C to the host cisco.com using the community string public.

```
Switch(config)# snmp-server community comaccess ro 4
Switch(config)# snmp-server enable traps snmp authentication
Switch(config)# snmp-server host cisco.com version 2c public
```

This example shows how to send Entity MIB traps to the host cisco.com. The community string is restricted. The first line enables the switch to send Entity MIB traps in addition to any traps previously enabled. The second line specifies the destination of these traps and overwrites any previous **snmp-server host** commands for the host cisco.com.

```
Switch(config)# snmp-server enable traps entity
Switch(config)# snmp-server host cisco.com restricted entity
```

This example shows how to enable the switch to send all traps to the host myhost.cisco.com using the community string public:

```
Switch(config)# snmp-server enable traps
Switch(config)# snmp-server host myhost.cisco.com public
```

This example shows how to associate a user with a remote host and to send **auth** (authNoPriv) authentication-level informs when you enter global configuration mode:

```
Switch(config)# snmp-server engineID remote 192.180.1.27 00000063000100a1c0b4011b
Switch(config)# snmp-server group authgroup v3 auth
Switch(config)# snmp-server user authuser authgroup remote 192.180.1.27 v3 auth md5
mypassword
Switch(config)# snmp-server user authuser authgroup v3 auth md5 mypassword
Switch(config)# snmp-server host 192.180.1.27 informs version 3 auth authuser config
Switch(config)# snmp-server enable traps
Switch(config)# snmp-server inform retries 0
```

Displaying SNMP Status

To display SNMP input and output statistics, including the number of illegal community string entries, errors, and requested variables, use the **show snmp** privileged EXEC command. You can also use the other privileged EXEC commands in [Table 75-4](#) to display SNMP information. For information about the fields in the output displays, see the *Cisco IOS Configuration Fundamentals Command Reference, Release 12.4*.

Table 75-4 *Commands for Displaying SNMP Information*

Feature	Default Setting
<code>show snmp</code>	Displays SNMP statistics.
<code>show snmp engineID</code>	Displays information on the local SNMP engine and all remote engines that have been configured on the device.
<code>show snmp group</code>	Displays information on each SNMP group on the network.
<code>show snmp pending</code>	Displays information on pending SNMP requests.
<code>show snmp sessions</code>	Displays information on the current SNMP sessions.
<code>show snmp user</code>	Displays information on each SNMP user name in the SNMP users table.

**Note**

You cannot use the **snmp-server enable informs** command. To enable the sending of SNMP inform notifications, use the **snmp-server enable traps** command combined with the **snmp-server host *host-addr* informs** command.



Configuring Flexible NetFlow



Note

Flexible NetFlow is supported only on Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, 7-E, and Catalyst 4500X.

Flow is defined as a unique set of key fields attributes, which might include fields of packet, packet routing attributes, and input and output interface information. A NetFlow feature defines a flow as a sequence of packets that have the same values for the feature key fields. Flexible NetFlow (FNF) allows you to collect and optionally export a flow record that specifies various flow attributes. NetFlow collection supports IP, IPv6 and Layer 2 traffic.



Note

This chapter provides Catalyst 4500 switch specific information. For more information, refer to the URL:

http://www.cisco.com/en/US/products/ps6965/products_ios_protocol_option_home.html

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).



Note

When IP routing is disabled, on the interface configured with NetFlow Lite, packets are not received on NetFlow collector. Enable IP routing for the NetFlow collector to work.

This chapter addresses both VSS and non-VSS environments:

- [VSS Environment, page 76-1](#)
- [Non-VSS Environment, page 76-8](#)

VSS Environment

The following items apply to a Catalyst 4500 series switch that belongs to a Virtual Switch System (VSS):

1. The Catalyst 4500 series switch supports ingress flow statistics collection for switched and routed packets; it does not support Flexible Netflow on egress traffic.

2. Each switch in an VSS has an independent NFE (Netflow Engine). This means that when there is ingress traffic on both the VSS Active and Standby switches, each is capable of creating flows for its ingress traffic
3. Configuration is performed on the VSS Active switch, which is synchronized to the VSS Standby switch.
4. Netflow **show** commands including Top Talkers, aggregate cache, and **clear** commands must be executed independently on VSS Active and Standby switch. The VSS Standby console will be available via remote console access from the VSS Active switch.
5. Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, 7-E, and Catalyst 4500X support a 100,000 entry hardware flow table. Both VSS Active and Standby switch have independent hardware flow tables of 100,000 entries. The hardware flow table is shared by all the flow monitors on a switch. To prevent one monitor from using all the flow table entries, the number of entries that it uses on a switch can be limited by the **cache entries number** command. This limit is per flow monitor, irrespective of the number of targets it is attached to.

The following example illustrates how to configure the flow monitor *m1* cache to hold 1000 entries. With this configuration, interface gig 1/3/1 (on the VSS Active) can create a maximum of 1000 flows and interface gig 2/3/2 (on the VSS Standby) can create a maximum of 1000 flows:

```
flow exporter e1
    ! exporter specifies where the flow records are send to
    destination 20.1.20.4
!
flow record r1
    ! record specifies packet fields to collect
    match ipv4 source address
    match ipv4 destination address
    collect counter bytes long
    collect counter packets long
    collect timestamp sys-uptime first
    collect timestamp sys-uptime last
!
flow monitor m1
    ! monitor refers record configuration and optionally exporter
    ! configuration. It specifies the cache size i.e. how many unique flow
    ! records to collect
    record r1
    exporter e1
    cache timeout active 60
    cache timeout inactive 30
    cache entries 1000

!interface GigabitEthernet 1/3/1
    ! layer2-switched allows collection of flow records even when the packet is
    ! bridged
    ip flow monitor m1 layer2-switched input
!
interface GigabitEthernet 2/3/2
    ip flow monitor m1 input
!
```

6. Catalyst 4500-E Series Switches with Sup-8E have two ASICs, and the ASIC that the software programs for a given flow monitor depends on what the flow monitor is attached to. When a datalink flow monitor is attached to an SSID (WLAN), the software programs the ASIC on the Daughter Card that creates flows only for pure Layer 2 packets (no IP header). By contrast, when a datalink flow monitor is attached to a port, or a port VLAN, or a VLAN for example, the software programs the ASIC (Netflow Engine) that creates flows for all packets.

7. If the flow exporter is configured to export packets through a specific VRF and a reload is initiated, it is possible that the flow data is not exported from the VSS standby switch. As a workaround, reconfigure the exporter.
8. Flow collection is supported on multiple targets (Port, VLAN, per-port per-VLAN (FNF can be enabled on a specific VLAN on a given port)) and on a port-channel (FNF is configured on the port-channel interface, rather than individual member ports). These targets can be on the VSS Active or on the VSS Standby. For example, if the target is a VLAN, it can consist of ports belonging to both switches. If there is ingress traffic in that VLAN on both switches, flows will be created in their independent flow caches. However, no Netflow configuration can be applied on the Virtual Switch Link (VSL) ports.
9. 64 unique flow record configurations are supported.
10. Flow QoS/UBRL and FNF cannot be configured on the same target. (For information on Flow-based QoS, see the section [Flow-based QoS, page 44-10.](#))
11. 14,000 unique IPv6 addresses can be monitored.
12. On a given target, one monitor per traffic type is allowed. However, you can configure multiple monitors on the same target for different traffic types.

For example, the following configuration is allowed:

```
! vlan config 10
  ip flow monitor <name> input
  ipv6 flow monitor <name> input
!
```

The following configuration is not allowed:

```
!
interface GigabitEthernet 3/1
  ip flow monitor m1 input
  ip flow monitor m2 input
```

13. On a given target monitoring Layer 2 and Layer 3, simultaneous traffic is not supported:

```
interface channel-group 1
  datalink flow monitor m1 input
  ip flow monitor m2 input
```

!

14. Selection of Layer 2 and Layer 3 packet fields in a single flow record definition is not allowed. However, ingress 802.1Q VLAN Id of packet and Layer 3 packet field selection is allowed.
15. To attach a monitor to port or port-vlan targets, a flow record matching on ingress 802.1Q VLANId key field, must match on input interface also as key field.



Note

The **match datalink dot1q vlan input** option is unavailable prior to IOS Release XE 3.3.0; you would only see the **input** option starting with the IOS Release XE 3.3.0.

16. Flow monitor matching on ingress 802.1Q VLANId as key field cannot be attached on a VNET trunk port target.
17. Only permanent and normal flow cache types are supported.
18. Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, 7-E, and Catalyst 4500X do not support:
 - predefined records like traditional routers (**record netflow ipv4 original-input**)
 - flow-based sampler.

19. On VLAN interfaces, when you use the **interface** option with the **Cos**, **Tos**, **TTL** or **Packet length** options, the system displays inaccurate results for the interface input field.
20. The VSS active and standby switches independently export flows, to the same or different Netflow collectors depending on flow exporter configuration. An IP route to the Netflow collector must exist and it should be reachable from the VSS for flow export.
21. At the collector, the flow sequence numbers are local to a switch and will be monotonically increasing for each member of VSS. Additionally, the `SourceId` field of the v9 export packet uniquely identifies the VSS switch number that it was exported from.
22. The configuration of the flow exporter does not support the option **output features**.
23. Maximum number of VRFs that can be used for the flow exporter destination address configuration in VSS is 5. This limit includes the Global Routing Table and is common across all flow exporters in the VSS.

For example, when the user tries to configure an exporter destination address using a sixth VRF limit is exceeded, the following warning is displayed:

```
flow exporter e10
      destination 20.1.20.4 vrf blue
%%Warning - Netflow exporter on Cat4k VSS switch cannot exceed a total max of 5 vrfs
used for destination address
configuration. Flow exporter e10 cannot export in vrf blue.
```

24. Flow aging in flow cache is controlled through active and in-active timer configuration. The minimum for active and in-active aging timers is 5 seconds. The timers must be in units of 5 seconds.



Note Flows in the hardware table are deleted after 5 seconds of in-activity irrespective of the active or in-active timer configuration values. This allows you to create new hardware flows quickly.

25. First and Last-seen flow timestamp accuracy is within 3 seconds.
26. 2048 Flow monitors and records are supported.

When TTL is configured as a flow field, the following values are reported for a given packet TTL value. [Table 76-1](#) lists the packet TTL and reported values.
27. Cisco TrustSec (CTS) fields are supported. These fields use Netflow collector to monitor and troubleshoot the CTS network, and to segregate traffic based on source group tag (SGT) values.
 - When configuring the source group tag (**collect flow cts source group-tag**), note the following:

The system copies the packets to software before it retrieves the CTS field. A large number of flows mean that a large number of packets are copied to the software, possibly affecting CPU performance.

The maximum number of (unique) hosts allowed in the switch (IP addresses) is 12,000.

In case of burst packets, the software may not be able to retrieve the CTS field because the software queue is throttled.
 - When configuring the destination group tag (**collect flow cts destination group-tag**), note that this CTS field value is collected only if you have already configured an IP-to-SGT mapping.
 - When configuring switch-derived source group tags (**collect flow cts switch derived-sgt**), note that the switch derives this value locally.
 - When configuring CTS fields on Supervisor Engine 8-E and 9-E, note that CTS fields are not supported on wireless interfaces (WLAN) and SSID.

Table 76-1 TTL Map: TTL Configured

Packet TT Value	Reported Value
0	0
1	1
2-10	10
11-25	25
26-50	50
51-100	100
100-150	150
150-255	255

- When packet length is configured as a flow field, the following values are reported for a given packet length value. [Table 76-2](#) lists the packet length and reported values.

Table 76-2 Packet Length Map: Packet Length Configured

Packet Length	Reported Value
0-64	64
65-128	128
129-256	256
257-512	512
513-756	756
757-1500	1500
1500-4000	4000
4000+	8192

The following table lists the options available through FNF and the supported fields.

Table 76-3 Options Available through FNF and the Supported Fields

Field	Description	Comments
Data Link Fields (Layer 2 Flow Label + A94)		
dot1q priority	802.1Q user	
dot1q vlan	802.1Q VLAN ID	Ingress VLAN is supported as key field.
mac destination-address	Upstream destination MAC address	
mac source-address	Down stream source MAC address	
IPv4 Fields		
destination address	IPv4 destination address	Yes

Table 76-3 Options Available through FNF and the Supported Fields

Field	Description	Comments
DSCP	IPv4 DSCP (part of TOS)	
fragmentation flags	IPv4 fragmentation flags	Supported as a non key field. DF flag is not supported
is-multicast	Indicator of an IPv4 multicast packet (0 - if it's not, 1 - if it is)	Supported as a non-key field.
Precedence	IPv4 precedence	
Protocol	IPv4 protocol	
source address	IPv4 source address	
total length	IPv4 datagram	Values are reported based on Table 76-2 .
Total length minimum	Minimum packet size seen	
Total length maximum	Maximum packet size seen	
Tos	IPv4 Type of Service (TOS)	
ttl	Pv4 Time to Live (TTL)	Values are reported based on Table 76-1 .
ttl minimum		Supported as a non-key field.
ttl maximum		Supported as a non-key field.
CTS Fields		
flow cts destination group-tag		Supported as a non-key field; configuring the IPv4 destination address is a prerequisite to using this field.
flow cts source group-tag		Supported as a non-key field; configuring the IPv4 source address is a prerequisite to using this field.
flow cts switch derived-sgt	Switch-derived source group-tag	Supported as a non-key field; configuring the IPv4 source address is a prerequisite to using this field.
IPv6 Fields		
destination address	IPv6 destination address	
dscp	IPv6 DSCP (part of IPv6 traffic class)	
flow-label	IPv6 flow label	

Table 76-3 Options Available through FNF and the Supported Fields

Field	Description	Comments
is-multicast	Indicator of an IPv6 multicast packet (0 - if it's not, 1 - if it is)	Supported as a non-key field
hop-limit	IPv6 hop limit (replaces IPv4 ttl)	Values are reported based on Table 76-1 .
hop-limit minimum	IPv6 minimum hop limit value seen in the flow.	Supported as a non-key field.
hop-limit maximum	IPv6 maximum hop limit value seen in the flow.	Supported as a non-key field.
next-header	IPv6 next header type	Only first next header is reported
total length	IPv6 total packet length	Values are based on Table 76-2 .
Total length minimum	Minimum packet size seen	
Total length maximum	Maximum packet size seen	
protocol	IPv6 next header type in the last IPv6 extension header	
source address	IPv6 source address	
traffic-class	IPv6 traffic class	Yes
Routing Attributes		
forwarding-status	Forwarding status for the packet (forwarded, terminated in the router, dropped by ACL, RPF, CAR)	Supported as a non-key field
Layer 4 Header Fields		
Field	Description	Comments
TCP Header Fields		
destination-port TCP destination number	TCP destination port	
flags [ack] [fin] [psh] [rst] [syn] [urg]	TCP flags.	Supported as non-key fields.
source-port	TCP source port	
UDP Header Fields		
destination-port	UDP destination port	
source-port	UDP source port	
ICMP Header Fields		
code	ICMP code	

Table 76-3 Options Available through FNF and the Supported Fields

Field	Description	Comments
type	ICMP type	
IGMP Header Fields		
type	IGMP	
Interface Fields		
input	Input interface index	
output	Input interface index	Output interface can be supported only as non-key.
Flexible NetFlow feature related fields		
direction: input		
Counter Fields		
bytes	32 bit counters	
bytes long	64 bit counter	
packets	32 bit counters	
packets long	64 bit counter of the packets in the flow	
Timestamp		
first seen	Time-stamp of the first packet that is accounted in the flow (in milliseconds, starting from the router boot-up)	3 sec accuracy
last seen	Time-stamp of the last packet that is accounted in the flow (in milliseconds, starting from the router boot-up)	3 sec accuracy

Configuring Flow Monitor Cache Values

Setting active cache timeout to a small value may cause the flows to be exported more frequently to the remote collector. This also causes software to delete flows from the local cache after exporting. So, cache statistics reported by switch may not display the actual flows being monitored.

Non-VSS Environment

The following items apply to the Catalyst 4500 series switch:

The Catalyst 4500 series switch supports ingress flow statistics collection for switched and routed packets; it does not support Flexible Netflow on egress traffic.

1. Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, and 7-E, and Catalyst 4500X support a 100,000 entry hardware flow table. The hardware flow table is shared by all the flow monitors on a switch. To prevent one monitor from using all the flow table entries, the number of entries that it uses on a switch can be limited by the **cache entries number** command. This limit is per flow monitor, irrespective of the number of targets it is attached to.

The following example illustrates how to configure the flow monitor *m1* cache to hold 1000 entries. With this configuration, interface gig 3/1 can create a maximum of 1000 flows and interface gig 3/2 can create a maximum of 1000 flows:

```
flow exporter e1
    ! exporter specifies where the flow records are sent to
    destination 20.1.20.4
!
flow record r1
    ! record specifies packet fields to collect
    match ipv4 source address
    match ipv4 destination address
    collect counter bytes long
    collect counter packets long
    collect timestamp sys-uptime first
    collect timestamp sys-uptime last
!
flow monitor m1
    ! monitor refers record configuration and optionally exporter
    ! configuration. It specifies the cache size i.e. how many unique flow
    ! records to collect
    record r1
    exporter e1
    cache timeout active 60
    cache timeout inactive 30
    cache entries 1000

!interface GigabitEthernet 3/1
    ! layer2-switched allows collection of flow records even when the packet is
    ! bridged
    ip flow monitor m1 layer2-switched input
!
interface GigabitEthernet 3/2
    ip flow monitor m1 input
!
```

2. Catalyst 4500-E Series Switches with Sup-8E have two ASICs, and the ASIC that the software programs for a given flow monitor depends on what the flow monitor is attached to. When a datalink flow monitor is attached to an SSID (WLAN), the software programs the ASIC on the Daughter Card that creates flows only for pure Layer 2 packets (no IP header). By contrast, when a datalink flow monitor is attached to a port, or a port VLAN, or a VLAN for example, the software programs the ASIC (Netflow Engine) that creates flows for all packets.
3. Flow collection is supported on multiple targets (Port, VLAN, per-port per-VLAN (FNF can be enabled on a specific VLAN on a given port)) and on a port-channel (FNF is configured on the port-channel interface, rather than individual member ports).
4. 64 unique flow record configurations are supported.
5. Flow QoS/UBRL and FNF cannot be configured on the same target. (For information on Flow-based QoS, see the section [Flow-based QoS, page 44-10.](#))
6. 14,000 unique IPv6 addresses can be monitored.

7. On a given target, one monitor per traffic type is allowed. However, you can configure multiple monitors on the same target for different traffic types.

For example, the following configuration is allowed:

```
! vlan config 10
  ip flow monitor <name> input
  ipv6 flow monitor <name> input
!
```

The following configuration is not allowed:

```
!
interface GigabitEthernet 3/1
  ip flow monitor m1 input
  ip flow monitor m2 input
```

8. On a given target monitoring Layer 2 and Layer 3, simultaneous traffic is not supported:

```
interface channel-group 1
  datalink flow monitor m1 input
  ip flow monitor m2 input
!
```

9. Selection of Layer 2 and Layer 3 packet fields in a single flow record definition is disallowed. However, ingress 802.1Q VLAN Id of packet and Layer 3 packet field selection is allowed.
10. To attach a monitor to port or port-vlan targets, a flow record matching on ingress 802.1Q VLAN Id as the key field, must also match on the input interface as the key field.



Note Flow monitor matching on ingress 802.1Q VLAN Id as the key field cannot be attached on a VNET trunk port target.

11. Only permanent and normal flow cache types are supported.
12. Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, 7-E, and Catalyst 4500X do not support:
- predefined records like traditional routers (**record netflow ipv4 original-input**)
 - flow based sampler.
13. On VLAN interfaces, when you use the **interface** option with the **Cos,Tos**, **TTL** or **Packet length** options, the system displays inaccurate results for the interface input field.
14. The configuration of the flow exporter does not support the option **output features**.
15. Flow aging in flow cache is controlled through active and in-active timer configuration. The minimum for active and in-active aging timers is 5 seconds. The timers must be in units of 5 seconds.



Note Flows in the hardware table are deleted after 5 seconds of in-activity irrespective of the active or in-active timer configuration values. This allows you to create new hardware flows quickly.

16. First and Last-seen flow timestamp accuracy is within 3 seconds.
17. 2048 Flow monitors and records are supported.
- When TTL is configured as a flow field, the following values are reported for a given packet TTL value. [Table 76-4](#) lists the packet TTL and reported values.
18. Cisco TrustSec (CTS) fields are supported. These fields use Netflow collector to monitor and troubleshoot the CTS network, and to segregate traffic based on source group tag (SGT) values.
- When configuring the source group tag (**collect flow cts source group-tag**), note the following:

The system copies the packets to software before it retrieves the CTS field. A large number of flows mean that a large number of packets are copied to the software, possibly affecting CPU performance.

The maximum number of (unique) hosts allowed in the switch (IP addresses) is 12,000.

In case of burst packets, the software may not be able to retrieve the CTS field because the software queue is throttled.

- When configuring the destination group tag (**collect flow cts destination group-tag**), note that this CTS field value is collected only if you have already configured an IP-to-SGT mapping.
- When configuring switch-derived source group tags (**collect flow cts switch derived-sgt**), note that the switch derives this value locally.
- When configuring CTS fields on Supervisor Engine 8-E and 9-E, note that CTS fields are not supported on wireless interfaces (WLAN) and SSID.

Table 76-4 TTL Map: TTL Configured

Packet TT Value	Reported Value
0	0
1	1
2-10	10
11-25	25
26-50	50
51-100	100
100-150	150
150-255	255

- When packet length is configured as a flow field, the following values are reported for a given packet length value. [Table 76-5](#) lists the packet length and reported values.

Table 76-5 Packet Length Map: Packet Length Configured

Packet Length	Reported Value
0-64	64
65-128	128
129-256	256
257-512	512
513-756	756
757-1500	1500
1500-4000	4000
4000+	8192

The following table lists the options available through FNF and the supported fields.

Table 76-6 Options Available through FNF and the Supported Fields

Field	Description	Comments
Data Link Fields (Layer 2 Flow Label + A94)		
dot1q priority	802.1Q user	
dot1q vlan	802.1Q VLAN ID	Ingress VLAN is supported as key field.
mac destination-address	Upstream destination MAC address	
mac source-address	Down stream source MAC address	
IPv4 Fields		
destination address	IPv4 destination address	Yes
DSCP	IPv4 DSCP (part of TOS)	
fragmentation flags	IPv4 fragmentation flags	Supported as a non-key field. DF flag is not supported
is-multicast	Indicator of an IPv4 multicast packet (0 - if it's not, 1 - if it is)	Supported as a non-key field.
Precedence	IPv4 precedence	
Protocol	IPv4 protocol	
source address	IPv4 source address	
total length	IPv4 datagram	Values are reported based on Table 76-5 .
Total length minimum	Minimum packet size seen	
Total length maximum	Maximum packet size seen	
Tos	IPv4 Type of Service (TOS)	
ttl	Pv4 Time to Live (TTL)	Values are reported based on Table 76-4 .
ttl minimum		Supported as a non-key field.
ttl maximum		Supported as a non-key field.
CTS Fields		

Table 76-6 Options Available through FNF and the Supported Fields

Field	Description	Comments
flow cts destination group-tag		Supported as a non-key field; configuring the IPv4 destination address is a prerequisite to using this field.
flow cts source group-tag		Supported as a non-key field; configuring the IPv4 source address is a prerequisite to using this field.
flow cts switch derived-sgt	Switch-derived source group-tag	Supported as a non-key field; configuring the IPv4 source address is a prerequisite to using this field.
IPv6 Fields		
destination address	IPv6 destination address	
dscp	IPv6 DSCP (part of IPv6 traffic class)	
flow-label	IPv6 flow label	
is-multicast	Indicator of an IPv6 multicast packet (0 - if it's not, 1 - if it is)	Supported as a non-key field
hop-limit	IPv6 hop limit (replaces IPv4 ttl)	Values are reported based on Table 76-4 .
hop-limit minimum	IPv6 minimum hop limit value seen in the flow.	Supported as a non-key field.
hop-limit maximum	IPv6 maximum hop limit value seen in the flow.	Supported as a non-key field.
next-header	IPv6 next header type	Only first next header is reported
total length	IPv6 total packet length	Values are based on Table 76-5 .
Total length minimum	Minimum packet size seen	
Total length maximum	Maximum packet size seen	
protocol	IPv6 next header type in the last IPv6 extension header	

Table 76-6 Options Available through FNF and the Supported Fields

Field	Description	Comments
source address	IPv6 source address	
traffic-class	IPv6 traffic class	Yes
Routing Attributes		
forwarding-status	Forwarding status for the packet (forwarded, terminated in the router, dropped by ACL, RPF, CAR)	Supported as a non-key field
Layer 4 Header Fields		
Field	Description	Comments
TCP Header Fields		
destination-port TCP destination number	TCP destination port	
flags [ack] [fin] [psh] [rst] [syn] [urg]	TCP flags.	Supported as non-key fields.
source-port	TCP source port	
UDP Header Fields		
destination-port	UDP destination port	
source-port	UDP source port	
ICMP Header Fields		
code	ICMP code	
type	ICMP type	
IGMP Header Fields		
type	IGMP	
Interface Fields		
input	Input interface index	
output	Output interface index	Output interface can be supported only as non-key.
Flexible NetFlow feature related fields		
direction: input		
Counter Fields		
bytes	32 bit counters	
bytes long	64 bit counter	
packets	32 bit counters	
packets long	64 bit counter of the packets in the flow	
Timestamp		

Table 76-6 *Options Available through FNF and the Supported Fields*

Field	Description	Comments
first seen	Time-stamp of the first packet that is accounted in the flow (in milliseconds, starting from the router boot-up)	3 sec accuracy
last seen	Time-stamp of the last packet that is accounted in the flow (in milliseconds, starting from the router boot-up)	3 sec accuracy

Configuring Flow Monitor Cache Values

Setting active cache timeout to a small value may cause the flows to be exported more frequently to the remote collector. This also causes software to delete flows from the local cache after exporting. So, cache statistics reported by switch may not display the actual flows being monitored.



Configuring Ethernet OAM and CFM

Ethernet Operations, Administration, and Maintenance (OAM) is a protocol for installing, monitoring, and troubleshooting Ethernet networks to increase management capability within the context of the overall Ethernet infrastructure. Starting with Cisco IOS Release 15.0(2)SG, the Catalyst 4500 series switch supports Standardized (Draft 8.1) IEEE 802.1ag Connectivity Fault Management (CFM) and IEEE 802.3ah Ethernet OAM discovery, link monitoring, remote fault detection, and remote loopback. It also supports IP Service Level Agreements (SLAs) for CFM, and ITU-T Y.1731 fault management. Ethernet OAM manager controls the interworking between CFM and 802.3ah OAM protocols.

This chapter provides information about configuring CFM and the Ethernet OAM protocol. It defines the differences between the ratified CFM 802.1ag standard (draft 8.1) and the previous version supported on the switch in Cisco IOS (draft 1.0). It also includes configuration information for CFM ITU-TY.1731 fault management support in this release.



Note

For complete command and configuration information for Ethernet OAM, CFM, and Y.1731, see the Cisco IOS Carrier Ethernet Configuration Guide at this URL:

<http://www.cisco.com/en/US/docs/ios-xml/ios/cether/configuration/15-mt/ce-15-mt-book.html>

For syntax of the commands used in this chapter, see The Cisco IOS Carrier Ethernet Command Reference at this URL:

<http://www.cisco.com/en/US/docs/ios-xml/ios/cether/command/ce-cr-book.html>

For complete command and configuration information for CFM, see the Cisco IOS feature module at this URL:

http://www.cisco.com/en/US/docs/ios/cether/configuration/guide/ce_cfm.html

The command reference for this release is at this URL:

[Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch.](#)

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

This chapter contains these sections:

- [About Ethernet CFM, page 77-2](#)
- [Configuring Ethernet CFM, page 77-6](#)
- [Understanding CFM ITU-T Y.1731 Fault Management, page 77-27](#)
- [Configuring Y.1731 Fault Management, page 77-29](#)
- [Managing and Displaying Ethernet CFM Information, page 77-31](#)

- [About Ethernet OAM Protocol, page 77-33](#)
- [Enabling and Configuring Ethernet OAM, page 77-35](#)
- [Displaying Ethernet OAM Protocol Information, page 77-49](#)
- [Ethernet CFM and Ethernet OAM Interaction, page 77-51](#)

About Ethernet CFM

Ethernet CFM is an end-to-end per-service-instance (per-VLAN) Ethernet layer OAM protocol. It includes proactive connectivity monitoring, fault verification, and fault isolation. End-to-end can be provider-edge-to-provider-edge (PE-to-PE) device or customer-edge-to-customer-edge (CE-to-CE) device. Ethernet CFM, as specified by IEEE 802.1ag, is the standard for Layer 2 ping, Layer 2 traceroute, and end-to-end connectivity verification of the Ethernet network.

These sections contain conceptual information about Ethernet CFM:

- [Ethernet CFM and OAM Definitions, page 77-2](#)
- [CFM Domain, page 77-3](#)
- [Maintenance Associations and Maintenance Points, page 77-4](#)
- [CFM Messages, page 77-5](#)
- [Crosscheck Function and Static Remote MEPs, page 77-5](#)
- [SNMP Traps and Fault Alarms, page 77-5](#)
- [Configuration Error List, page 77-6](#)
- [IP SLAs Support for CFM, page 77-6](#)

Ethernet CFM and OAM Definitions

The following table describes many of the terms in this chapter that are related to OAM and CFM features:

Term	Definition
CC	Continuity Check
CFM	Connectivity Fault Management
EI	Ethernet Infrastructure or EVC Infrastructure
EVC	Ethernet Virtual Circuit
MEP	Maintenance Endpoint
MIP	Maintenance Intermediate Point
OAM	Operations Administration and Maintenance
UNI	User to Network Interface

CFM Domain

A CFM maintenance domain is a management space on a network that is owned and operated by a single entity and defined by a set of internal boundary ports. You assign a unique maintenance level (from 0 to 7) to define the domain hierarchy. The larger the domain, the higher the level. For example, as shown in [Figure 77-1](#), a service-provider domain would be larger than an operator domain and might have a maintenance level of 6, while the operator domain maintenance level would be 3 or 4.

As shown in [Figure 77-2](#), domains cannot intersect or overlap because that would require management by more than one entity, which is not allowed. Domains can touch or nest (if the outer domain has a higher maintenance level than the nested domain). Nesting domains can be useful when a service provider contracts with one or more operators to provide Ethernet service. Each operator has its own maintenance domain and the service provider domain is a superset of the operator domains. Maintenance levels of nesting domains should be communicated among the administering organizations. CFM exchanges messages and performs operations on a per-domain basis.

Figure 77-1 CFM Maintenance Domains

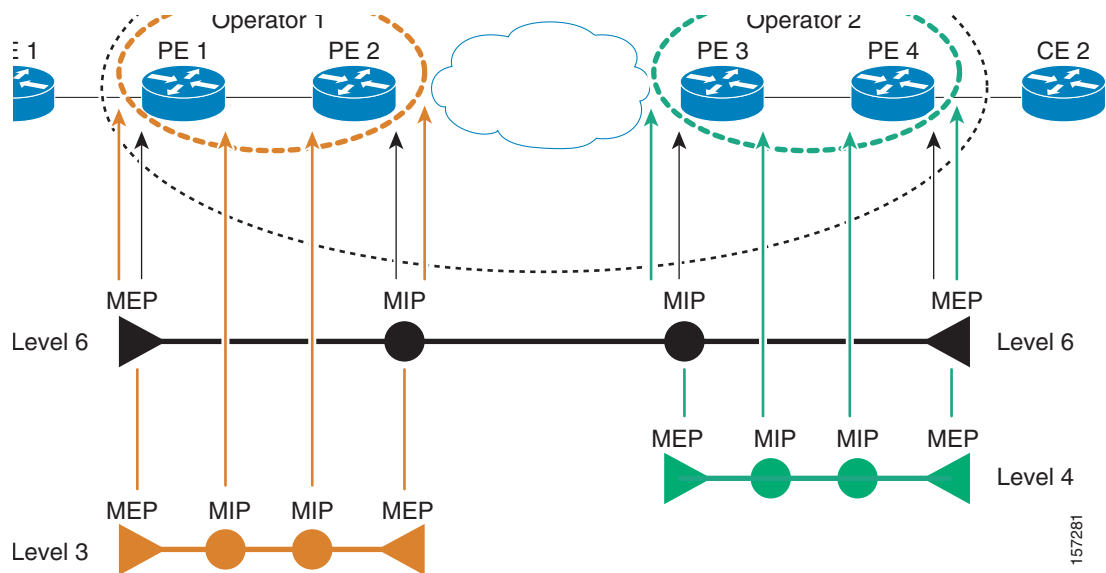


Figure 77-2 Allowed Domain Relationships

Scenario A:
Touching Domains OK

Scenario B:
Nested Domains OK

Scenario C:
Intersecting Domains
Not Allowed

157282

Maintenance Associations and Maintenance Points

A maintenance association (MA) identifies a service that can be uniquely identified within the maintenance domain. The CFM protocol runs within a maintenance association. A maintenance point is a demarcation point on an interface that participates in CFM within a maintenance domain. Maintenance points drop all lower-level frames and forward all higher-level frames. There are two types of maintenance points:

- Maintenance end points (MEPs) are points at the edge of the domain that define the boundaries and confine CFM messages within these boundaries. Outward facing or Down MEPs communicate through the wire side (connected to the port). Inward facing or Up MEPs communicate through the relay function side, not the wire side.



Note CFM draft 1 referred to inward and outward-facing MEPs. CFM draft 8.1 refers to up and down MEPs, respectively. This document uses the CFM 8.1 terminology for direction.

CFM draft 1 supported only up MEPs on a per-port or per-VLAN basis. CFM 802.1ag supports up and down per-VLAN MEPs, as well as port MEPs, which are untagged down MEPs that are not associated with a VLAN. Port MEPs are configured to protect a single hop and used to monitor link state through CFM. If a port MEP is not receiving continuity check messages from its peer (static remote MEP), for a specified interval, the port is put into an operational down state in which only CFM and OAM packets pass through, and all other data and control packets are dropped.

- An up MEP sends and receives CFM frames through the relay function. It drops all CFM frames at its level or lower that come from the wire side, except traffic going to the down MEP. For CFM frames from the relay side, it processes the frames at its level and drops frames at a lower level. The MEP transparently forwards all CFM frames at a higher level, regardless of whether they are received from the relay or wire side. If the port on which MEP is configured is blocked by STP, the MEP can still send or receive CFM messages through the relay function. CFM runs at the provider maintenance level (UPE-to-UPE), specifically with up MEPs at the user network interface (UNI).
- A down MEP sends and receives CFM frames through the wire connected to the port on which the MEP is configured. It drops all CFM frames at its level or lower that come from the relay side. For CFM frames from the wire side, it processes all CFM frames at its level and drops CFM frames at lower levels except traffic going to the other lower-level down MEP. The MEP transparently forwards all CFM frames at a higher level, regardless of whether they are received from the relay or through the wire
- Maintenance intermediate points (MIPs) are internal to a domain, not at the boundary, and respond to CFM only when triggered by traceroute and loopback messages. They forward CFM frames received from MEPs and other MIPs, drop all CFM frames at a lower level (unless MIP filtering is enabled), and forward all CFM frames at a higher level and at a lower level and regardless of whether they are received from the relay or wire side. When MIP filtering is enabled, the MIP drops CFM frames at a lower level. MIPs also catalog and forward continuity check messages (CCMs), but do not respond to them.

In the first draft of CFM, MIP filtering was always enabled. In draft 8.1, MIP filtering is disabled by default, and you can configure it to be enabled or disabled. When MIP filtering is disabled, all CFM frames are forwarded.

You can manually configure a MIP or configure the switch to automatically create a MIP. You can configure a MEP without a MIP. In case of a configuration conflict, manually created MIPs take precedence over automatically created MIPs.

If port on which the MEP is configured is blocked by Spanning-Tree Protocol (STP), the MIP can receive and might respond to CFM messages from both the wire and relay side, but cannot forward any CFM messages. This differs from CFM draft 1, where STP blocked ports could not send or receive CFM messages.

CFM Messages

CFM uses standard Ethernet frames distinguished by EtherType or (for multicast messages) by MAC address. All CFM messages are confined to a maintenance domain and to a service-provider VLAN (S-VLAN). These CFM messages are supported:

- **Continuity Check (CC) messages**—multicast heartbeat messages exchanged periodically between MEPs that allow MEPs to discover other MEPs within a domain and allow MIPs to discover MEPs. CC messages are configured to a domain or VLAN. Enter the **continuity-check Ethernet service** configuration command to enable CCM.

The default continuity check message (CCM) interval on the switch is 10 seconds. You can set it to be 100 ms, 1 second, 1 minute, or 10 minutes by entering the **continuity-check interval Ethernet service mode** command. Because faster CCM rates are more CPU intensive, we do not recommend configuring a large number of MEPs running at 100 ms intervals.

- **Loopback messages**—unicast or multicast frames transmitted by a MEP at administrator request to verify connectivity to a particular maintenance point, indicating if a destination is reachable. A loopback message is similar to an Internet Control Message Protocol (ICMP) ping message. Refer to the **ping ethernet** privileged EXEC command.
- **Traceroute messages**—multicast frames transmitted by a MEP at administrator request to track the path (hop-by-hop) to a destination MEP. Traceroute messages are similar in concept to UDP traceroute messages. Refer to the **traceroute ethernet** privileged EXEC command.

Crosscheck Function and Static Remote MEPs

The crosscheck function verifies a post-provisioning timer-driven service between dynamically configured MEPs (using crosscheck messages) and expected MEPs (by configuration) for a service. It verifies that all endpoints of a multipoint service are operational. The crosscheck function is performed only one time and is initiated from the command-line interface (CLI).

CFM 802.1ag also supports static remote MEPs or static RMEP check. Unlike the crosscheck function, which is performed only once, configured static RMEP checks run continuously. To configure static RMEP check, enter the **continuity-check static rmeip** Ethernet CFM service mode command.

SNMP Traps and Fault Alarms

The MEPs generate two types of SNMP traps: CC traps and crosscheck traps. Supported CC traps are MEP up, MEP down, cross-connect (a service ID does not match the VLAN), loop, and configuration error. The crosscheck traps are service up, MEP missing (an expected MEP is down), and unknown MEP.

Fault alarms are unsolicited notifications sent to alert the system administrator when CFM detects a fault. In CFM draft 1, fault alarms were sent instantaneously when detected. In CFM 802.1ag, you can configure the priority level of alarms that trigger an SNMP trap or syslog message. You can also configure a delay period before a fault alarm is sent and the time before the alarm is reset.

Configuration Error List

CFM configuration errors in CFM 802.1ag can be misconfigurations or extra configuration commands detected during MEP configuration. They can be caused by overlapping maintenance associations. For example, if you create a maintenance association with a VLAN list and a MEP on an interface, a potential leak error could occur if other maintenance associations associated with the same VLAN exist at a higher level without any MEPs configured. You can display the configuration error list, which is informational only, by entering the **show ethernet cfm errors** configuration privileged EXEC command.

IP SLAs Support for CFM

The switch supports CFM with IP Service Level Agreements (SLAs), which gathers Ethernet layer network performance metrics. Available statistical measurements for the IP SLAs CFM operation include round-trip time, jitter (interpacket delay variance), and packet loss. You can schedule multiple IP SLA operations and use Simple Network Management Protocol (SNMP) trap notifications and syslog messages to monitor threshold violations proactively.

IP SLA integration with CFM gathers Ethernet layer statistical measurements by sending and receiving Ethernet data frames between CFM MEPs. Performance is measured between the source MEP and the destination MEP. Unlike other IP SLA operations that provide performance metrics for only the IP layer, IP SLAs with CFM provide performance metrics for Layer 2.

You can manually configure individual Ethernet ping or jitter operations. You can also configure an IP SLA automatic Ethernet operation that queries the CFM database for all MEPs in a given maintenance domain and VLAN. The operation then automatically creates individual Ethernet ping or jitter operations based on the discovered MEPs.

Because IP SLAs is a Cisco proprietary feature, interoperability between CFM draft 1 and CFM 802.1ag is handled automatically by the switch.

For more information about IP SLA operation with CFM, see the *IP SLAs for Metro-Ethernet* feature module at this URL:

http://www.cisco.com/en/US/docs/ios/12_2sr/12_2srb/feature/guide/sr_meth.html

Configuring Ethernet CFM

CFM draft 8.1 on Catalyst 4500 series switch mandates that you enter the **ethernet cfm ieee** command before configuring any other CFM CLI. Without this command, no other CFM CLIs are applied. Configuring Ethernet CFM requires that you configure the CFM domain. You can optionally configure and enable other CFM features (such as crosschecking, static remote MEP, port MEPs, CVLAN MEPs/MIPs, SNMP traps, and fault alarms). Some of the configuration commands and procedures differ from those used in CFM draft 1. CLIs in draft 1 that have been changed are no longer available; they have been deprecated and are not allowed. Only the CLIs mentioned in the following sections are required for draft 8.1.



Note

Upgrading software from CFM draft 1 to draft 8.1 causes a switch to silently drop the draft 1 configuration on the draft 8.1 image. Also, no CFM stateful sync happens between draft 1 and draft 8.1 images. After the upgrade, all CFM configurations must be reconfigured according to the procedures mentioned for draft 8.1.

To configure Ethernet CFM you must prepare the network and configuring services. You can optionally configure and enable crosschecking. These sections are included:

- [Ethernet CFM Default Configuration, page 77-7](#)
- [Ethernet CFM Configuration Guidelines, page 77-7](#)
- [Configuring the CFM Domain, page 77-8](#)
- [Configuring Ethernet CFM Crosscheck, page 77-11](#)
- [Configuring Static Remote MEP, page 77-13](#)
- [Configuring a Port MEP, page 77-14](#)
- [Configuring SNMP Traps, page 77-16](#)
- [Configuring Fault Alarms, page 77-16](#)
- [Configuring IP SLAs CFM Operation, page 77-18](#)
- [Configuring CFM on C-VLAN \(Inner VLAN\), page 77-24](#)

Ethernet CFM Default Configuration

CFM is globally disabled.

CFM is enabled on all interfaces when CFM is globally enabled.

A port can be configured as a flow point (MIP/MEP), a transparent port, or disabled (CFM disabled). By default, ports are transparent ports until configured as MEP, MIP, or disabled.

There are no MEPs or MIPs configured.

When configuring a MA, if you do not configure direction, the default is up (inward facing).

Ethernet CFM Configuration Guidelines

When configuring Ethernet CFM, consider these guidelines and restrictions:

- You must enter the **ethernet cfm ieee** global configuration command before configuring any other CFM CLI. If not, all other CFM CLIs are not applied.
- CFM is not supported on and cannot be configured on either routed ports or Layer 3 EtherChannels.
- You can configure a Layer 2 EtherChannel port channel as Up MEP, Down MEP, or MIP. However, such configurations are not supported on individual ports that belong to an EtherChannel. You cannot add a port with this configuration to an EtherChannel group.
- Port MEP is not supported and cannot be configured on Layer 2 EtherChannels.
- CFM is not supported and cannot be configured on VLAN interfaces.
- On isolated host, community host, or a promiscuous access port, only Down MEP is supported on isolated, community and primary VLANs, respectively.
- Up MEP is supported only on regular VLANs on PVLAN trunks. Down MEP is supported on regular VLANs as well as isolated VLANs on PVLAN secondary trunks. Similarly, Down MEP is supported on regular VLANs as well as primary VLANs on promiscuous trunk ports.
- The CFM service on a PVLAN ends at the PVLAN port. The translation of CFM service between PVLANs is not supported between the PVLAN ports.

- CFM Unicast packets (Loopback Messages and Traceroute Reply), are not allowed on Down MEP on STP blocked ports. The blocked port cannot respond to ping and traceroute. You must configure a port MEP at a lower level than any service (VLAN) MEPs on an interface.
- An 802.1Q (QinQ) tunnel port can be an Up MEP or a port MEP.
- A QinQ port cannot be a Down MEP or a MIP; you can configure the port as a MIP, but it is not active or visible in traceroute. Port MEP frames received on a QinQ interface are not tunneled and are processed locally.
- CFM on a C-VLAN is supported on Traditional and Selective QinQ and not supported on One-to-One VLAN Mapping on Trunk ports.
- Do not configure a port with tunnel mode using the native VLAN as the S-VLAN or the C-VLAN.
- For port MEP on a QinQ port, do not enter the **vlan dot1q tag native** global configuration command to enable tagging on native VLAN frames.
- If you are running CFM draft 1 and upgrade to a software version that supports CFM draft 8.1, the switch silently drops the draft 1 configuration on draft 8.1 image. Some of the CLIs have been changed from draft1 to draft8.1. You need to reconfigure all the required configurations on draft 8.1 image.

Configuring the CFM Domain

To configure the Ethernet CFM domain, configure a service to connect the domain to a VLAN, or configure a port to act as a MEP, perform this task. You can also enter the optional commands to configure other parameters, such as continuity checks.

	Command	Purpose
Step 1	<code>configure terminal</code>	Enter global configuration mode.
Step 2	<code>ethernet cfm ieee</code>	A must have configuration for draft 8.1. This is required to be configured before any other configuration.
Step 3	<code>ethernet cfm global</code>	Globally enable Ethernet CFM on the switch.
Step 4	<code>ethernet cfm traceroute cache [size entries / hold-time minutes]</code>	(Optional) Configure the CFM traceroute cache. You can set a maximum cache size or hold time. <ul style="list-style-type: none"> • (Optional) For size, enter the cache size in number of entry lines. The range is from 1 to 4095; the default is 100 lines. • (Optional) For hold-time, enter the maximum cache hold time in minutes. The range is from 1 to 65535; the default is 100 minutes.
Step 5	<code>ethernet cfm mip auto-create level level-id vlan vlan-id</code>	(Optional) Configure the switch to automatically create MIPs for VLAN IDS that are not associated with specific maintenance associations at the specified level. The level range is 0 to 7. Note Configure MIP auto-creation only for VLANs that MIPs should monitor. Configuring for all VLANs can be CPU and memory-intensive.
Step 6	<code>ethernet cfm mip filter</code>	(Optional) Enable MIP filtering, which means that all CFM frames at a lower level are dropped. The default is disabled.

	Command	Purpose
Step 7	ethernet cfm domain <i>domain-name</i> level <i>level-id</i>	Define a CFM domain, set the domain level, and enter ethernet-cfm configuration mode for the domain. The maintenance level number range is 0 to 7.
Step 8	id { <i>mac-address domain_number</i> dns <i>name</i> null }	(Optional) Assign a maintenance domain identifier. <ul style="list-style-type: none"> <i>mac-address domain_number</i>—Enter the MAC address and a domain number. The number can be from 0 to 65535. dns <i>name</i>—Enter a DNS name string. The name can be a maximum of 43 characters. null—Assign no domain name.
Step 9	service { <i>ma-name</i> <i>ma-number</i> <i>vpn-id vpn</i> } { vlan <i>vlan-id</i> [direction down] port }	Define a customer service maintenance association (MA) name or number or VPN ID to be associated with the domain, a VLAN ID or port MEP, and enter ethernet-cfm-service configuration mode. <ul style="list-style-type: none"> <i>ma-name</i>—a string of no more than 100 characters that identifies the MAID. <i>ma-number</i>—a value from 0 to 65535. <i>vpn-id vpn</i>—enter a VPN ID as the <i>ma-name</i>. vlan <i>vlan-id</i>—VLAN range is from 1 to 4094. You cannot use the same VLAN ID for more than one domain at the same level. (Optional) direction down—specify the service direction as down. port—Configure port MEP, a down MEP that is untagged and not associated with a VLAN.
Step 10	continuity-check	Enable sending and receiving of continuity check messages.
Step 11	continuity-check interval <i>value</i>	(Optional) Set the interval at which continuity check messages are sent. The available values are 100 ms, 1 second, 10 seconds, 1 minute and 10 minutes. The default is 10 seconds. Note Because faster CCM rates are more CPU-intensive, we do not recommend configuring a large number of MEPs running at 100 ms intervals.
Step 12	continuity-check loss-threshold <i>threshold-value</i>	(Optional) Set the number of continuity check messages to be missed before declaring that an MEP is down. The range is 2 to 255; the default is 3.
Step 13	maximum meps <i>value</i>	(Optional) Configure the maximum number of MEPs allowed across the network. The range is from 1 to 65535. The default is 100.

	Command	Purpose
Step 14	<code>sender-id {chassis none}</code>	(Optional) Include the sender ID TLVs, attributes containing type, length, and values for neighbor devices. <ul style="list-style-type: none"> chassis—Send the chassis ID (host name). none—Do not include information in the sender ID.
Step 15	<code>mip auto-create [lower-mep-only none]</code>	(Optional) Configure auto creation of MIPs for the service. <ul style="list-style-type: none"> lower-mep-only—Create a MIP only if there is a MEP for the service in another domain at the next lower active level. none—No MIP auto-create.
Step 16	<code>exit</code>	Return to ethernet-cfm configuration mode.
Step 17	<code>mip auto-create [lower-mep-only]</code>	(Optional) Configure auto creation of MIPs for the domain. <ul style="list-style-type: none"> lower-mep-only—Create a MIP only if there is a MEP for the service in another domain at the next lower active level.
Step 18	<code>mep archive-hold-time minutes</code>	(Optional) Set the number of minutes that data from a missing maintenance end point is kept before it is purged. The range is 1 to 65535; the default is 100 minutes.
Step 19	<code>exit</code>	Return to global configuration mode.
Step 20	<code>interface interface-id</code>	Specify an interface to configure, and enter interface configuration mode.
Step 21	<code>switchport mode trunk</code>	(Optional) Configure the port as a trunk port.
Step 22	<code>ethernet cfm mip level level-id</code>	(Optional) Configure a customer level or service-provider level maintenance intermediate point (MIP) for the interface. The MIP level range is 0 to 7. Note This step is not required if you have entered the ethernet cfm mip auto-create global configuration command or the mip auto-create ethernet-cfm or ethernet-cfm-srv configuration mode.
Step 23	<code>ethernet cfm mep domain domain-name mpid identifier {vlan vlan-id port}</code>	Configure maintenance end points for the domain, and enter ethernet cfm mep mode. <ul style="list-style-type: none"> domain domain-name—Specify the name of the created domain. mpid identifier—Enter a maintenance end point identifier. The identifier must be unique for each VLAN (service instance). The range is 1 to 8191. vlan vlan-id—Enter the service provider VLAN ID or IDs as a VLAN-ID (1 to 4094), a range of VLAN-IDs separated by a hyphen, or a series of VLAN IDs separated by comma. port—Configure port MEP.
Step 24	<code>cos value</code>	(Optional) Specify the class of service (CoS) value to be sent with the messages. The range is 0 to 7.

	Command	Purpose
Step 25	end	Return to privileged EXEC mode.
Step 26	show ethernet cfm maintenance-points {local remote}	Verify the configuration.
Step 27	show ethernet cfm errors [configuration]	(Optional) Display the configuration error list.
Step 28	copy running-config startup-config	(Optional) Save your entries in the configuration file.

Use the **no** versions of the commands to remove the configuration or return to the default configurations.

This is an example of the basic CFM configuration:

```
Switch(config)# ethernet cfm ieee
Switch(config)# ethernet cfm global
Switch(config)# ethernet cfm domain abc level 3
Switch(config-ecfm)# service test vlan 5
Switch(config-ecfm-srv)# continuity-check
Switch(config-ecfm-srv)# exit
Switch(config-ecfm)# exit
Switch(config)# interface gigabitethernet1/0/2
Switch(config-if)# ethernet cfm mep domain abc mpid 222 vlan 5
Switch(config-if-ecfm-mep)# exit
```

Configuring Ethernet CFM Crosscheck

To configure Ethernet CFM crosscheck, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# ethernet cfm mep crosscheck start-delay delay	Configures the number of seconds that the device waits for remote MEPs to come up before the crosscheck is started. The range is 1 to 65535; the default is 30 seconds.
Step 3	Switch(config)# ethernet cfm domain domain-name level level-id	Defines a CFM domain, set the domain level, and enter ethernet-cfm configuration mode for the domain. The maintenance level number range is 0 to 7.
Step 4	Switch(config)# service {ma-name ma-number vpn-id vpn} { vlan vlan-id}	Define a customer service maintenance association name or number or VPN ID to be associated with the domain, and a VLAN ID, and enter ethernet-cfm-service configuration mode. <ul style="list-style-type: none"> <i>ma-name</i>—A string of no more than 100 characters that identifies the MAID. <i>ma-number</i>—A value from 0 to 65535. <i>vpn-id vpn</i>—Enter a VPN ID as the ma-name. <i>vlan vlan-id</i>—VLAN range is from 1 to 4094. You cannot use the same VLAN ID for more than one domain at the same level.
Step 5	Switch(config-ether-cfm)# mep mpid identifier	Define the MEP maintenance end point identifier in the domain and service. The range is 1 to 8191.
Step 6	Switch(config)# end	Returns to privileged EXEC mode.

	Command	Purpose
Step 7	Switch# ethernet cfm mep crosscheck {enable disable} domain <i>domain-name</i> {vlan { <i>vlan-id</i> any} port}	<ul style="list-style-type: none"> • Enable or disable CFM crosscheck for one or more VLANs or a port MEP in the domain. • domain <i>domain-name</i>—Specify the name of the created domain. • vlan {<i>vlan-id</i> any}—Enter the service provider VLAN ID or IDs as a VLAN-ID (1 to 4094), a range of VLAN-IDs separated by a hyphen, or a series of VLAN IDs separated by comma. Enter any for any VLAN. • port—Identify a port MEP
Step 8	Switch# show ethernet cfm maintenance-points remote crosscheck	Verifies the configuration.
Step 9	Switch# show ethernet cfm errors [configuration]	Enter this command after you enable CFM crosscheck to display the results of the crosscheck operation. Enter the configuration keyword to display the configuration error list.
Step 10	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Use the **no** form of each command to remove a configuration or to return to the default settings.

The following example illustrates how to configure Ethernet CFM crosscheck:

```
Switch(config)# ethernet cfm mep crosscheck start-delay 60
Switch(config)# ethernet cfm domain abc level 3
Switch(config-ecfm)# service test vlan 5
Switch(config-ecfm-srv)# mep mpid 23
Switch(config-ecfm-srv)# mep mpid 34
Switch(config-ecfm-srv)# end
Switch# ethernet cfm mep crosscheck enable domain abc vlan 5
```

```
Switch# show ethernet cfm maintenance-points remote crosscheck
```

```
-----
MPID Domain Name                               Lvl Type Id      Mep-Up
  MA Name
-----
    23 abc                                     3 Vlan 5         No
      test
    34 abc                                     3 Vlan 5         No
      test
```

```
Switch# show ethernet cfm errors
```

```
-----
MPID Domain Id                               Mac Address      Type Id
  MA Name                               Reason          Lvl  Age
-----
    34 abc                               0000.0000.0000   Vlan 5
      test                               RMEP missing    3    95s
    23 abc                               0000.0000.0000   Vlan 5
      test                               RMEP missing    3    95s
Switch#
```

Configuring Static Remote MEP

To configure Ethernet CFM static remote MEP, perform this task:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enter global configuration mode.
Step 2	<code>ethernet cfm domain domain-name level level-id</code>	Define a CFM domain, set the domain level, and enter ethernet-cfm configuration mode for the domain. The maintenance level number range is 0 to 7.
Step 3	<code>service {ma-name / ma-number / vpn-id vpn} {vlan vlan-id [direction down] port}</code>	Define a customer service maintenance association name or number or a VPN ID to be associated with the domain, and a VLAN ID or peer MEP, and enter ethernet-cfm-service configuration mode. <ul style="list-style-type: none"> <i>ma-name</i>—a string of no more than 100 characters that identifies the MAID. <i>ma-number</i>—a value from 0 to 65535. <i>vpn-id</i>—enter a VPN ID as the <i>ma-name</i>. vlan vlan-id—VLAN range is from 1 to 4094. You cannot use the same VLAN ID for more than one domain at the same level. (Optional) direction down—specify the service direction as down. port—Configure port MEP, a down MEP that is untagged and not associated with a VLAN.
Step 4	<code>continuity-check</code>	Enable sending and receiving of continuity check messages.
Step 5	<code>mep mpid identifier</code>	Define the static remote maintenance end point identifier. The range is 1 to 8191.
Step 6	<code>continuity-check static rmp</code>	Enable checking of the incoming continuity check message from a remote MEP that is configured in the MEP list.
Step 7	<code>end</code>	Return to privileged EXEC mode.
Step 8	<code>show ethernet cfm maintenance-points remote static</code>	Verify the configuration.
Step 9	<code>show ethernet cfm errors [configuration]</code>	Enter this command after you enable CFM crosscheck to display the results of the crosscheck operation. Enter the configuration keyword to display the configuration error list.
Step 10	<code>copy running-config startup-config</code>	(Optional) Save your entries in the configuration file.

Use the **no** form of each command to remove a configuration or to return to the default settings.

The following example illustrates how to configure Ethernet CFM static remote MEP:

```
Switch(config)# ethernet cfm domain abc level 3
Switch(config-ecfm)# service test vlan 5
Switch(config-ecfm-srv)# continuity-check
Switch(config-ecfm-srv)# mep mpid 23
```

```

Switch(config-ecfm-srv) # mep mpid 34
Switch(config-ecfm-srv) # continuity-check static rmep

Switch# show ethernet cfm maintenance-points remote static
-----
MPID Domain Name                               Lvl Type Id      Mep-Up
  MA Name
-----
    23 abc                                     3 Vlan 5          No
      test
    34 abc                                     3 Vlan 5          No
      test
Switch# show ethernet cfm errors
-----
MPID Domain Id                               Mac Address      Type Id
  MA Name                               Reason          Lvl  Age
-----
    34 abc                               0000.0000.0000   Vlan 5
      test                               RMEP missing     3    421s
    23 abc                               0000.0000.0000   Vlan 5
      test                               RMEP missing     3    421s
Switch#

```

Configuring a Port MEP

A port MEP is a down MEP that is not associated with a VLAN and that uses untagged frames to carry CFM messages. You configure port MEPs on two connected interfaces. Port MEPs are always configured at a lower domain level than native VLAN MEPs.

To configure Ethernet CFM port MEPs, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# ethernet cfm domain <i>domain-name level level-id</i>	Defines a CFM domain, set the domain level, and enter ethernet-cfm configuration mode for the domain. The maintenance level number range is 0 to 7.
Step 3	Switch(config-ecfm)# service { <i>ma-name</i> / <i>ma-number</i> / <i>vpn-id</i> } port	Defines a customer service maintenance association name or number or VPN ID to be associated with the domain, define a port MEP, and enter ethernet-cfm-service configuration mode. <ul style="list-style-type: none"> • <i>ma-name</i>—a string of no more than 100 characters that identifies the MAID. • <i>ma-number</i>—a value from 0 to 65535. • <i>vpn-id vpn</i>—enter a VPN ID as the <i>ma-name</i>.
Step 4	Switch(config-ecfm-srv)# mep mpid <i>identifier</i>	Defines the static remote maintenance end point identifier in the domain and service. The range is 1 to 8191.
Step 5	Switch(config-ecfm-srv)# continuity-check	Enables sending and receiving of continuity check messages.

	Command	Purpose
Step 6	Switch(config-ecfm-srv)# continuity-check interval <i>value</i>	(Optional) Sets the interval at which continuity check messages are sent. The available values are 1 second, 10 seconds, 1 minute, and 10 minutes. The default is 10 seconds. Note Because faster CCM rates are more CPU-intensive, we do not recommend configuring a large number of MEPs running at 1 s intervals.
Step 7	Switch(config-ecfm-srv)# continuity-check loss-threshold <i>threshold-value</i>	(Optional) Sets the number of continuity check messages to be missed before declaring that an MEP is down. The range is 2 to 255; the default is 3.
Step 8	Switch(config-ecfm-srv)# continuity-check static rmep	Enables checking of the incoming continuity check message from a remote MEP that is configured in the MEP list.
Step 9	Switch(config-ecfm-srv)# exit	Returns to ethernet-cfm configuration mode.
Step 10	Switch(config-ecfm)# exit	Returns to global configuration mode.
Step 11	Switch(config)# interface <i>interface-id</i>	Identifies the port MEP interface and enter interface configuration mode.
Step 12	Switch(config-if)# ethernet cfm mep domain <i>domain-name</i> mpid identifier port	Configures the interface as a port MEP for the domain. <ul style="list-style-type: none"> domain <i>domain-name</i>—Specify the name of the created domain. mpid identifier—Enter a maintenance end point identifier. The identifier must be unique for each VLAN (service instance). The range is 1 to 8191.
Step 13	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 14	Switch)# show ethernet cfm maintenance-points remote static	Verifies the configuration.
Step 15	Switch)# show ethernet cfm errors [<i>configuration</i>]	Enters this command after you enable CFM crosscheck to display the results of the crosscheck operation. Enter the configuration keyword to display the configuration error list.
Step 16	Switch)# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Use the **no** form of each command to remove a configuration or to return to the default settings.

This is a sample configuration for a port MEP:

```
Switch(config)# ethernet cfm domain abc level 3
Switch(config-ecfm)# service PORTMEP port
Switch(config-ecfm-srv)# mep mpid 222
Switch(config-ecfm-srv)# continuity-check
Switch(config-ecfm-srv)# continuity-check static rmep
Switch(config-ecfm-srv)# exit
Switch(config-ecfm)# exit
Switch(config)# interface gigabitethernet1/0/1
Switch(config-if)# ethernet cfm mep domain abc mpid 111 port
Switch(config-if)# end
```

Configuring SNMP Traps

To configure traps for Ethernet CFM, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# snmp-server enable traps ethernet cfm cc [mep-up] [mep-down] [config] [loop] [cross-connect]	(Optional) Enables Ethernet CFM continuity check traps.
Step 3	Switch(config)# snmp-server enable traps ethernet cfmalarm	(Optional) Enables Ethernet CFM fault alarm trap.
Step 4	Switch(config)# snmp-server enable traps ethernet cfm crosscheck [mep-unknown] [mep-missing] [service-up]	(Optional) Enable s Ethernet CFM crosscheck traps.
Step 5	Switch(config)# end	Returns to privileged EXEC mode.
Step 6	Switch# show running-config	Verifies your entries.
Step 7	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Use the **no** form of each command to remove a configuration or to return to the default settings.

The following example illustrates how to configure SNMP traps:

```
Switch(config)# snmp-server enable traps ethernet cfm alarm
Switch(config)# snmp-server enable traps ethernet cfm cc mep-down
Switch(config)# snmp-server enable traps ethernet cfm crosscheck mep-missing
```

Configuring Fault Alarms

To configure Ethernet CFM fault alarms, perform this task.



Note

You can configure fault alarms in either global configuration or Ethernet CFM interface MEP mode. When a conflict exists, the interface MEP mode configuration takes precedence.

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>ethernet cfm alarm notification {all error-xcon mac-remote-error-xcon none remote-error-xcon xcon}</code>	<p>Globally enables Ethernet CFM fault alarm notification for the specified defects:</p> <ul style="list-style-type: none"> • all—report all defects. • error-xcon—Report only error and connection defects. • mac-remote-error-xcon—Report only MAC-address, remote, error, and connection defects. • none—Report no defects. • remote-error-xcon—Report only remote, error, and connection defects. • xcon—Report only connection defects.
Step 3	<code>ethernet cfm alarm delay value</code>	(Optional) Sets a delay period before a CFM fault alarm is sent. The range is 2500 to 10000 milliseconds (ms). The default is 2500 ms.
Step 4	<code>ethernet cfm alarm reset value</code>	(Optional) Specifies the time period before the CFM fault alarm is reset. The range is 2500 to 10000 milliseconds (ms). The default is 10000 ms.
Step 5	<code>ethernet cfm logging alarm ieee</code>	Configures the switch to generate system logging messages for the alarms.
Step 6	<code>interface interface-id</code>	(Optional) Specifies an interface to configure, and enter interface configuration mode.
Step 7	<code>ethernet cfm mep domain domain-name mpid identifier vlan vlan-id</code>	<p>Configures maintenance end points for the domain, and enter ethernet cfm interface mep mode.</p> <ul style="list-style-type: none"> • domain domain-name—Specify the name of the created domain. • mpid identifier—Enter a maintenance end point identifier. The identifier must be unique for each VLAN (service instance). The range is 1 to 8191. • vlan vlan-id—Enter the service provider VLAN ID or IDs as a VLAN-ID (1 to 4094), a range of VLAN-IDs separated by a hyphen, or a series of VLAN IDs separated by comma.
Step 8	<code>alarm notification {all error-xcon mac-remote-error-xcon none remote-error-xcon xcon}</code>	<p>(Optional) Enables Ethernet CFM fault alarm notification for the specified defects on the interface.</p> <p>Note The Ethernet CFM interface MEP alarm configuration takes precedence over the global configuration.</p>
Step 9	<code>alarm {delay value reset value}</code>	<p>(Optional) Sets an alarm delay period or a reset period.</p> <p>Note The Ethernet CFM interface MEP alarm configuration takes precedence over the global configuration.</p>

	Command	Purpose
Step 10	end	Returns to privileged EXEC mode.
Step 11	show running-config	Verifies your entries.
Step 12	copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Use the **no** form of each command to remove a configuration or to return to the default settings.

The following example illustrates how to configure Ethernet CFM fault alarms:

```
Switch(config)# ethernet cfm alarm notification remote-error-xcon
Switch(config)# ethernet cfm logging alarm ieee
Switch(config)# interface gigabitethernet1/2
Switch(config-if)# ethernet cfm mep domain abc mpid 222 vlan 5
Switch(config-if-ecfm-mep)# alarm notification mac-remote-error-xcon
Switch(config-if)# end
```

Configuring IP SLAs CFM Operation

You can manually configure an IP SLA's Ethernet ping or jitter echo operation, or you can configure IP SLAs Ethernet operation with endpoint discovery. You can also configure multiple operation scheduling. For accurate one-way delay statistics, the clocks on the endpoint switches must be synchronized. You can configure the endpoint switches with Network Time Protocol (NTP) so that the switches are synchronized to the same clock source.

For detailed information about configuring IP SLAs Ethernet operations, see the *Cisco IOS IP SLAs for Metro-Ethernet* feature module at this URL:

http://www.cisco.com/en/US/docs/ios-xml/ios/ipsla/configuration/15-1s/Configuring_Cisco_IOS_IP_SLAs_for_Metro-Ethernet.html

and

http://www.cisco.com/en/US/docs/ios-xml/ios/ipsla/configuration/xr-2/Configuring_IP_SLAs_for_Metro-Ethernet.html

For detailed information about IP SLAs operations, see the *Cisco IOS IP SLAs Configuration Guide, Release 12.4T* at this URL:

http://www.cisco.com/en/US/products/ps6441/products_installation_and_configuration_guides_list.html

For detailed information about IP SLAs commands, see the command reference at this URL:

http://www.cisco.com/en/US/docs/ios/ipsla/command/reference/sla_book.html

This section includes these procedures:

- [Manually Configuring an IP SLAs CFM Probe or Jitter Operation, page 77-19](#)
- [Configuring an IP SLAs Operation with Endpoint Discovery, page 77-21](#)

Manually Configuring an IP SLAs CFM Probe or Jitter Operation

To manually configure an IP SLAs Ethernet echo (ping) or jitter operation, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# ip sla operation-number	Creates an IP SLAs operation, and enters IP SLAs configuration mode.
Step 3	Switch(config-ip-sla)# ethernet echo mpid identifier domain domain-name vlan vlan-id or ethernet jitter mpid identifier domain domain-name vlan vlan-id [interval interpacket-interval] [num-frames number-of frames transmitted]	Configures the IP SLAs operation as an echo (ping) or jitter operation, and enter IP SLAs Ethernet echo configuration mode. <ul style="list-style-type: none"> Enter echo for a ping operation or jitter for a jitter operation. For mpid identifier, enter a maintenance endpoint identifier. The identifier must be unique for each VLAN (service instance). The range is 1 to 8191. For domain domain-name, enter the CFM domain name. For vlan vlan-id, the VLAN range is from 1 to 4095. (Optional—for jitter only) Enter the interval between sending of jitter packets. (Optional—for jitter only) Enter the num-frames and the number of frames to be sent.
Step 4	Switch(config-ip-sla-ethernet-monitor)# cos cos-value	(Optional) Sets a class of service value for the operation. Before configuring the cos parameter on the switch, you must globally enable QoS by entering the mls qos global configuration command.
Step 5	Switch(config-ip-sla-ethernet-monitor)# frequency seconds	(Optional) Sets the rate at which the IP SLAs operation repeats. The range is from 1 to 604800 seconds; the default is 60 seconds.
Step 6	Switch(config-ip-sla-ethernet-monitor)# history history-parameter	(Optional) Specifies parameters for gathering statistical history information for the IP SLAs operation.
Step 7	Switch(config-ip-sla-ethernet-monitor)# owner owner-id	(Optional) Configures the SNMP owner of the IP SLAs operation.
Step 8	Switch(config-ip-sla-ethernet-monitor)# request-data-size bytes	(Optional) Specifies the protocol data size for an IP SLAs request packet. The range is from 0 to the maximum size allowed by the protocol being used; the default is 66 bytes.
Step 9	Switch(config-ip-sla-ethernet-monitor)# tag text	(Optional) Creates user-specified identifier for an IP SLAs operation.
Step 10	Switch(config-ip-sla-ethernet-monitor)# threshold milliseconds	(Optional) Specifies the upper threshold value in milliseconds (ms) for calculating network monitoring statistics. The range is 0 to 2147483647; the default is 5000.

	Command	Purpose
Step 11	Switch(config-ip-sla-ethernet-monitor)# timeout <i>milliseconds</i>	(Optional) Specifies the amount of time in ms that the IP SLAs operation waits for a response from its request packet. The range is 0 to 604800000; the default value is 5000.
Step 12	Switch(config-ip-sla-ethernet-monitor)# exit	Returns to global configuration mode.
Step 13	Switch(config)# ip sla schedule <i>operation-number</i> [ageout <i>seconds</i>] [life { forever <i>seconds</i> }] [recurring] [start-time { <i>hh:mm</i> { <i>:ss</i> } [<i>month day</i> <i>day month</i>] pending now after <i>hh:mm:ss</i> }]	Schedules the time parameters for the IP SLAs operation. <ul style="list-style-type: none"> <i>operation-number</i>—Enter the IP SLAs operation number. (Optional) ageout <i>seconds</i>—Enter the number of seconds to keep the operation in memory when it is not actively collecting information. The range is 0 to 2073600 seconds. The default is 0 seconds. (Optional) life—Set the operation to run indefinitely (forever) or for a specific number of <i>seconds</i>. The range is from 0 to 2147483647. The default is 3600 seconds (1 hour) (Optional) recurring—Set the probe to be automatically scheduled every day. (Optional) start-time—Enter the time for the operation to begin collecting information: <ul style="list-style-type: none"> To start at a specific time, enter the hour, minute, second (in 24-hour notation), and day of the month. Enter pending to select no information collection until a start time is selected. Enter now to start the operation immediately. Enter after <i>hh:mm:ss</i> to show that the operation should start after the entered time has elapsed.
Step 14	Switch(config)# end	Returns to privileged EXEC mode.
Step 15	Switch# show ip sla configuration [<i>operation-number</i>]	Shows the configured IP SLAs operation.
Step 16	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To remove an IP SLAs operation, enter the **no ip sla operation-number** global configuration command.

The following example shows how to configure an IP SLA CFM Probe or Jitter Operation:

```
Switch(config)# ip sla 1
Switch(config-ip-sla)# ethernet echo mpid 23 domain abc vlan 5
Switch(config-ip-sla-ethernet-echo)# exit
Switch(config)# ip sla schedule 1 start-time now

Switch# show ip sla configuration 1
IP SLAs, Infrastructure Engine-II.

Entry number: 1
Owner:
Tag:
Type of operation to perform: 802.1ag Echo
```

```

Target domain: abc
Target MPID: 23
Target VLAN ID: 5
Request size (Padding portion): 0
Operation timeout (milliseconds): 5000
Class Of Service parameters: 0
Schedule:
  Operation frequency (seconds): 60
  Next Scheduled Start Time: Start Time already passed
  Group Scheduled : FALSE
  Randomly Scheduled : FALSE
  Life (seconds): 3600
  Entry Ageout (seconds): never
  Recurring (Starting Everyday): FALSE
  Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 5000
Distribution Statistics:
  Number of statistic hours kept: 2
  Number of statistic distribution buckets kept: 1
  Statistic distribution interval (milliseconds): 20
Enhanced History:
History Statistics:
  Number of history Lives kept: 0
  Number of history Buckets kept: 15
  History Filter Type: None

```

Switch#

Configuring an IP SLAs Operation with Endpoint Discovery

To use IP SLAs to automatically discover the CFM endpoints for a domain and VLAN ID, perform this task. You can configure ping or jitter operations to the discovered endpoints.

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# ip sla ethernet-monitor <i>operation-number</i>	Begins configuration of an IP SLAs automatic Ethernet operation, and enter IP SLAs Ethernet monitor configuration mode.

	Command	Purpose
Step 3	<pre>Switch(config-ip-sla-ethernet-monitor)# type echo domain <i>domain-name</i> vlan <i>vlan-id</i> [exclude-mpids <i>mp-ids</i>] or type jitter domain <i>domain-name</i> vlan <i>vlan-id</i> [exclude-mpids <i>mp-ids</i>] [interval <i>interpacket-interval</i>] [num-frames <i>number-of</i> <i>frames transmitted</i>]</pre>	<p>Configures the automatic Ethernet operation to create echo (ping) or jitter operation and enters IP SLAs Ethernet echo configuration mode.</p> <ul style="list-style-type: none"> Enter type echo for a ping operation or type jitter for a jitter operation. For mpid identifier, enter a maintenance endpoint identifier. The range is 1 to 8191. For domain domain-name, enter the CFM domain name. For vlan vlan-id, the VLAN range is from 1 to 4095. (Optional) Enter exclude-mpids mp-ids to exclude the specified maintenance endpoint identifiers. (Optional—for jitter only) Enter the interval between sending of jitter packets. (Optional—for jitter only) Enter the num-frames and the number of frames to be sent.
Step 4	<pre>Switch(config-ip-sla-ethernet-echo)# cos <i>cos-value</i></pre>	(Optional) Sets a class of service value for the operation.
Step 5	<pre>Switch(config-ip-sla-ethernet-echo)# owner <i>owner-id</i></pre>	(Optional) Configures the SNMP owner of the IP SLAs operation.
Step 6	<pre>Switch(config-ip-sla-ethernet-echo)# request-data-size <i>bytes</i></pre>	(Optional) Specifies the protocol data size for an IP SLAs request packet. The range is from 0 to the maximum size allowed by the protocol being used; the default is 66 bytes.
Step 7	<pre>Switch(config-ip-sla-ethernet-echo)# tag <i>text</i></pre>	(Optional) Creates a user-specified identifier for an IP SLAs operation.
Step 8	<pre>Switch(config-ip-sla-ethernet-echo)# threshold <i>milliseconds</i></pre>	(Optional) Specifies the upper threshold value in milliseconds for calculating network monitoring statistics. The range is 0 to 2147483647; the default is 5000.
Step 9	<pre>Switch(config-ip-sla-ethernet-echo)# timeout <i>milliseconds</i></pre>	(Optional) Specifies the amount of time in milliseconds that the IP SLAs operation waits for a response from its request packet. The range is 0 to 604800000; the default value is 5000.
Step 10	<pre>Switch(config-ip-sla-ethernet-echo)# exit</pre>	Returns to global configuration mode.

	Command	Purpose
Step 11	Switch(config)# ip sla schedule <i>operation-number</i> [ageout <i>seconds</i>] [life { forever <i>seconds</i> }] [recurring] [start-time { <i>hh:mm</i> { <i>:ss</i> } [<i>month</i> <i>day</i> <i>day</i> <i>month</i>] pending now after <i>hh:mm:ss</i> }]	Schedules the time parameters for the IP SLAs operation. <ul style="list-style-type: none"> • <i>operation-number</i>—Enter the IP SLAs operation number. • (Optional) ageout <i>seconds</i>—Enter the number of seconds to keep the operation in memory when it is not actively collecting information. The range is 0 to 2073600 seconds. The default is 0 seconds. • (Optional) life—Set the operation to run indefinitely (forever) or for a specific number of <i>seconds</i>. The range is from 0 to 2147483647. The default is 3600 seconds (1 hour) • (Optional) recurring—Set the probe to be automatically scheduled every day. • (Optional) start-time—Enter the time for the operation to begin collecting information: <ul style="list-style-type: none"> – To start at a specific time, enter the hour, minute, second (in 24-hour notation), and day of the month. – Enter pending to select no information collection until a start time is selected. – Enter now to start the operation immediately. – Enter after <i>hh:mm:ss</i> to show that the operation should start after the entered time has elapsed.
Step 12	Switch(config)# end	Returns to privileged EXEC mode.
Step 13	Switch# show ip sla ethernet-monitor configuration [<i>operation-number</i>]	Shows the configured IP SLAs Auto Ethernet Monitor operation.
Step 14	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To remove an IP SLAs operation, enter the **no ip sla operation-number** global configuration command.

The following example shows how to configure an IP SLAs Operation with Endpoint Discovery:

```
Switch(config)# ip sla ethernet-monitor 10
Switch(config-ip-sla-ethernet-monitor)# type echo domain abc vlan 34
Switch(config-ip-sla-ethernet-params)# exit
Switch(config)# ip sla ethernet-monitor schedule 10 schedule-period 60 start-time now
Switch(config)# exit
```

```
Switch# show ip sla ethernet-monitor configuration 10
Entry Number : 10
Modification time : *10:12:01.725 UTC Mon Nov 29 2010
Operation Type : echo
Domain Name : abc
VLAN ID : 5
Excluded MPIDs :
Owner :
Tag :
Timeout(ms) : 5000
Threshold(ms) : 5000
Frequency(sec) : 60
```

```

Operations List      : Empty
Schedule Period(sec): 60
Request size        : 0
CoS                 : 0
Start Time          : Start Time already passed
SNMP RowStatus      : Active

```

Switch#

Configuring CFM on C-VLAN (Inner VLAN)

IEEE 802.1ag CFM brings in a support that allows customers to provision maintenance intermediate points (MIPs) and Up maintenance endpoints (MEPs) on the C-VLAN (inner VLAN) component of QinQ ports to provide visibility on the C-VLAN. C-VLANs are now supported on 802.1q tunnel ports. This allows monitoring or troubleshooting when QinQ is enabled on the provider edge (PE) device.

For more information about this feature and the supported commands, see:

http://www.cisco.com/en/US/docs/ios/cether/configuration/guide/ce_cfm-ieee_cvlan.html

The switch supports 802.1q-tunnel-port mode.

To configure Ethernet CFM CVLAN Up MEPs, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# ethernet cfm domain <i>domain-name level level-id</i>	Define a CFM domain, set the domain level, and enter ethernet-cfm configuration mode for the domain. The maintenance level number range is 0 to 7.
Step 3	Switch(config-ecfm)# service { <i>ma-name</i> <i>ma-number</i> <i>vpn-id</i> } vlan <i>svlan-id</i> inner-vlan <i>cvlan-id</i>	Define a customer service maintenance association name or number or VPN ID to be associated with the domain, define a CVLAN service, and enter ethernet-cfm-service configuration mode. <ul style="list-style-type: none"> • <i>ma-name</i>—a string of no more than 100 characters that identifies the MAID. • <i>ma-number</i>—a value from 0 to 65535. • <i>vpn-id</i> <i>vpn</i>—enter a VPN ID as the ma-name. • vlan <i>svlan-id</i>—VLAN range is from 1 to 4094. This identifies the outer VLAN (service provider vlan id) that CFM frames go out with. • inner-vlan <i>cvlan-id</i>—VLAN range is from 1 to 4094. This identifies the inner VLAN (customer VLAN) that is monitored through CFM.
Step 4	Switch(config-ecfm-arv)# continuity-check	Enables sending and receiving of continuity check messages.

	Command	Purpose
Step 5	Switch(config-ecfm-arv)# continuity-check interval <i>value</i>	(Optional) Set the interval at which continuity check messages are sent. The available values are 1 second, 10 seconds, 1 minute and 10 minutes. The default is 10 seconds. Note Because faster CCM rates are more CPU-intensive, we do not recommend configuring a large number of MEPs running at 1 s intervals.
Step 6	Switch(config-ecfm-arv)# continuity-check loss-threshold <i>threshold-value</i>	(Optional) Set the number of continuity check messages to be missed before declaring that an MEP is down. The range is 2 to 255; the default is 3.
Step 7	Switch(config-ecfm-arv)# exit	Returns to Return to ethernet-cfm configuration mode.
Step 8	Switch(config-ecfm)# exit	Returns to global configuration mode.
Step 9	Switch(config)# interface <i>interface-id</i>	Identify the CVLAN MEP interface and enter interface configuration mode.
Step 10	Switch(config-if)# ethernet cfm mep domain <i>domain-name</i> mpid identifier service { <i>ma-name</i> <i>ma-number</i> <i>vpn-id</i> }]	Configure the interface as a CVLAN Up MEP for the domain. <ul style="list-style-type: none"> • domain <i>domain-name</i>—Specify the name of the created domain. • mpid identifier—Enter a maintenance end point identifier. The identifier must be unique for each VLAN (service instance). The range is 1 to 8191. • service {<i>ma-name</i> <i>ma-number</i> <i>vpn-id</i>}—Use the same service identifier that was used for configuring CVLAN Service above in Step3.
Step 11	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 12	Switch# show ethernet cfm maintenance-points local	Verify the configuration.
Step 13	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Use the **no** form of each command to remove a configuration or to return to the default settings.

This is a sample configuration for a CVLAN Up MEP:

```
Switch(config)# ethernet cfm domain abc level 3
Switch(config-ecfm)# service CVLANMEP vlan 10 inner-vlan 20
Switch(config-ecfm-srv)# continuity-check
Switch(config-ecfm-srv)# exit
Switch(config-ecfm)# exit
Switch(config)# interface gigabitethernet1/1
Switch(config-if)# ethernet cfm mep domain abc mpid 1020 service CVLANMEP
Switch(config-if)# end
```

Similarly, a manual configuration of MIP for CVLAN is configured using the **ethernet cfm mip level level-id vlan svlan-id inner-vlan cvlan-id** command.

Feature Support and Behavior

CFM S-VLAN component support:

- Up MEPs at any level (0 to 7).

Up MEPs use the port access VLAN ID (the outer tag or S-VLAN).

CFM frames sent and received by Up MEPs have a single VLAN tag, and the VLAN identifier is the port access VLAN ID (S-VLAN). Because the 802.1q tunnel interface marks the endpoint of the S-VLAN, the associated S-VLAN component should mark the endpoint of the CFM domain running over the S-VLAN space.

CFM C-VLAN component support:

- Up MEP functions at any level (0 to 7).

Up MEPs use two tags: an outer tag with a VLAN ID that is the port access VLAN (S-VLAN) and an inner tag with a selected C-VLAN that is allowed through the 802.1q tunnel port. CFM frames sent and received by these Up MEPs are always double-tagged.

- MIP functions at any level (0 to 7).

MIPs process CFM frames that are single-tagged when coming from the wire-side and double-tagged when coming from the relay-function side.

- Transparent point functions.

Supported maintenance points on 802.1q tunnels:

- Up MEP on the C-VLAN component for selective or all-to-one bundling
- Up MEP on the S-VLAN
- Port MEP
- MIP support on C-VLAN component for selective or all-to-one bundling



Note

The switch supports only manual configuration of MIPs. It does not support MIP autocreation on C-VLANs.

Platform Restrictions and Limitations

- Maximum supported MEPs per switch at each continuity check message (CCM) interval:
 - 1600 MEP local and 1600 MEP remote (on C-VLAN and S-VLAN) with 10-second intervals
 - 250 MEP local and 250 MEP remote (on C-VLAN and S-VLAN) with 1-second intervals
- Maximum supported MIPs at each CCM interval:
 - 300 MIPs at 10 seconds
 - 125 MIPs at 1 second
- There could be issues detecting cross-connect errors on the Catalyst 4500 series switch.
- These features are not supported:
 - CFM C-component on the native VLAN
 - Down MEP on S or C-VLAN (provider network port)
 - MIP on S-VLAN (provider network port)
 - CFM C-VLAN alarm indication signal (AIS)

- 802.3ah interworking with CFM C-VLAN
- CFM C-VLAN IP SLAs
- CFM C-VLAN MIP autocreation
- CFM C-VLAN with One-to-One VLAN mapping on Trunk ports.

Understanding CFM ITU-T Y.1731 Fault Management

The ITU-T Y.1731 feature provides new CFM functionality for fault and performance management for service providers in large network. The switch supports Ethernet Alarm Indication Signal (ETH-AIS), Ethernet Remote Defect Indication (ETH-RDI), Ethernet Locked Signal (ETH-LCK), and Ethernet Multicast Loopback Message (MCAST-LBM) functionality for fault detection, verification, and isolation.

- [Y.1731 Terminology, page 77-27](#)
- [Alarm Indication Signals, page 77-28](#)
- [Ethernet Remote Defect Indication, page 77-28](#)
- [Multicast Ethernet Loopback, page 77-29](#)

Y.1731 Terminology

- Server MEP—the combination of the server layer termination function and server or Ethernet adaptation layer termination function or server or Ethernet adaptation function, where the server layer termination function is expected to run OAM mechanisms specific to the server layer. The supported mechanisms are link up, link down, and 802.3ah.
- Server layer—a virtual MEP layer capable of detecting fault conditions.
- Defect conditions:
 - Loss of continuity (LOC): the MEP stopped receiving CCM frames from a peer MEP
 - Mismatch: the MEP received a CCM frame with a correct maintenance level (matching the MEP level) but an incorrect maintenance ID.
 - Unexpected MEP: the MEP received a CCM frame with the correct maintenance level (matching the MEP's level) and correct maintenance ID, but an unexpected MEP ID.
 - Unexpected maintenance level: the MEP received a CCM frame with an incorrect maintenance level.
 - Unexpected period: the MEP received a CCM frame with a correct maintenance level, a correct maintenance ID, a correct MEP ID, but a different transmission period field.
- Signal fail—the MEP declares a signal fail condition when it detects a defect condition.
- Alarm Indication Signal (AIS) condition—the MEP received an AIS frame.
- Remote Defect Indication (RDI) condition—The MEP received a CCM frame with the RDI field set.

Alarm Indication Signals

The Ethernet Alarm Signal function (ETH-AIS) is used to suppress alarms after defects are detected at the *server* (sub) layer, which is a virtual MEP layer capable of detecting fault conditions. A fault condition could be a signal fail condition, an AIS condition, or a LCK condition.

**Note**

Although the configuration is allowed, you should not configure AIS in networks running STP. An STP configuration might cause AIS interruption or redirection.

When a MEP or a service MEP (SMEP) detects a connectivity fault at a specific maintenance association level, it multicasts AIS frames in the direction away from the detected failure at the client maintenance association level. The frequency of AIS frame transmission is based on the AIS transmission period. The first AIS frame is always sent immediately following the detection of the defect condition. We recommend a transition period of 1 second in a network of only a few VLANs to ensure that the first AIS frame is sent immediately following error detection. We recommend a 60-second interval in a network of multiple (up to 4094) VLANs to prevent stressing the network with 1-second transmissions.

A MEP that receives a frame with ETH-AIS information cannot determine the specific server with the defect condition or the set of peer MEPs for which it should suppress alarms. Therefore, it suppresses alarms for all peer MEPs, whether or not they are connected.

When a MEP receives an AIS frame, it examines it to be sure that the Maintenance Entity Group (MEG) level matches its own MEG and then detects the AIS default condition. (A MEG is Y.1731 terminology for maintenance association in 802.1ag.) After this detection, if no AIS frames are received for an interval of 3.5 times the AIS transmission period, the MEP clears the AIS defect condition. For example, if the AIS timer is set for 60 seconds, the AIS timer period expires after 3.5 times 60, or 210 seconds.

The AIS condition is terminated when a valid CCM is received with all error conditions cleared or when the AIS period timer expires (the default time is 60 seconds).

Ethernet Remote Defect Indication

When Ethernet OAM continuity check (ETH-CC) transmission is enabled, the Ethernet Remote Defect Indication (ETH-RDI) function uses a bit in the CFM CC message to communicate defect conditions to the MEP peers. For ETH-RDI functionality, you must configure the MEP MEG level, the ETH-CC transmission period, and the ETH-CC frame priority. ETH-RDI does not require any MIP configuration.

When a MEP receives frames with ETH-RDI information, it determines that its peer MEP has encountered a defect condition and sets the RDI files in the CCM frames for the duration of the defect condition. When the defect condition clears, the MEP clears the RDI field.

When a MEP receives a CCM frame, it examines it to ensure that its MEG level is the same and if the RDI field is set, it detects an RDI condition. For point-to-point Ethernet connections, a MEP can clear the RDI condition when it receives the first frame from its peer MEP with the RDI field cleared. However, for multipoint Ethernet connectivity, the MEP cannot determine the associated subset of peer MEPs with which the sending MEP has seen the defect condition. It can clear the RDI condition after it receives CCM frames with the RDI field cleared from its entire list of peer MEPs.

Multicast Ethernet Loopback

The multicast Ethernet loopback (ETH-LB) function verifies bidirectional connectivity of a MEP with its peer MEPs and is an on-demand OAM function. When the feature is invoked on a MEP by entering the **ping** privileged EXEC command, the MEP sends a multicast frame with ETH-LB request information to peer MEPs in the same MEG. The MEP expects to receive a unicast frame with ETH-LB reply information from its peer MEPs within a specified time period. A MEP receiving a multicast frame with ETH-LB request information validates the frame and transmits a frame with reply information.

To configure multicast ETH-LB, you configure the MEG level of the MEP and the priority of the multicast frames with ETH-LB requests. Multicast frames with ETH-LB request information are always marked as drop ineligible. No MIP configuration is required.

The MEP sends multicast LB message frames on an on-demand basis. After sending a multicast LBM frame, the MEP expects to receive LB reply frames within 5 seconds.

When a MEP receives a valid LBM frame, it generates an LB reply frame and sends it to the requested MEP after a random delay in the range of 0 to 1 second. The validity of the frame is determined on its having the correct MEG level.

When a MEP sends a multicast LBM frame and receives an LB reply frame within 5 seconds, the LB reply frame is valid.

Configuring Y.1731 Fault Management

To configure Y.1731 fault management, you must enable CFM and configure MIPs on the participating interfaces. AIS messages are generated only on interfaces with a configured MIP.

- [Default Y.1731 Configuration, page 77-29](#)
- [Configuring ETH-AIS, page 77-29](#)
- [Using Multicast Ethernet Loopback, page 77-31](#)

Default Y.1731 Configuration

ETH-AIS and ETH-LCK are enabled by default when CFM is enabled.

When you configure ETH-AIS or ETH-LCK, you must configure CFM before ETH-AIS or ETH-LCK is operational.

ETH-RDI is set automatically when continuity check messages are enabled.

Configuring ETH-AIS

Beginning in privileged EXEC mode, follow these steps to configure Ethernet AIS on a switch:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>ethernet cfm ais link-status global</code>	Configures AIS-specific SMEP commands by entering config-ais-link-cfm mode.

	Command	Purpose
Step 3	level <i>level-id</i> or disable	Configures the maintenance level for sending AIS frames transmitted by the SMEP. The range is 0 to 7. or Disables generation of ETH-AIS frames.
Step 4	period <i>value</i>	Configures the SMEP AIS transmission period interval. Allowable values are 1 second or 60 seconds.
Step 5	exit	Returns to global configuration mode.
Step 6	ethernet cfm domain <i>domain-name</i> level <i>level-id</i>	Defines a CFM domain, set the domain level, and enter ethernet-cfm configuration mode for the domain. The maintenance level number range is 0 to 7.
Step 7	service { <i>ma-name</i> / <i>ma-number</i> / <i>vpn-id</i> <i>vpn</i> } { vlan <i>vlan-id</i> [direction down] port }	Defines a customer service maintenance association (MA) name or number to be associated with the domain, or a VLAN ID or VPN-ID, and enter ethernet-cfm-service configuration mode. <ul style="list-style-type: none"> • <i>ma-name</i>—a string of no more than 100 characters that identifies the MAID. • <i>ma-number</i>—a value from 0 to 65535. • <i>vpn-id</i>—enter a VPN ID as the <i>ma-name</i>. • vlan <i>vlan-id</i>—VLAN range is from 1 to 4094. You cannot use the same VLAN ID for more than one domain at the same level. • (Optional) direction down—specify the service direction as down. • port—Configure port MEP, a down MEP that is untagged and not associated with a VLAN.
Step 8	ais level <i>level-id</i>	(Optional) Configures the maintenance level for sending AIS frames transmitted by the MEP. The range is 0 to 7.
Step 9	ais period <i>value</i>	(Optional) Configures the MEP AIS transmission period interval. Allowable values are 1 second or 60 seconds.
Step 10	ais expiry-threshold <i>value</i>	(Optional) Sets the expiring threshold for the MA as an integer. The range is 2 to 255. The default is 3.5.
Step 11	no ais suppress-alarms	(Optional) Overrides the suppression of redundant alarms when the MEP goes into an AIS defect condition after receiving an AIS message.
Step 12	exit	Returns to ethernet-cfm configuration mode.
Step 13	exit	Returns to global configuration mode.
Step 14	interface <i>interface-id</i>	Specifies an interface ID, and enter interface configuration mode.
Step 15	[no] ethernet cfm ais link-status	Enables or disable sending AIS frames from the SMEP on the interface.
Step 16	ethernet cfm ais link-status period <i>value</i>	Configures the ETH-AIS transmission period generated by the SMEP on the interface. Allowable values are 1 second or 60 seconds.

	Command	Purpose
Step 17	<code>ethernet cfm ais link-status level <i>level-id</i></code>	Configures the maintenance level for sending AIS frames transmitted by the SMEP on the interface. The range is 0 to 7.
Step 18	<code>end</code>	Returns to privileged EXEC mode.
Step 19	<code>show ethernet cfm smep [<i>interface interface-id</i>]</code>	Verifies the configuration.
Step 20	<code>show ethernet cfm error</code>	Displays received ETH-AIS frames and other errors.
Step 21	<code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

Use the **no** form of the commands to return to the default configuration or to remove a configuration. To disable the generation of ETH-AIS frames, enter the **disable** config-ais-link-cfm mode command.

This is an example of the output from the **show ethernet cfm smep** command when Ethernet AIS has been enabled:

```
Switch# show ethernet cfm smep
SMEP Settings:
-----
Interface: GigabitEthernet1/0/3
LCK-Status: Enabled
LCK Period: 60000 (ms)
Level to transmit LCK: Default
AIS-Status: Enabled
AIS Period: 60000 (ms)
Level to transmit AIS: Default
Defect Condition: AIS
```

Using Multicast Ethernet Loopback

You can use the **ping** privileged EXEC command to verify bidirectional connectivity of a MEP, as in this example:

```
Switch# ping ethernet multicast domain CD vlan 10
Type escape sequence to abort.
Sending 5 Ethernet CFM loopback messages to 0180.c200.0037, timeout is 5 seconds:
Reply to Multicast request via interface FastEthernet1/0/3, from 001a.a17e.f880, 8 ms
Total Loopback Responses received: 1
```

Managing and Displaying Ethernet CFM Information

You can use the privileged EXEC commands in these tables to clear Ethernet CFM information.

Table 77-1 Clearing CFM Information

Command	Purpose
<code>clear ethernet cfm ais domain <i>domain-name</i> mpid <i>id</i> {vlan <i>vlan-id</i> port}</code>	Clears MEPs with matching domain and VLAN ID out of AIS defect condition.
<code>clear ethernet cfm ais link-status interface <i>interface-id</i></code>	Clears a SMEP out of AIS defect condition.
<code>clear ethernet cfm error</code>	Clears all CFM error conditions, including AIS.

You can use the privileged EXEC commands in [Table 77-2](#) to display Ethernet CFM information.

Table 77-2 **Displaying CFM Information**

Command	Purpose
<code>show ethernet cfm domain [brief]</code>	Displays CFM domain information or brief domain information.
<code>show ethernet cfm errors [configuration domain-id]</code>	Displays CFM continuity check error conditions logged on a device since it was last reset or the log was last cleared. When CFM crosscheck is enabled, displays the results of the CFM crosscheck operation.
<code>show ethernet cfm maintenance-points local [detail domain interface level mep mip]</code>	Displays maintenance points configured on a device.
<code>show ethernet cfm maintenance-points remote [crosscheck detail domain static]</code>	Displays information about a remote maintenance point domains or levels or details in the CFM database.
<code>show ethernet cfm mpdb</code>	Displays information about entries in the MIP continuity-check database.
<code>show ethernet cfm smep [interface interface-id]</code>	Displays Ethernet CFM SMEP information.
<code>show ethernet cfm traceroute-cache</code>	Displays the contents of the traceroute cache.

This is an example of output from the `show ethernet cfm domain brief` command:

```
Switch# show ethernet cfm domain brief
Domain Name                               Index Level Services Archive(min)
level5                                   1      5      1      100
level3                                   2      3      1      100
test                                     3      3      3      100
name                                     4      3      1      100
test1                                    5      2      1      100
lck                                      6      1      1      100Total Services : 1
```

This is an example of output from the `show ethernet cfm errors` command:

```
Switch# show ethernet cfm errors
-----
MPID Domain Id                               Mac Address      Type  Id  Lvl
      MAName                               Reason              Age
-----
6307 level3                                0021.d7ee.fe80   Vlan  7   3
      vlan7                                Receive RDI       5s
```

This is an example of output from the `show ethernet cfm maintenance-points local detail` command:

```
Switch# show ethernet cfm maintenance-points local detail
Local MEPs:
-----
MPID: 7307
DomainName: level3
Level: 3
Direction: Up
Vlan: 7
Interface: Gi0/3
CC-Status: Enabled
CC Loss Threshold: 3
MAC: 0021.d7ef.0700
LCK-Status: Enabled
LCK Period: 60000(ms)
LCK Expiry Threshold: 3.5
```



```

Level to transmit LCK: Default
Defect Condition: No Defect
presentRDI: FALSE
AIS-Status: Enabled
AIS Period: 60000(ms)
AIS Expiry Threshold: 3.5
Level to transmit AIS: Default
Suppress Alarm configuration: Enabled
Suppressing Alarms: No

```

MIP Settings:

```
-----
```

Local MIPs:

```
* = MIP Manually Configured
```

```
-----
```

Level	Port	MacAddress	SrvcInst	Type	Id
*5	Gi0/3	0021.d7ef.0700	N/A	Vlan	2,7

```
-----
```

This is an example of output from the **show ethernet cfm traceroute** command:

```

Switch# show ethernet cfm traceroute
Current Cache-size: 0 Hops
Max Cache-size: 100 Hops
Hold-time: 100 Minutes

```

You can use the privileged EXEC commands in [Table 77-3](#) to display IP SLAs Ethernet CFM information.

Table 77-3 **Displaying IP SLAs CFM Information**

Command	Purpose
show ip sla configuration [<i>entry-number</i>]	Displays configuration values including all defaults for all IP SLAs operations or a specific operation.
show ip sla ethernet-monitor configuration [<i>entry-number</i>]	Displays the configuration of the IP SLAs automatic Ethernet operation.
show ip sla statistics [<i>entry-number</i> / aggregated / details]	Displays current or aggregated operational status and statistics.

About Ethernet OAM Protocol

The Ethernet OAM protocol for installing, monitoring, and troubleshooting Metro Ethernet networks and Ethernet WANs relies on an optional sublayer in the data link layer of the OSI model. Normal link operation does not require Ethernet OAM. You can implement Ethernet OAM on any full-duplex point-to-point or emulated point-to-point Ethernet link for a network or part of a network (specified interfaces).

OAM frames, called OAM protocol data units (OAM PDUs) use the slow protocol destination MAC address 0180.c200.0002. They are intercepted by the MAC sublayer and cannot propagate beyond a single hop within an Ethernet network. Ethernet OAM is a relatively slow protocol, with a maximum transmission rate of 10 frames per second, resulting in minor impact to normal operations. However, because the CPU must poll error counters frequently, when you enable link monitoring, the number of required CPU cycles is proportional to the number of interfaces that must be polled.

Ethernet OAM has two major components:

- The *OAM client* establishes and manages Ethernet OAM on a link and enables and configures the OAM sublayer. During the OAM discovery phase, the OAM client monitors OAM PDUs received from the remote peer and enables OAM functionality. After the discovery phase, it manages the rules of response to OAM PDUs and the OAM remote loopback mode.
- The *OAM sublayer* presents two standard IEEE 802.3 MAC service interfaces facing the superior and inferior MAC sublayers. It provides a dedicated interface for the OAM client to pass OAM control information and PDUs to and from the client. The sublayer includes these components:
 - The *control block* provides the interface between the OAM client and other OAM sublayer internal blocks.
 - The *multiplexer* manages frames from the MAC client, the control block, and the parser and passes OAM PDUs from the control block and loopback frames from the parser to the subordinate layer.
 - The *parser* classifies frames as OAM PDUs, MAC client frames, or loopback frames and sends them to the appropriate entity: OAM PDUs to the control block, MAC client frames to the superior sublayer, and loopback frames to the multiplexer.

OAM Features

These OAM features are defined by IEEE 802.3ah:

- *Discovery* identifies devices in the network and their OAM capabilities. It uses periodic OAM PDUs to advertise OAM mode, configuration, and capabilities; PDU configuration; and platform identity. An optional phase allows the local station to accept or reject the configuration of the peer OAM entity.
- *Link monitoring* detects and indicates link faults under a variety of conditions and uses the event notification OAM PDU to notify the remote OAM device when it detects problems on the link. Error events include when the number of symbol errors, the number of frame errors, the number of frame errors within a specified number of frames, or the number of error seconds within a specified period exceeding a configured threshold.
- *Remote failure indication* conveys a slowly deteriorating quality of an OAM entity to its peers by communicating these conditions: Link Fault means a loss of signal, Dying Gasp means an unrecoverable condition, and Critical Event means an unspecified vendor-specific critical event. The switch can receive and process but not generate Link Fault or Critical Event OAM PDUs. It can generate Dying Gasp OAM PDUs to show that Ethernet OAM is disabled, the interface is shut down, the interface enters the error-disabled state, or the switch is reloading. It can respond to, but not generate, Dying Gasp PDUs based on loss of power.
- *Remote loopback mode* ensures link quality with a remote peer during installation or troubleshooting. In this mode, when the switch receives a frame that is not an OAM PDU or a pause frame, it sends it back on the same port. The link appears to you to be functioning. You can use the returned loopback acknowledgement to test delay, jitter, and throughput.

OAM Messages

Ethernet OAM messages or PDUs are standard length, untagged Ethernet frames between 64 and 1518 bytes. They do not go beyond a single hop and have a maximum transmission rate of 10 OAM PDUs per second. Message types are information, event notification, loopback control, or vendor-specific OAM PDUs.

Enabling and Configuring Ethernet OAM

This section includes this information:

- [Ethernet OAM Default Configuration, page 77-35](#)
- [Ethernet OAM Configuration Guidelines, page 77-35](#)
- [Enabling Ethernet OAM on an Interface, page 77-36](#)
- [Enabling Ethernet OAM Remote Loopback, page 77-37](#)
- [Configuring Ethernet OAM Link Monitoring, page 77-38](#)
- [Configuring Ethernet OAM Remote Failure Indications, page 77-42](#)
- [Configuring Ethernet OAM Templates, page 77-45](#)

Ethernet OAM Default Configuration

The default configuration is as follows:

- Ethernet OAM is disabled on all interfaces.
- When Ethernet OAM is enabled on an interface, link monitoring is automatically turned on.
- Remote loopback is disabled.
- No Ethernet OAM templates are configured.

Ethernet OAM Configuration Guidelines

Follow these guidelines when configuring Ethernet OAM:

- The switch does not support monitoring of egress frames sent with cyclic redundancy code (CRC) errors. The **ethernet oam link-monitor transmit crc** interface-configuration or template-configuration commands are visible but are not supported on the switch. The commands are accepted but are not applied to an interface.
- For a remote failure indication, the switch does not generate Link Fault or Critical Event OAM PDUs. However, if these PDUs are received from a link partner, they are processed. The switch generates and receives Dying Gasp OAM PDUs when Ethernet OAM is disabled, the interface is shut down, the interface enters the error-disabled state, or the switch is reloading. It can respond to, but not generate, Dying Gasp PDUs based on loss of power.
- The switch does not support Ethernet OAM loopback on ports that belong to an EtherChannel, ISL trunk, and promiscuous trunk.

Enabling Ethernet OAM on an Interface

To enable Ethernet OAM on an interface, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Defines an interface to configure as an EOM interface, and enters interface configuration mode.
Step 3	Switch(config-if)# ethernet oam	Enables Ethernet OAM on the interface.
Step 4	Switch(config-if)# ethernet oam [max-rate <i>oampdus</i> min-rate <i>seconds</i> mode { active passive } timeout <i>seconds</i>]	<p>Configures these optional OAM parameters:</p> <ul style="list-style-type: none"> (Optional) Enter max-rate <i>oampdus</i> to configure the maximum number of OAM PDUs sent per second. The range is from 1 to 10. (Optional) Enter min-rate <i>seconds</i> to configure the minimum transmission rate in seconds when one OAM PDU is sent per second. The range is from 1 to 10. (Optional) Enter mode active to set OAM client mode to active. active is the default. (Optional) Enter mode passive to set OAM client mode to passive. <p>Note When Ethernet OAM mode is enabled on two interfaces passing traffic, at least one must be in the active mode.</p> <ul style="list-style-type: none"> (Optional) Enter timeout <i>seconds</i> to set a time for OAM client timeout. The range is from 2 to 30.
Step 5	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 6	Switch# show ethernet oam status [interface <i>interface-id</i>]	Verifies the configuration.
Step 7	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Enter the **no ethernet oam** interface configuration command to disable Ethernet OAM on the interface.

This example shows how to set basic OAM parameters on the switch:

```
Switch(config)# int gi1/3
Switch(config-if)# ethernet oam
Switch(config-if)# ethernet oam max-rate 9
Switch(config-if)# ethernet oam mode passive
Switch(config-if)# end
Switch# show ethernet oam status int gi1/2
GigabitEthernet1/2

General
-----
Admin state:          enabled
Mode:                 passive
PDU max rate:         9 packets per second
PDU min rate:         1 packet per 1 second
Link timeout:         5 seconds
High threshold action: no action
```

```

Link fault action:      no action
Dying gasp action:     no action
Critical event action:  no action

Link Monitoring
-----
Status: supported (on)

Symbol Period Error
Window:                100 x 1048576 symbols
Low threshold:         1 error symbol(s)
High threshold:        none

Frame Error
Window:                10 x 100 milliseconds
Low threshold:         1 error frame(s)
High threshold:        none

Frame Period Error
Window:                1000 x 10000 frames
Low threshold:         1 error frame(s)
High threshold:        none

Frame Seconds Error
Window:                100 x 100 milliseconds
Low threshold:         1 error second(s)
High threshold:        none

Receive-Frame CRC Error
Window:                10 x 100 milliseconds
Low threshold:         10 error frame(s)
High threshold:        none

Transmit-Frame CRC Error: Not Supported

```

Enabling Ethernet OAM Remote Loopback

You must enable Ethernet OAM remote loopback on an interface for the local OAM client to initiate OAM remote loopback operations. Changing this setting causes the local OAM client to exchange configuration information with its remote peer. Remote loopback is disabled by default.

Remote loopback has these limitations:

- Only data packets are looped back.
- You cannot configure Ethernet OAM remote loopback on ISL ports or ports that belong to an EtherChannel.
- Remote loopback can be supported on a max of 16 ports.

To enable Ethernet OAM remote loopback on an interface, follow these steps:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Defines an interface to configure as an EOM interface, and enters interface configuration mode.

	Command	Purpose
Step 3	Switch(config-if)# ethernet oam remote-loopback { supported timeout <i>seconds</i> }	Enables Ethernet remote loopback on the interface or set a loopback timeout period. <ul style="list-style-type: none"> Enter supported to enable remote loopback. Enter timeout <i>seconds</i> to set a remote loopback timeout period. The range is from 1 to 10 seconds.
Step 4	Switch(config-if)# ethernet oam remote-loopback { start stop } { interface <i>interface-id</i> }	Turns on or turn off Ethernet OAM remote loopback on an interface.
Step 5	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 6	Switch# show ethernet oam status [interface <i>interface-id</i>]	Verifies the configuration.
Step 7	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Use the **no ethernet oam remote-loopback** {**supported** | **timeout**} interface configuration command to disable remote loopback support or remove the timeout setting.

This example shows how to enable OAM Remote Loopback:

```
Switch(config)# int gi1/3
Switch(config-if)# ethernet oam
Switch(config-if)# ethernet oam remote-loopback supported
Switch(config-if)# end
Switch# show running int gi1/1
Building configuration...

Current configuration : 209 bytes
!
interface GigabitEthernet1/1
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,19
 switchport mode trunk
 ethernet oam remote-loopback supported
 ethernet oam
end

Switch# ethernet oam remote-loopback start int gi1/1
it is a intrusive loopback.
Therefore, while you test Ethernet OAM MAC connectivity,
you will be unable to pass traffic across that link.
Proceed with Remote Loopback? [confirm]

Switch# ethernet oam remote-loopback stop int gi1/1
Switch#
*Apr  9 12:52:39.793: %ETHERNET_OAM-6-LOOPBACK: Interface Gi1/1 has exited the master
loopback mode.
```

Configuring Ethernet OAM Link Monitoring

You can configure high and low thresholds for link-monitoring features. If no high threshold is configured, the default is **none**; no high threshold is set. If you do not set a low threshold, the default is a value lower than the high threshold.

Link event PDUs for rxcrc and trxcrc errors are not generated because they are nonstandard.

To configure Ethernet OAM link monitoring on an interface, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Defines an interface, and enters interface configuration mode.
Step 3	Switch(config-if)# ethernet oam link-monitor supported	Enables the interface to support link monitoring. it is the default. You need to enter this command only if it has been disabled by previously entering the no ethernet oam link-monitor supported command.
Step 4	Switch(config-if)# ethernet oam link-monitor symbol-period { threshold { high { <i>high symbols</i> none } low { <i>low-symbols</i> }} window <i>symbols</i> } Repeat this step to configure both high and low thresholds.	(Optional) Configures high and low thresholds for an error-symbol period that trigger an error-symbol period link event. <ul style="list-style-type: none"> Enter threshold high <i>high-symbols</i> to set a high threshold in number of symbols. The range is 1 to 65535. The default is none. Enter threshold high none to disable the high threshold if it was set. it is the default. Enter threshold low <i>low-symbols</i> to set a low threshold in number of symbols. The range is 0 to 65535. It must be lower than the high threshold. Enter window <i>symbols</i> to set the window size (in number of symbols) of the polling period. The range is 1 to 65535 symbols.
Step 5	Switch(config-if)# ethernet oam link-monitor frame { threshold { high { <i>high-frames</i> none } low { <i>low-frames</i> }} window <i>milliseconds</i> } Repeat this step to configure both high and low thresholds.	(Optional) Configures high and low thresholds for error frames that trigger an error-frame link event. <ul style="list-style-type: none"> Enter threshold high <i>high-frames</i> to set a high threshold in number of frames. The range is 1 to 65535. The default is none. Enter threshold high none to disable the high threshold if it was set. it is the default. Enter threshold low <i>low-frames</i> to set a low threshold in number of frames. The range is 0 to 65535. The default is 1. Enter window <i>milliseconds</i> to set the a window and period of time during which error frames are counted. The range is 10 to 600 and represents the number of milliseconds in multiples of 100. The default is 100.

Command	Purpose
<p>Step 6</p> <pre>Switch(config-if)# ethernet oam link-monitor frame-period {threshold {high {high-frames none} low {low-frames}}} window frames}</pre> <p>Repeat this step to configure both high and low thresholds.</p>	<p>(Optional) Configures high and low thresholds for the error-frame period that triggers an error-frame-period link event.</p> <ul style="list-style-type: none"> • Enter threshold high <i>high-frames</i> to set a high threshold in number of frames. The range is 1 to 65535. The default is none. • Enter threshold high none to disable the high threshold if it was set. it is the default. • Enter threshold low <i>low-frames</i> to set a low threshold in number of frames. The range is 0 to 65535. The default is 1. • Enter window <i>frames</i> to set the a polling window size in number of frames. The range is 1 to 65535; each value is a multiple of 10000 frames. The default is 1000.
<p>Step 7</p> <pre>Switch(config-if)# ethernet oam link-monitor frame-seconds {threshold {high {high-frames none} low {low-frames}}} window milliseconds}</pre> <p>Repeat this step to configure both high and low thresholds.</p>	<p>(Optional) Configures high and low thresholds for the frame-seconds error that triggers an error-frame-seconds link event.</p> <ul style="list-style-type: none"> • Enter threshold high <i>high-frames</i> to set a high error frame-seconds threshold in number of seconds. The range is 1 to 900. The default is none. • Enter threshold high none to disable the high threshold if it was set. it is the default. • Enter threshold low <i>low-frames</i> to set a low threshold in number of frames. The range is 1 to 900. The default is 1. • Enter window <i>frames</i> to set the a polling window size in number of milliseconds. The range is 100 to 9000; each value is a multiple of 100 milliseconds. The default is 1000.

	Command	Purpose
Step 8	<p>Switch(config-if)# ethernet oam link-monitor receive-crc {threshold {high {high-frames none} low {low-frames}} window milliseconds}</p> <p>Repeat this step to configure both high and low thresholds.</p>	<p>(Optional) Configures thresholds for monitoring ingress frames received with cyclic redundancy code (CRC) errors for a period of time.</p> <ul style="list-style-type: none"> Enter threshold high high-frames to set a high threshold for the number of frames received with CRC errors. The range is 1 to 65535 frames. Enter threshold high none to disable the high threshold. Enter threshold low low-frames to set a low threshold in number of frames. The range is 0 to 65535. The default is 1. Enter window milliseconds to set the a window and period of time during which frames with CRC errors are counted. The range is 10 to 1800 and represents the number of milliseconds in multiples of 100. The default is 100.
Step 9	Switch(config-if)# [no] ethernet link-monitor on	(Optional) Starts or stop (when the no keyword is entered) link-monitoring operations on the interface. Link monitoring operations start automatically when support is enabled.
Step 10	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 11	Switch# show ethernet oam status [interface interface-id]	Verifies the configuration.
Step 12	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

The **ethernet oam link-monitor transmit-crc {threshold {high {high-frames | none} | low {low-frames}} | window milliseconds}** command is visible on the switch and you can enter it, but it is not supported. Enter the **no** form of the command to disable the configuration. Use the **no** form of each command to disable the threshold setting.

Symbol error counters are supported on the following line cards and supervisor engine cards:

- Supervisor engine cards: WS-X4515, WS-X4516, WS-X4013+, WS-X4013+TS, WS-X4516-10GE, WS-X4013+10GE
- Line cards: WS-X4148-RJ, WS-X4124-RJ, WS-X4232, WS-X4232-RJ-XX, WS-X4148-RJ21, WS-X4504-FX-MT, WS-X4224-RJ21-XX, WS-X4124-FX-MT, WS-X4232-L3

The rest of the cards do not support symbol error counters.

This example shows how to configure Ethernet OAM link monitoring:

```
Switch(config)# interface gi1/1
Switch(config-if)# ethernet oam link-monitor receive-crc threshold high 1000
Switch(config-if)# ethernet oam link-monitor receive-crc threshold low 10
Switch(config-if)# ethernet oam link-monitor symbol-period threshold high 5000
Switch(config-if)# ethernet oam link-monitor symbol-period threshold low 5
Switch(config-if)# ethernet oam link-monitor frame threshold high 8000
Switch(config-if)# ethernet oam link-monitor frame threshold low 8
Switch(config-if)# ethernet oam link-monitor frame-period threshold high 9000
Switch(config-if)# ethernet oam link-monitor frame-period threshold low 9
```

```
Switch# show ethernet oam status int gi1/1
```

```

GigabitEthernet1/1
General
-----
Admin state:          enabled
Mode:                 active
PDU max rate:         10 packets per second
PDU min rate:         1 packet per 1 second
Link timeout:         5 seconds
High threshold action: error disable interface
Link fault action:    no action
Dying gasp action:    no action
Critical event action: no action

Link Monitoring
-----
Status: supported (on)

Symbol Period Error
Window:              100 x 1048576 symbols
Low threshold:       5 error symbol(s)
High threshold:      5000 error symbol(s)

Frame Error
Window:              10 x 100 milliseconds
Low threshold:       8 error frame(s)
High threshold:      8000 error frame(s)

Frame Period Error
Window:              1000 x 10000 frames
Low threshold:       9 error frame(s)
High threshold:      9000 error frame(s)

Frame Seconds Error
Window:              100 x 100 milliseconds
Low threshold:       1 error second(s)
High threshold:      none

Receive-Frame CRC Error
Window:              10 x 100 milliseconds
Low threshold:       10 error frame(s)
High threshold:      1000 error frame(s)

Transmit-Frame CRC Error: Not Supported

```

Configuring Ethernet OAM Remote Failure Indications

You can configure an error-disable action to occur on an interface when the following occur:

- Crossing the high thresholds configured on the interface for link monitoring
- On reception of Dying Gasp, executing **shut** on the interface
- On reception of Dying Gasp, **executing reload** command
- On reception of Dying Gasp, **executing no ethernet oam** command on the interface

To enable Ethernet OAM remote-failure indication actions on an interface, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Defines an interface and enters interface configuration mode.
Step 3	Switch(config-if)# ethernet oam remote-failure [dying-gasp] action error-disable-interface	Configures the Ethernet OAM remote-failure action on the interface. You can configure disabling the interface by selecting dying-gasp to shut down the interface when Ethernet OAM is disabled or the interface enters the error-disabled state.
Step 4	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 5	Switch# show ethernet oam status [interface <i>interface-id</i>]	Verifies the configuration.
Step 6	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

This example shows how to configure Ethernet OAM remote-failure action on the switch interface:

```
Switch# conf terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# int g1/1
Switch(config-if)# ethernet oam remote-failure dying-gasp action error
Switch(config-if)# ethernet oam link-monitor high-threshold action error
Switch(config-if)# end
Switch# show running-config int g1/1
Building configuration...

Current configuration : 353 bytes
!
interface GigabitEthernet1/1
 switchport trunk encapsulation dot1q
 switchport trunk allowed vlan 1,19
 switchport mode trunk
 ethernet oam remote-loopback supported
 ethernet oam link-monitor high-threshold action error-disable-interface
 ethernet oam remote-failure dying-gasp action error-disable-interface
 ethernet oam
end
Switch# show ethernet oam status int g1/1
GigabitEthernet1/1
General
-----
Admin state:          enabled
Mode:                 active
PDU max rate:         10 packets per second
PDU min rate:         1 packet per 1 second
Link timeout:         5 seconds
High threshold action: error disable interface
Link fault action:    no action
Dying gasp action:    error disable interface
Critical event action: no action
```

Link Monitoring

Status: supported (on)

Symbol Period Error

Window: 100 x 1048576 symbols
 Low threshold: 1 error symbol(s)
 High threshold: none

Frame Error

Window: 10 x 100 milliseconds
 Low threshold: 1 error frame(s)
 High threshold: none

Frame Period Error

Window: 1000 x 10000 frames
 Low threshold: 1 error frame(s)
 High threshold: none

Frame Seconds Error

Window: 100 x 100 milliseconds
 Low threshold: 1 error second(s)
 High threshold: none

Receive-Frame CRC Error

Window: 10 x 100 milliseconds
 Low threshold: 10 error frame(s)
 High threshold: none

Transmit-Frame CRC Error: Not Supported

To enable Ethernet OAM failover action on an EtherChannel interface, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface port-channel <i>interface-id</i>	Defines an interface and enters interface configuration mode.
Step 3	Switch(config-if)# switchport mode <i>mode</i>	Configures the mode of the EtherChannel interface.
Step 4	Switch(config-if)# ethernet oam link-monitor high-threshold action failover	Configures the Ethernet OAM remote-failure action on the port channel interface to failover. This action is configurable only for link monitoring RFI. If failover is configured on the EtherChannel interface, the interface is not error-disabled if it is the last member port of the EtherChannel.
Step 5	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 6	Switch# show ethernet oam status [<i>interface</i> <i>interface-id</i>]	Verifies the configuration.
Step 7	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

The switch does not generate Link Fault or Critical Event OAM PDUs. However, if these PDUs are received from a link partner, they are processed. The switch supports sending and receiving Dying Gasp OAM PDUs when Ethernet OAM is disabled, the interface is shut down, the interface enters the

error-disabled state, or the switch is reloading. It can respond to but not generate Dying Gasp PDUs based on loss of power. Enter the **no ethernet remote-failure {critical-event | dying-gasp | link-fault} action** command to disable the remote failure indication action.

Configuring Ethernet OAM Templates

You can create a template for configuring a common set of options on multiple Ethernet OAM interfaces. The template can be configured to monitor frame errors, frame-period errors, frame-second errors, received CRS errors, and symbol-period errors and thresholds. You can also set the template to put the interface in error-disabled state if any high thresholds are exceeded. These steps are optional and can be performed in any sequence or repeated to configure different options.

To configure an Ethernet OAM template and to associate it with an interface, follow these steps:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# template <i>template-name</i>	Creates a template and enters template configuration mode.
Step 3	Switch(config-template)# ethernet oam link-monitor receive-crc {threshold {high {<i>high-frames</i> none} low {<i>low-frames</i>}} window <i>milliseconds</i>}	<p>(Optional) Configures thresholds for monitoring ingress frames received with cyclic redundancy code (CRC) errors for a period of time.</p> <ul style="list-style-type: none"> • Enter threshold high <i>high-frames</i> to set a high threshold for the number of frames received with CRC errors. The range is 1 to 65535 frames. • Enter threshold high none to disable the high threshold. • Enter threshold low <i>low-frames</i> to set a low threshold in number of frames. The range is 0 to 65535. The default is 1. • Enter window <i>milliseconds</i> to set the a window and period of time during which frames with CRC errors are counted. The range is 10 to 1800 and represents the number of milliseconds in multiples of 100. The default is 100.

Command	Purpose
Step 4 Switch(config-template)# ethernet oam link-monitor symbol-period { threshold { high { <i>high symbols</i> / none } low { <i>low-symbols</i> }} window <i>symbols</i> }	(Optional) Configures high and low thresholds for an error-symbol period that triggers an error-symbol period link event. <ul style="list-style-type: none"> • Enter threshold high <i>high-symbols</i> to set a high threshold in number of symbols. The range is 1 to 65535. • Enter threshold high none to disable the high threshold. • Enter threshold low <i>low-symbols</i> to set a low threshold in number of symbols. The range is 0 to 65535. It must be lower than the high threshold. • Enter window <i>symbols</i> to set the window size (in number of symbols) of the polling period. The range is 1 to 65535 symbols.
Step 5 Switch(config-template)# ethernet oam link-monitor frame { threshold { high { <i>high-frames</i> none } low { <i>low-frames</i> }} window <i>milliseconds</i> }	(Optional) Configures high and low thresholds for error frames that trigger an error-frame link event. <ul style="list-style-type: none"> • Enter threshold high <i>high-frames</i> to set a high threshold in number of frames. The range is 1 to 65535. You must enter a high threshold. • Enter threshold high none to disable the high threshold. • Enter threshold low <i>low-frames</i> to set a low threshold in number of frames. The range is 0 to 65535. The default is 1. • Enter window <i>milliseconds</i> to set the a window and period of time during which error frames are counted. The range is 10 to 600 and represents the number of milliseconds in a multiple of 100. The default is 100.
Step 6 Switch(config-template)# ethernet oam link-monitor frame-period { threshold { high { <i>high-frames</i> none } low { <i>low-frames</i> }} window <i>frames</i> }	(Optional) Configures high and low thresholds for the error-frame period that triggers an error-frame-period link event. <ul style="list-style-type: none"> • Enter threshold high <i>high-frames</i> to set a high threshold in number of frames. The range is 1 to 65535. You must enter a high threshold. • Enter threshold high none to disable the high threshold. • Enter threshold low <i>low-frames</i> to set a low threshold in number of frames. The range is 0 to 65535. The default is 1. • Enter window <i>frames</i> to set the a polling window size in number of frames. The range is 1 to 65535; each value is a multiple of 10000 frames. The default is 1000.

	Command	Purpose
Step 7	Switch(config-template)# ethernet oam link-monitor frame-seconds {threshold {high {high-seconds none} low {low-seconds}} window milliseconds}	(Optional) Configures frame-seconds high and low thresholds for triggering an error-frame-seconds link event. <ul style="list-style-type: none"> Enter threshold high high-seconds to set a high threshold in number of seconds. The range is 1 to 900. You must enter a high threshold. Enter threshold high none to disable the high threshold. Enter threshold low low-frames to set a low threshold in number of frames. The range is 1 to 900. The default is 1. Enter window frames to set the a polling window size in number of frames. The range is 100 to 9000; each value is a multiple of 100 milliseconds. The default is 1000.
Step 8	Switch(config-template)# ethernet oam link-monitor high threshold action error-disable-interface	(Optional) Configures the switch to put an interface in an error disabled state when a high threshold for an error is exceeded.
Step 9	Switch(config-template)# exit	Returns to global configuration mode.
Step 10	Switch(config)# interface interface-id	Defines an Ethernet OAM interface and enters interface configuration mode.
Step 11	Switch(config-if# source-template template-name	Associates the template to apply the configured options to the interface.
Step 12	Switch(config-if# end	Returns to privileged EXEC mode.
Step 13	Switch# show ethernet oam status [interface interface-id]	Verifies the configuration.
Step 14	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

The switch does not support monitoring egress frames with CRC errors. The **ethernet oam link-monitor transmit-crc {threshold {high {high-frames | none} | low {low-frames}} | window milliseconds}** command is visible on the switch and you can enter it, but it is not supported. Use the **no** form of each command to remove the option from the template. Use the **no source-template template-name** command to remove the source template association.

The following example illustrates how to configure an Ethernet OAM template and to associate it with an interface:

```
Switch# conf t
Switch(config)# template oam
Switch(config-template)# ethernet oam link-monitor receive-crc threshold high 1000
Switch(config-template)# ethernet oam link-monitor receive-crc threshold low 10
Switch(config-template)# ethernet oam link-monitor symbol-period threshold high 5000
Switch(config-template)# ethernet oam link-monitor symbol-period threshold low 5
Switch(config-template)# ethernet oam link-monitor frame threshold high 8000
Switch(config-template)# ethernet oam link-monitor frame threshold low 8
Switch(config-template)# ethernet oam link-monitor frame-period threshold high 9000
Switch(config-template)# ethernet oam link-monitor frame-period threshold low 9
Switch(config-template)# ethernet oam link-monitor high action error-disable-interface
Switch(config-template)# exit
Switch(config)# int gi1/2
Switch(config-if)# source template oam
Switch(config-if)# end
```

```

Switch# show ethernet oam status int gi1/2
GigabitEthernet1/2
General
-----
Admin state:          enabled
Mode:                 active
PDU max rate:         10 packets per second
PDU min rate:         1 packet per 1 second
Link timeout:         5 seconds
High threshold action: error disable interface
Link fault action:    no action
Dying gasp action:    no action
Critical event action: no action

Link Monitoring
-----
Status: supported (on)

Symbol Period Error
Window:              100 x 1048576 symbols
Low threshold:       5 error symbol(s)
High threshold:      5000 error symbol(s)

Frame Error
Window:              10 x 100 milliseconds
Low threshold:       8 error frame(s)
High threshold:      8000 error frame(s)

Frame Period Error
Window:              1000 x 10000 frames
Low threshold:       9 error frame(s)
High threshold:      9000 error frame(s)

Frame Seconds Error
Window:              100 x 100 milliseconds
Low threshold:       1 error second(s)
High threshold:      none

Receive-Frame CRC Error
Window:              10 x 100 milliseconds
Low threshold:       10 error frame(s)
High threshold:      1000 error frame(s)

Transmit-Frame CRC Error: Not Supported

```


Displaying Ethernet OAM Protocol Information

To display Ethernet OAM protocol information, you can use the privileged EXEC commands in [Table 77-4](#).

Table 77-4 *Displaying Ethernet OAM Protocol Information*

Command	Purpose
show ethernet oam discovery [interface interface-id]	Displays discovery information for all Ethernet OAM interfaces or the specified interface.
show ethernet oam statistics [interface interface-id]	Displays detailed information about Ethernet OAM packets.
show ethernet oam status [interface interface-id]	Displays Ethernet OAM configuration for all interfaces or the specified interface.
show ethernet oam summary	Displays active Ethernet OAM sessions on the switch.

These examples show how to apply these commands:

```
Switch# show ethernet oam discovery
GigabitEthernet1/1
Local client
-----
Administrative configurations:
  Mode:                active
  Unidirection:        not supported
  Link monitor:         supported (on)
  Remote loopback:      supported
  MIB retrieval:        not supported
  Mtu size:             1500

Operational status:
  Port status:          operational
  Loopback status:      no loopback
  PDU revision:         10

Remote client
-----
MAC address: 000f.8f03.3591
Vendor(oui): 00000C(cisco)

Administrative configurations:
  PDU revision:         2
  Mode:                active
  Unidirection:        not supported
  Link monitor:         supported
  Remote loopback:      supported
  MIB retrieval:        not supported
  Mtu size:             1500

Switch# show ethernet oam statistics
GigabitEthernet1/1
Counters:
-----
Information OAMPDU Tx           : 101163
Information OAMPDU Rx           : 51296
Unique Event Notification OAMPDU Tx : 0
```

```

Unique Event Notification OAMPDU Rx      : 0
Duplicate Event Notification OAMPDU TX   : 0
Duplicate Event Notification OAMPDU RX   : 0
Loopback Control OAMPDU Tx               : 12
Loopback Control OAMPDU Rx               : 0
Variable Request OAMPDU Tx               : 0
Variable Request OAMPDU Rx               : 0
Variable Response OAMPDU Tx              : 0
Variable Response OAMPDU Rx              : 0
Cisco OAMPDU Tx                          : 7
Cisco OAMPDU Rx                          : 8
Unsupported OAMPDU Tx                    : 0
Unsupported OAMPDU Rx                    : 0
Frames Lost due to OAM                   : 0

Local Faults:
-----
 0 Link Fault records
 2 Dying Gasp records
   Total dying gasps      : 7
   Time stamp             : 1d01h

   Total dying gasps      : 6
   Time stamp             : 1d01h

 0 Critical Event records

Remote Faults:
-----
 0 Link Fault records
 2 Dying Gasp records
   Total dying gasps      : 8
   Time stamp             : 1d01h

   Total dying gasps      : 7
   Time stamp             : 1d01h

 0 Critical Event records

Local event logs:
-----
 0 Errored Symbol Period records
 0 Errored Frame records
 0 Errored Frame Period records
 0 Errored Frame Second records

Remote event logs:
-----
 0 Errored Symbol Period records
 0 Errored Frame records
 0 Errored Frame Period records
 0 Errored Frame Second records

Switch# show ethernet oam summary
Symbols:      * - Master Loopback State, # - Slave Loopback State
              & - Error Block State
Capability codes: L - Link Monitor, R - Remote Loopback
                  U - Unidirection, V - Variable Retrieval

   Local                               Remote
Interface      MAC Address    OUI      Mode      Capability

Gi1/1          000f.8f03.3591 00000C active    L R

```

Ethernet CFM and Ethernet OAM Interaction

You can also configure the OAM Manager infrastructure to interact between CFM and Ethernet OAM. When the Ethernet OAM protocol is running on an interface that has CFM MEPs configured, Ethernet OAM informs CFM of the state of the interface. Interaction is unidirectional from the Ethernet OAM to the CFM protocol, and the only information exchanged is your (user) network interface port status.

The Ethernet OAM protocol notifies CFM when these conditions occur:

- Error thresholds are crossed at the local interface.

CFM responds to the notification by sending a port status of *Local_Excessive_Errors* in the Port StatusType Length Value (TLV).

- Ethernet OAM receives an OAM PDU from the remote side showing that an error threshold is exceeded on the remote endpoint.

CFM responds to the notification by sending a port status of *Remote_Excessive_Errors* in the Port Status TLV.

- The local port is set into loopback mode.

CFM responds by sending a port status of Test in the Port Status TLV.

- The remote port is set into loopback mode.

CFM responds by sending a port status of Test in the Port Status TLV.

This section includes this information:

- [Configuring Ethernet OAM Interaction with CFM, page 77-51](#)
- [Example: Configuring Ethernet OAM and CFM, page 77-53](#)

For more information about CFM and interaction with Ethernet OAM, see the Ethernet Connectivity Fault Management feature module at this URL:

http://www.cisco.com/en/US/docs/ios/12_2sr/12_2sra/feature/guide/srethcfm.html

Configuring Ethernet OAM Interaction with CFM

For Ethernet OAM to function with CFM, you must configure an Ethernet Virtual Circuit (EVC) and the OAM manager, and associate the EVC with CFM. You must use an inward facing MEP for interaction with the OAM manager.



Note

If you configure, change, or remove a UNI service type, EVC, Ethernet service instance, or CE-VLAN configuration, all configurations are verified to ensure that the UNI service types match the EVC configuration and that Ethernet service instances are matched with the CE-VLAN configuration. Configurations are rejected if the pairs do not match.

Configuring the OAM Manager

To configure the OAM manager on a PE device, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Defines an interface to configure as an Ethernet OAM interface and enter interface configuration mode.
Step 3	Switch(config-if)# ethernet cfm domain <i>domain-name</i> level <i>level-id</i>	Defines a CFM domain, set the domain level, and enter ethernet-cfm configuration mode for the domain. The maintenance level number range is 0 to 7.
Step 4	Switch(config-if)# service <i>csi-id</i> vlan <i>vlan-id</i>	Defines a universally unique customer service instance (CSI) and VLAN ID within the maintenance domain. <ul style="list-style-type: none"> <i>csi-id</i>—String of no more than 100 characters that identifies the CSI. <i>vlan-id</i>—VLAN range is from 1 to 4095. You cannot use the same VLAN ID for more than one domain at the same level.
Step 5	Switch(config-if)# exit	Returns to global configuration mode.
Step 6	Switch(config)# ethernet evc <i>evc-id</i>	Defines an EVC, and enter EVC configuration mode
Step 7	Switch(config-evc)# oam protocol cfm svlan <i>vlan-id</i> domain <i>domain-name</i>	Configures the EVC OAM protocol as CFM, and identify the service provider VLAN-ID (S-VLAN-ID) for the CFM domain maintenance level as configured in Steps 2 and 3.
Step 8	Switch(config-evc)# exit	Returns to global configuration mode.
Step 9	Repeat Steps 2 through 7 to define other CFM domains that you want OAM manager to monitor.	
Step 10	Switch(config)# ethernet cfm enable	Globally enables CFM.
Step 11	Switch(config)# end	Returns to privileged EXEC mode.
Step 12	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Enabling Ethernet OAM

To enable Ethernet OAM on an interface, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Defines an interface to configure as an Ethernet OAM interface and enter interface configuration mode.

	Command	Purpose
Step 3	Switch(config-if)# ethernet oam [max-rate <i>oampdus</i> / min-rate <i>seconds</i> / mode { active / passive } timeout <i>seconds</i>]	Enables Ethernet OAM on the interface <ul style="list-style-type: none"> • (Optional) Enter max-rate <i>oampdus</i> to set the maximum rate (per second) to send OAM PDUs. The range is 1 to 10 PDUs per second; the default is 10. • (Optional) Enter min-rate <i>seconds</i> to set the minimum rate in seconds. The range is 1 to 10 seconds. • (Optional) Set the OAM client mode as active or passive. The default is active. • (Optional) Enter timeout <i>seconds</i> to set the time after which a device declares the OAM peer to be nonoperational and resets its state machine. The range is 2 to 30 seconds; the default is 5 seconds.
Step 4	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 5	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.
Step 6	Switch# show ethernet cfm maintenance points remote	(Optional) Displays the port states as reported by Ethernet OAM.

Example: Configuring Ethernet OAM and CFM

These are configuration examples of the interworking between Ethernet OAM and CFM in a sample service provider network. This example network would contain a provider-edge switch connected to a customer edge switch at each endpoint. You must configure CFM, E-LMI, and Ethernet OAM between the customer edge and the provider edge switch.

Customer-edge switch 1 (CE1) configuration:

```
Switch# config terminal
Switch(config)# interface GigabitEthernet1/1
Switch(config-if)# switchport trunk allowed vlan 10
Switch(config-if)# switchport mode trunk
Switch(config-if)# ethernet oam remote-loopback supported
Switch(config-if)# ethernet oam
Switch(config-if)# exit
```

Provider-edge switch 1 (PE1) configuration:

```
Switch# config terminal
Switch(config)# interface FastEthernet1/20
Switch(config-if)# switchport trunk encapsulation dot1q
Switch(config-if)# switchport mode trunk
Switch(config-if)# ethernet cfm mip level 7
Switch(config-if)# ethernet cfm mep level 4 mpid 100 vlan 100
Switch(config-if)# ethernet oam remote-loopback supported
Switch(config-if)# ethernet oamt
```

Provider-edge switch 2 (PE2) configuration:

```
Switch# config terminal
Switch(config)# interface GigabitEthernet1/20
Switch(config-if)# switchport mode trunk
Switch(config-if)# ethernet cfm mip level 7
Switch(config-if)# ethernet cfm mep level 4 mpid 101 vlan 10
Switch(config-if)# ethernet oam remote-loopback supported
```

```
Switch(config-if)# ethernet oam
```

Customer-edge switch 2 (CE2) configuration:

```
Switch# config terminal
Switch(config)# interface GigabitEthernet1/1
Switch(config-if)# switchport trunk allowed vlan 10
Switch(config-if)# switchport mode trunk
Switch(config-if)# ethernet oam remote-loopback supported
Switch(config-if)# ethernet oam
Switch(config-if)# exit
```

These output examples show provider-edge switch port status of the configuration. Port status shows as *UP* at both switches.

Switch PE1:

```
Switch# show ethernet cfm maintenance points remote
MPID Level Mac Address      Vlan PortState InGressPort      Age(sec) Service ID
101 * 4      0015.633f.6900 10   UP          Gi1/1            27      blue
```

Switch PE2:

```
Switch# show ethernet cfm maintenance points remote
MPID Level Mac Address      Vlan PortState InGressPort      Age(sec) Service ID
100 * 4      0012.00a3.3780 10   UP          Gi1/1            8       blue
Total Remote MEPs: 1
```

This example shows the output when you start remote loopback on CE1 (or PE1). The port state on the remote PE switch shows as *Test* and the remote CE switch enters into error-disable mode.

```
Switch# ethernet oam remote-loopback start interface gigabitethernet 1/1
it is a intrusive loopback.
Therefore, while you test Ethernet OAM MAC connectivity,
you will be unable to pass traffic across that link.
Proceed with Remote Loopback? [confirm]
```

Switch PE1:

```
Switch# show ethernet cfm maintenance points remote
MPID Level Mac Address      Vlan PortState InGressPort      Age(sec) Service ID
101 * 4      0015.633f.6900 10   UP          Gi1/1            27      blue
```

Switch PE2:

```
Switch# show ethernet cfm maintenance points remote
MPID Level Mac Address      Vlan PortState InGressPort      Age(sec) Service ID
100 * 4      0012.00a3.3780 10   TEST       Gi1/1            8       blue
Total Remote MEPs: 1
```

In addition, if you shut down the CE1 interface that connects to PE1, the remote PE2 port shows a PortState of *Down*.



Configuring Y.1731 (AIS and RDI)

The Catalyst 4500 series switch supports Y.1731 Ethernet Alarm Indication Signal function (ETH-AIS) and Ethernet Remote Defect Indication function (ETH-RDI) to provide fault and performance management for service providers in large networks. This chapter describes how to configure Y.1731 ETH-AIS and ETH-RDI.

This chapter contains these sections:

- [AIS and RDI Terminology, page 78-1](#)
- [About Y.1731, page 78-2](#)
- [Configuring Y.1731, page 78-4](#)
- [Displaying Y.1731 Information, page 78-5](#)

For complete command and configuration information for Y.1731, see the Cisco IOS feature module at this URL:

<http://www.cisco.com/en/US/docs/ios-xml/ios/cether/configuration/12-2sr/ce-cfm-y1731.html>

AIS and RDI Terminology

Term	Definition
CC	Ethernet OAM Continuity Check
CCM	Ethernet OAM Continuity Check Message
CCDB	Ethernet OAM Continuity Check Database
CFM	Ethernet Connectivity Fault Management
EI	Ethernet Infrastructure or EVC Infrastructure
EVC	Ethernet Virtual Circuit
LMEP	Local Mep
MEP	Maintenance Endpoint
MIP	Maintenance Intermediate Point
OAM	Operations Administration and Maintenance
Service VLAN	The VLAN tag that uniquely identifies a Customer Service Instance within the Provider network

About Y.1731

These sections contain conceptual information about Y.1731:

- [Server MEP, page 78-2](#)
- [Alarm Indication Signal, page 78-2](#)
- [Ethernet Remote Defect Indication, page 78-3](#)

The advent of Ethernet as a metropolitan and WAN technology imposes a new set of Operations, Administration, and Maintenance (OAM) requirements on Ethernet's traditionally Enterprise-oriented functions. The expansion of this technology into the larger and more complex wider user base makes operational management of link uptime crucial. Isolating and responding to failures quickly directly affects the competitiveness of the service provider.

Server MEP

A Server MEP is a combined function of the server layer termination function and the server and ETH adaptation function. It issues frames with ETH-AIS information upon detecting a defect at the Server layer by the server layer termination function or the adaptation function.

A Virtual MEP represents the logical termination point of Connectivity Fault Management (CFM) MAs defined at the link or transport layer. A server MEP runs or is defined at Maintenance Level -1. For example, you could associate an outward-facing Server MEP with each termination point of IEEE 802.3ah OAM, or with each termination point of MPLS PW OAM.

Alarm Indication Signal

ETH-AIS allows you to suppress alarms when defects are detected at the server (sub) layer. Because of STP's ability to restore, you would not expect to apply ETH-AIS in the STP environments. For the Catalyst 4500 Metro switch, an administrator can enable and disable AIS in the STP environment.

You can enable or disable transmission of frames with ETH-AIS information on a MEP (or on a Server MEP).

You also can issue frames with ETH-AIS information at the client Maintenance Level by a MEP, including a Server MEP upon detecting defect conditions.

The defect conditions may include:

- Signal fail conditions with ETH-CC enabled
- AIS condition with ETH-CC disabled

For multipoint ETH connectivity, a MEP cannot determine the specific server (sub) layer entity that has encountered defect conditions upon receiving a frame with ETH-AIS information. More importantly, it cannot determine the associated subset of its peer MEPs for which it should suppress alarms because the received ETH-AIS information does not contain that information. When a MEP receives a frame with ETH-AIS information, it suppresses alarms for all peer MEPs whether there is still connectivity or not.

For a point-to-point ETH connection, however, a MEP has only one peer MEP. There is no ambiguity regarding the peer MEP for which it should suppress alarms when it receives the ETH-AIS information.

Only a MEP, including a Server MEP, is configured to issue frames with ETH-AIS information. Once the MEP detects a defect condition, it immediately starts transmitting periodic frames with ETH-AIS information at a configured client maintenance level. We send the AIS frames at a configured MIP level

for an interface. A MEP continues to transmit periodic frames with ETH-AIS information until the defect condition is removed. Upon receiving a frame with ETH-AIS information, a MEP detects AIS condition and suppresses loss of continuity alarms associated with all its peer MEPs. A MEP resumes loss of continuity alarm generation upon detecting loss of continuity defect conditions in the absence of AIS condition.

Ethernet Remote Defect Indication

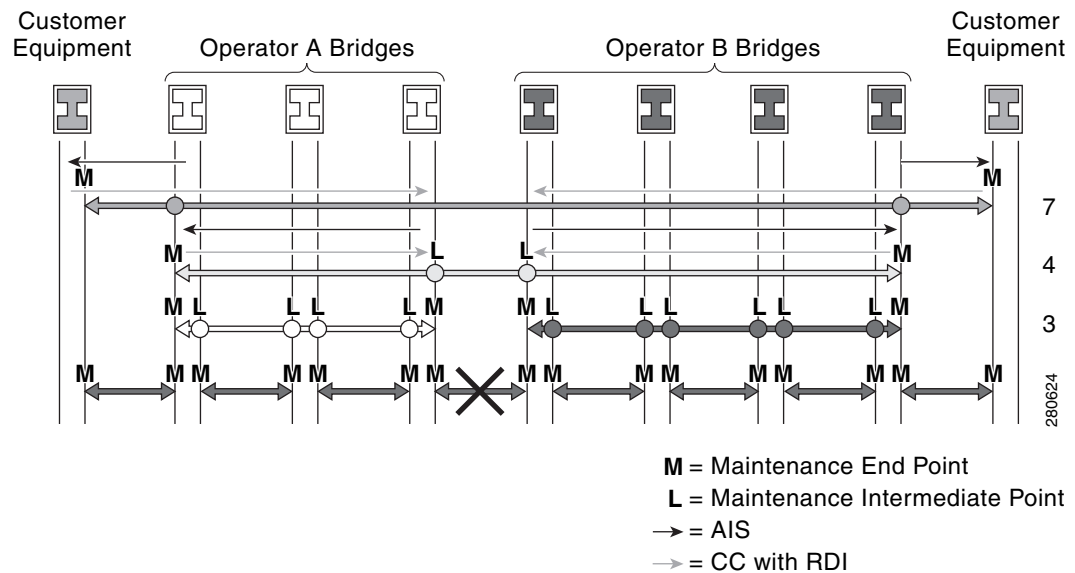
A MEP can use ETH-RDI to notify its peer MEPs that it detects a defect condition. ETH-RDI is used only when ETH-CC transmission is enabled.

ETH-RDI has the following two applications:

- Single-ended fault management—The receiving MEP detects an RDI defect condition, which is correlated with other defect conditions in this MEP and may cause a fault. The absence of received ETH-RDI information in a single MEP indicates the absence of defects in the entire maintenance.
- Contribution to far-end performance monitoring— It reflects a defect condition in the far-end which serves as input to the performance monitoring process.

A MEP that is in a defect condition transmits frames with ETH-RDI information. A MEP, upon receiving frames with ETH-RDI information, determines that its peer MEP has encountered a defect condition. For multipoint ETH connectivity, however, a MEP, upon receiving frames with ETH-RDI information, cannot determine the associated subset of its peer MEPs with which the MEP transmitting RDI information encounters defect conditions. It is because the transmitting MEP itself does not always have that information.

Figure 78-1 Generating and Propagating AIS Messages Upon a Defect (Link Fail)



Configuring Y.1731



Note

Y.1731 is enabled by default.

These sections are included:

- [Y.1731 Configuration Guidelines, page 78-4](#)
- [Configuring AIS Parameters, page 78-4](#)
- [Clearing MEP from the AIS Defect Condition, page 78-5](#)
- [Clearing SMEP from the AIS Defect Condition, page 78-5](#)

Y.1731 Configuration Guidelines

Configuration guidelines and restrictions for Y.1731 include the following:

- Because of STP’s restoration capability, do not expect ETH-AIS to be applied in the STP environments.
- AIS is enabled by default on a CFM maintenance domain. The following section illustrates the commands you can use to disable AIS on a maintenance domain. Likewise, RDI is a flag bit in the CC message. Provided CC transmission is enabled, the present RDI flag of the CC message is set to true or false.

Configuring AIS Parameters

To set the parameters for AIS, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch (config)# ethernet cfm ais domain name vlan range	Enters <i>config-ais-mep-cfm</i> submode to configure parameters for all local Meps belonging to that per Maintenance Association (MA). <i>name</i> refers to the domain name. <i>range</i> represents VLAN IDs 100, 200-300, 400, 500, or 1-4095 For the domain name vlan range command only configurations that are VLAN disjoint or congruent are accepted.
Step 3	Switch(config-ais-mep-cfm)# disable	Disables AIS transmission.
Step 4	Switch(config-ais-mep-cfm)# period period	Sets the AIS transmission period.
Step 5	Switch(config-ais-mep-cfm)# level level	Establishes a maintenance level to send AIS frames for MEPs belonging to MA. Valid levels are 0 to 7.

	Command	Purpose
Step 6	Switch(config-ais-mep-cfm) # expiry-threshold threshold	Sets the AIS expiry threshold. By default, expiry threshold is 3.5. With this CLI we can change the expiry threshold parameter for MA.
Step 7	Switch(config-ais-mep-cfm) # express alarm	Configures alarm suppression when an AIS message causes the MEP enters an AIS defect condition.
Step 8	Switch(config-ais-mep-cfm) # exit	Returns to global configuration.
Step 9	Switch(config)# [no] ethernet cfm ais link-status global	Enters <i>config-ais-link-cfm</i> submode, enabling you to configure parameters required to follow when link status goes down.
Step 10	Switch(config-if)# [no] ethernet cfm ais link-status period period	Configures ETH-AIS transmission period generated by the link-status on the interface.
Step 11	Switch(config-if)# [no] ethernet cfm ais link-status level level	Configures maintenance level to send AIS frames transmitted by the link-status on the interface.
Step 12	end	Returns to privileged EXEC mode.
Step 13	Switch# show ethernet cfm smep interface name Switch# show ethernet cfm maintenance-points local detail Switch# show ethernet cfm maintenance-points remote detail Switch# show ethernet cfm error	Verifies the configuration.
Step 14	Switch# show running-config	Verifies your entries.
Step 15	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Use the **no** versions of the commands to remove the configuration or return to the default configurations.

Clearing MEP from the AIS Defect Condition

To clear the MEP, enter one of the following commands:

```
Switch# clear ethernet cfm ais domain domain name mpid local mpid vlan vlan#
Switch# clear ethernet cfm ais domain domain name mpid local mpid evc evc_name
```

Clearing SMEP from the AIS Defect Condition

To clear the CSMP, enter one of the following commands:

```
Switch# clear ethernet cfm ais link-status interface interface_name
Switch# clear ethernet cfm error
```



Note

This operation also clears all error conditions including AIS.

Displaying Y.1731 Information

To display Y.1731 information, you can use the following commands ([Table 78-1](#)).

Table 78-1 **Displaying Y.1731 Information**

Command	Purpose
<code>show ethernet cfm maintenance-point local detail</code>	Displays AIS status and defect condition on local maintenance points.
<code>show ethernet cfm smep [interface <name>]</code>	Displays AIS status and defect condition on SMEP.
<code>show ethernet cfm error</code>	Displays errors due to AIS defect condition.
<code>show ethernet cfm maintenance-points remote [detail]</code>	Displays AIS status and defect Condition on remote maintenance points.

This example shows how to track the RDI defect and to verify the configuration parameters:

```
Switch# show ethernet cfm main local detail
MEP Settings:
-----
MPID: 1109
DomainName: PROVIDER_DOMAIN
Level: 4
Direction: I
EVC: evc_1
Interface: Gi3/1
CC-Status: Enabled
MAC: 001b.d550.91fd
Defect Condition: No Defect
presentRDI: FALSE (RDI defect is NOT present)
AIS-Status: Enabled
AIS Period: 60000(ms)
AIS Expiry Threshold: 3.5
Level to transmit AIS: Default
Suppress Alarm configuration: Enabled
Suppressing Alarms: No

MIP Settings:
-----
Level Type Port MAC
7 MIP Gi3/1 001b.d550.91fd
4 MIP Tel/2 001b.d550.91fd
Switch#
*Feb 18 05:40:35.659: %ETHER_CFM-6-ENTER_AIS: local mep with mpid 1109 level 4 id 100 dir
I Interface GigabitEthernet3/1 enters AIS defect condition (gi3/2 enters AIS state)
Switch# show ethernet cfm main local detail
MEP Settings:
-----
MPID: 1109
DomainName: PROVIDER_DOMAIN
Level: 4
Direction: I
EVC: evc_1
Interface: Gi3/1
CC-Status: Enabled
MAC: 001b.d550.91fd
Defect Condition: AIS
presentRDI: TRUE (RDI defect IS present)
AIS-Status: Enabled
AIS Period: 60000(ms)
AIS Expiry Threshold: 3.5
Level to transmit AIS: Default
Suppress Alarm configuration: Enabled
Suppressing Alarms: Yes
```

MIP Settings:

Level	Type	Port	MAC
7	MIP	Gi3/1	001b.d550.91fd
4	MIP	Te1/2	001b.d550.91fd

Switch# **show ethernet cfm error**

Level	Vlan	MPID	Remote MAC	Reason	Service ID
4	100	2101	001d.4566.aa3d	0 lifetime TLV	customerX
4	100	-	001b.d550.91fd	Receive AIS	customerX

Switch#

*Feb 18 05:51:08.567: %ETHER_CFM-6-EXIT_AIS: local mep with mpid 1109 level 4 id 100 dir I
Interface GigabitEthernet3/1 exited AIS defect condition (gi3/1 exits AIS state)

Switch# **show ethernet cfm main local detail**

MEP Settings:

MPID: 1109

DomainName: PROVIDER_DOMAIN

Level: 4

Direction: I

EVC: evc_1

Interface: Gi3/1

CC-Status: Enabled

MAC: 001b.d550.91fd

Defect Condition: No Defect

presentRDI: FALSE **(RDI defect is not present anymore)**

AIS-Status: Enabled

AIS Period: 60000(ms)

AIS Expiry Threshold: 3.5

Level to transmit AIS: Default

Suppress Alarm configuration: Enabled

Suppressing Alarms: No

MIP Settings:

Level	Type	Port	MAC
7	MIP	Gi3/1	001b.d550.91fd
4	MIP	Te1/2	001b.d550.91fd

Switch#



Configuring Call Home

This chapter describes how to configure the Call Home feature in Catalyst 4500 Series Switch.

This chapter includes the following sections:

- [About Call Home, page 79-1](#)
- [Obtaining Smart Call Home, page 79-2](#)
- [Configuring Call Home, page 79-3](#)
- [Configuring Contact Information, page 79-4](#)
- [Configuring Destination Profiles, page 79-5](#)
- [Subscribing to Alert Groups, page 79-6](#)
- [Configuring General E-Mail Options, page 79-9](#)
- [Enabling Call Home, page 79-10](#)
- [Testing Call Home Communications, page 79-10](#)
- [Configuring and Enabling Smart Call Home, page 79-13](#)
- [Displaying Call Home Configuration Information, page 79-13](#)
- [Call Home Default Settings, page 79-18](#)
- [Alert Group Trigger Events and Commands, page 79-18](#)
- [Message Contents, page 79-21](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About Call Home

Call Home provides e-mail-based and web-based notification of critical system events. A versatile range of message formats are available for optimal compatibility with pager services, standard e-mail, or XML-based automated parsing applications. Common uses of this feature may include direct paging of

a network support engineer, e-mail notification to a Network Operations Center, XML delivery to a support website, and utilization of Cisco Smart Call Home services for direct case generation with the Cisco Systems Technical Assistance Center (TAC).

The Call Home feature can deliver alert messages containing information on configuration, diagnostics, environmental conditions, inventory, and syslog events.

The Call Home feature can deliver alerts to multiple recipients, referred to as *Call Home destination profiles*, each with configurable message formats and content categories. A predefined destination profile is provided for sending alerts to the Cisco TAC (callhome@cisco.com), and you also can define your own destination profiles.

Flexible message delivery and format options make it easy to integrate specific support requirements.

The Call Home feature offers the following advantages:

- Multiple message-format options:
 - Short Text—Suitable for pagers or printed reports.
 - Plain Text—Full formatted message information suitable for human reading.
 - XML—Matching readable format using Extensible Markup Language (XML) and Adaptive Markup Language (AML) document type definitions (DTDs). The XML format enables communication with the Cisco TAC.
- Multiple concurrent message destinations.
- Multiple message categories including configuration, diagnostics, environmental conditions, inventory, and syslog events.
- Filtering of messages by severity and pattern matching.
- Scheduling of periodic message sending.

Obtaining Smart Call Home

If you have a service contract directly with Cisco Systems, you can register your devices for the Smart Call Home service. Smart Call Home provides fast resolution of system problems by analyzing Call Home messages sent from your devices and providing background information and recommendations. For issues that can be identified as known, particularly GOLD diagnostics failures, Automatic Service Requests will be generated with the Cisco TAC.

Smart Call Home offers the following features:

- Boot-up diagnostics alerts for line cards and supervisor engines in the chassis.
- Analysis of Call Home messages from your device, and where appropriate Automatic Service Request generation, routed to the appropriate TAC team, including detailed diagnostic information to speed problem resolution.
- Secure message transport directly from your device or through a downloadable Transport Gateway (TG) aggregation point. You can use a TG aggregation point in cases requiring support for multiple devices or in cases where security requirements mandate that your devices may not be connected directly to the Internet.
- Web-based access to Call Home messages and recommendations, inventory and configuration information for all Call Home devices. Provides access to associated field notices, Security Advisories, and End-of-Life information.

You need to register the following items:

- The SMARTnet contract number for your switch
- Your e-mail address
- Your Cisco.com ID

For detailed information on Smart Call Home, refer to the Smart Call Home page at this URL:

http://supportforums.cisco.com/community/netpro/solutions/smart_services/smartcallhome

Configuring Call Home

How you configure Call Home depends on how you intend to use the feature. Consider the following information before you configure Call Home:

- At least one destination profile (predefined or user-defined) must be configured. The destination profile(s) used depends on whether the receiving entity is a pager, e-mail, or automated service such as Cisco Smart Call Home.
 - If the destination profile uses e-mail message delivery, you must specify a Simple Mail Transfer Protocol (SMTP) server.
 - If the destination profile uses secure HTTP (HTTPS) message transport, you must configure a trustpoint certificate authority (CA).
- The contact e-mail, phone, and street address information should be configured so that the receiver can determine the origin of messages received.
- The switch must have IP connectivity to an e-mail server or the destination HTTP server using the **ip domain name** command.
- If Cisco Smart Call Home is used, an active service contract must cover the device being configured.

To configure Call Home, follow these steps:

-
- | | |
|---------------|--|
| Step 1 | Configure your site's contact information. |
| Step 2 | Configure destination profiles for each of your intended recipients. |
| Step 3 | Subscribe each destination profile to one or more alert groups, and set alert options. |
| Step 4 | Configure e-mail settings or HTTPS settings (including CA certificate), depending on the transport method. |
| Step 5 | Enable the Call Home feature. |
| Step 6 | Test Call Home messages. |
-



Tip

From the Smart Call Home web application, you can download a basic configuration script to assist you in the configuration of the Call Home feature for use with Smart Call Home and the Cisco TAC. The script will also assist in configuring the trustpoint CA for secure communications with the Smart Call Home service. The script, provided on an as-is basis, can be downloaded from this URL:

http://supportforums.cisco.com/community/netpro/solutions/smart_services/smartcallhome

Configuring Contact Information

Each switch must include a contact e-mail address. You can optionally include a phone number, street address, contract ID, customer ID, and site ID.

To assign the contact information, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters configuration mode.
Step 2	Switch(config)# call-home	Enters the Call Home configuration submode.
Step 3	Switch(cfg-call-home)# contact-email-addr <i>email-address</i>	Assigns the customer's e-mail address. Enter up to 200 characters in e-mail address format with no spaces.
Step 4	Switch(cfg-call-home)# phone-number <i>+phone-number</i>	(Optional) Assigns the customer's phone number. Note The number must begin with a plus (+) prefix, and may contain only dashes (-) and numbers. Enter up to 16 characters. If you include spaces, you must enclose your entry in quotes ("").
Step 5	Switch(cfg-call-home)# street-address <i>street-address</i>	(Optional) Assigns the customer's street address where RMA equipment can be shipped. Enter up to 200 characters. If you include spaces, you must enclose your entry in quotes ("").
Step 6	Switch(cfg-call-home)# customer-id <i>text</i>	(Optional) Identifies the customer ID. Enter up to 64 characters. If you include spaces, you must enclose your entry in quotes ("").
Step 7	Switch(cfg-call-home)# site-id <i>text</i>	(Optional) Identifies the customer site ID. Enter up to 200 characters. If you include spaces, you must enclose your entry in quotes ("").
Step 8	Switch(cfg-call-home)# contract-id <i>text</i>	(Optional) Identifies the customer's contract ID for the switch. Enter up to 64 characters. If you include spaces, you must enclose your entry in quotes ("").

This example shows the configuration of contact information:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# call-home
Switch(cfg-call-home)# contact-email-addr username@example.com
Switch(cfg-call-home)# phone-number +1-800-555-4567
Switch(cfg-call-home)# street-address "1234 Picaboo Street, Any city, Any state, 12345"
Switch(cfg-call-home)# customer-id Customer1234
Switch(cfg-call-home)# site-id Site1ManhattanNY
Switch(cfg-call-home)# contract-id Company1234
Switch(cfg-call-home)# exit
Switch(config)#
```

Configuring Destination Profiles

A destination profile contains the required delivery information for an alert notification. At least one destination profile is required. You can configure multiple destination profiles of one or more types.

You can use the predefined destination profile or define a desired profile. If you define a new destination profile, you must assign a profile name.



Note

If you use the Cisco Smart Call Home service, the destination profile must use the XML message format.

You can configure the following attributes for a destination profile:

- **Profile name**—A string that uniquely identifies each user-defined destination profile. The profile name is limited to 31 characters and is not case-sensitive. You cannot use **all** as a profile name.
- **Transport method**—The transport mechanism, either e-mail or HTTP (including HTTPS), for delivery of alerts.
 - For user-defined destination profiles, e-mail is the default, and you can enable either or both transport mechanisms. If you disable both methods, e-mail will be enabled.
 - For the predefined Cisco TAC profile, you can enable either transport mechanism, but not both.
- **Destination address**—The actual address related to the transport method to which the alert should be sent.
- **Message formatting**—The message format used for sending the alert.
 - For user-defined destination profiles, the format options are long-text, short-text, or XML. The default is XML.
 - For the predefined Cisco TAC profile, only XML is allowed.
- **Message size**—The maximum destination message size. The valid range is 50 to 3,145,728 bytes and the default is 3,145,728 bytes.

To create and configure a destination profile, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters configuration mode.
Step 2	Switch(config)# call-home	Enters the Call Home configuration submode.
Step 3	Switch(cfg-call-home)# profile name	Enters the Call Home destination profile configuration submode for the specified destination profile. If the specified destination profile does not exist, it is created.
Step 4	Switch(cfg-call-home-profile)# [no] destination transport-method {email http}	(Optional) Enables the message transport method. The no option disables the method.
Step 5	Switch(cfg-call-home-profile)# destination address {email email-address http url}	Configures the destination e-mail address or URL to which Call Home messages will be sent. Note When entering a destination URL, include either http:// or https:// , depending on whether the server is a secure server. If the destination is a secure server, you must also configure a trustpoint CA.
Step 6	Switch(cfg-call-home-profile)# destination preferred-msg-format {long-text short-text xml}	(Optional) Configures a preferred message format. The default is XML.

	Command	Purpose
Step 7	Switch(cfg-call-home-profile)# destination message-size-limit <i>bytes</i>	(Optional) Configures a maximum destination message size for the destination profile.
Step 8	Switch(cfg-call-home-profile)# active	Enables the destination profile. By default, the profile is enabled when it is created.
Step 9	Switch(cfg-call-home-profile)# exit	Exits the Call Home destination profile configuration submode and returns to the Call Home configuration submode.
Step 10	Switch(cfg-call-home)# end	Returns to privileged EXEC mode.
Step 11	Switch# show call-home profile { <i>name</i> all }	Displays the destination profile configuration for a specified profile or all configured profiles.

Copying a Destination Profile

To create a new destination profile by copying an existing profile, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters configuration mode.
Step 2	Switch(config)# call-home	Enters the Call Home configuration submode.
Step 3	Switch(cfg-call-home)# copy profile <i>source-profile target-profile</i>	Creates a new destination profile with the same configuration settings as the existing destination profile.

Subscribing to Alert Groups

An alert group is a predefined subset of Call Home alerts supported on the switch. Different types of Call Home alerts are grouped into different alert groups depending on their type. These alert groups are available:

- Configuration
- Diagnostic
- Environment
- Inventory
- Syslog

The triggering events for each alert group are listed in the [“Alert Group Trigger Events and Commands” section on page 79-18](#), and the contents of the alert group messages are listed in the [“Message Contents” section on page 79-21](#).

You can select one or more alert groups to be received by a destination profile.



Note

A Call Home alert is only sent to destination profiles that have subscribed to the alert group containing that Call Home alert. In addition, the alert group must be enabled.

To subscribe a destination profile to an alert group, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters configuration mode.
Step 2	Switch(config)# call-home	Enters Call Home configuration submode.
Step 3	Switch(cfg-call-home)# alert-group { all configuration diagnostic environment inventory syslog }	Enables the specified alert group. Use the keyword all to enable all alert groups. By default, all alert groups are enabled.
Step 4	Switch(cfg-call-home)# profile name	Enters the Call Home destination profile configuration submode for the specified destination profile.
Step 5	Switch(cfg-call-home-profile)# subscribe-to-alert-group configuration [periodic { daily <i>hh:mm</i> monthly <i>date hh:mm</i> weekly <i>day hh:mm</i> }]	Subscribes this destination profile to the Configuration alert group. The Configuration alert group can be configured for periodic notification, as described in the “ Configuring Periodic Notification ” section on page 79-8.
	Switch(cfg-call-home-profile)# subscribe-to-alert-group all	Subscribes to all available alert groups.
Step 6	Switch(cfg-call-home-profile)# subscribe-to-alert-group diagnostic [severity <i>catastrophic</i> <i>disaster</i> <i>fatal</i> <i>critical</i> <i>major</i> <i>minor</i> <i>warning</i> <i>notification</i> <i>normal</i> <i>debugging</i>]	Subscribes this destination profile to the Diagnostic alert group. The Diagnostic alert group can be configured to filter messages based on severity, as described in the “ Configuring Message Severity Threshold ” section on page 79-8.
Step 7	Switch(cfg-call-home-profile)# subscribe-to-alert-group environment [severity <i>catastrophic</i> <i>disaster</i> <i>fatal</i> <i>critical</i> <i>major</i> <i>minor</i> <i>warning</i> <i>notification</i> <i>normal</i> <i>debugging</i>]	Subscribes this destination profile to the Environment alert group. The Environment alert group can be configured to filter messages based on severity, as described in the “ Configuring Message Severity Threshold ” section on page 79-8.
Step 8	Switch(cfg-call-home-profile)# subscribe-to-alert-group inventory [periodic { daily <i>hh:mm</i> monthly <i>date hh:mm</i> weekly <i>day hh:mm</i> }]	Subscribes this destination profile to the Inventory alert group. The Inventory alert group can be configured for periodic notification, as described in the “ Configuring Periodic Notification ” section on page 79-8.
Step 9	Switch(cfg-call-home-profile)# subscribe-to-alert-group syslog [severity <i>catastrophic</i> <i>disaster</i> <i>fatal</i> <i>critical</i> <i>major</i> <i>minor</i> <i>warning</i> <i>notification</i> <i>normal</i> <i>debugging</i>] [pattern <i>string</i>]	Subscribes this destination profile to the Syslog alert group. The Syslog alert group can be configured to filter messages based on severity, as described in the “ Configuring Message Severity Threshold ” section on page 79-8. You can specify a pattern to be matched in the syslog message. If the pattern contains spaces, you must enclose it in quotes (“”).
Step 10	Switch(cfg-call-home-profile)# exit	Exits the Call Home destination profile configuration submode.

Configuring Periodic Notification

When you subscribe a destination profile to either the Configuration or the Inventory alert group, you can choose to receive the alert group messages asynchronously or periodically at a specified time. The sending period can be one of the following:

- **Daily**—Specify the time of day to send, using an hour:minute format *hh:mm*, with a 24-hour clock (for example, 14:30).
- **Weekly**—Specify the day of the week and time of day in the format *day hh:mm*, where the day of the week is spelled out (for example, monday).
- **Monthly**—Specify the numeric date, from 1 to 31, and the time of day, in the format *date hh:mm*.

Configuring Message Severity Threshold

When you subscribe a destination profile to the Diagnostic, Environment, or Syslog alert group, you can set a threshold for sending alert group messages based on the message's level of severity. Any message with a value lower than the threshold is not sent to the destination.

The severity threshold is configured using the keywords in [Table 79-1](#), and ranges from catastrophic (level 9, highest level of urgency) to debugging (level 0, lowest level of urgency). If no severity threshold is configured, the default is normal (level 1).



Note

Call Home severity levels differ from the system message logging severity levels.

Table 79-1 Severity and Syslog Level Mapping

Level	Keyword	Syslog Level	Description
9	catastrophic	N/A	Network-wide catastrophic failure
8	disaster	N/A	Significant network impact
7	fatal	Emergency (0)	System unusable
6	critical	Alert (1)	Critical conditions, immediate attention needed
5	major	Critical (2)	Major conditions
4	minor	Error (3)	Minor conditions
3	warning	Warning (4)	Warning conditions
2	notification	Notice (5)	Basic notification and informational messages; possibly independently insignificant
1	normal	Information (6)	Normal event signifying return to normal state
0	debugging	Debug (7)	Debugging messages

Configuring Syslog Pattern Matching

When you subscribe a destination profile to the Syslog alert group, you can optionally specify a text pattern to be matched within each syslog message. If you configure a pattern, a Syslog alert group message will be sent only if it contains the specified pattern and meets the severity threshold. If the pattern contains spaces, you must enclose it in quotes (") when configuring it. You can specify up to five patterns for each destination profile.

Configuring General E-Mail Options

To use the e-mail message transport, you must configure at least one Simple Mail Transfer Protocol (SMTP) e-mail server address. You can configure the from and reply-to e-mail addresses, and you can specify up to four backup e-mail servers. You can also set a rate limit on e-mail or HTTP messages.

Starting with Cisco IOS Release 15.0(2)SG, you can configure the vrf and source interface or source IP address to send the e-mail messages. If you want to configure similar options to send http messages, you must enter the **ip http client source-interface** *interface-name* command where the source-interface can be associated with the vrf you want to set.

To configure general e-mail options, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters configuration mode.
Step 2	Switch(config)# call-home	Enters Call Home configuration submode.
Step 3	Switch(cfg-call-home)# mail-server {ipv4-address name} priority number	Assigns an e-mail server address and its relative priority among configured e-mail servers. Provide either of these: <ul style="list-style-type: none"> The e-mail server's IP address The e-mail server's fully qualified domain <i>name</i> (FQDN) of 64 characters or less. Assign a priority <i>number</i> between 1 (highest priority) and 100 (lowest priority).
Step 4	Switch(cfg-call-home)# sender from email-address	(Optional) Assigns the e-mail address that will appear in the from field in Call Home e-mail messages. If no address is specified, the contact e-mail address is used.
Step 5	Switch(cfg-call-home)# sender reply-to email-address	(Optional) Assigns the e-mail address that will appear in the reply-to field in Call Home e-mail messages.
Step 6	Switch(cfg-call-home)# rate-limit number	(Optional) Specifies a limit on the number of messages sent per minute, from 1 to 60. The default is 20.
Step 7	Switch(cfg-call-home)# vrf vrf-name	(Optional) Specifies the VRF instance to send Call Home e-mail messages. If no VRF is specified, the global routing table is used.
Step 8	Switch(cfg-call-home)# source-interface interface-name	(Optional) Specifies the source interface name to send Call Home e-mail messages. If no source interface name or source ip address is specified, an interface in the routing table is used.
Step 9	Switch(cfg-call-home)# source-ip-address ip-address	(Optional) Specifies the source IP address to send Call Home e-mail messages. If no source IP address or source interface name is specified, an interface in the routing table is used. Note At one time, you can specify the source-interface name or the source-ip-address, but not both.

The following notes apply when configuring general e-mail options:

- Backup e-mail servers can be defined by repeating the **mail-server** command using different priority numbers.

- The **mail-server priority number** parameter can be configured from 1 to 100. The server with the highest priority (lowest priority number) will be tried first.

This example shows the configuration of general e-mail parameters, including a primary and secondary e-mail server:

```
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# call-home
Switch(cfg-call-home)# mail-server smtp.example.com priority 1
Switch(cfg-call-home)# mail-server 192.168.0.1 priority 2
Switch(cfg-call-home)# sender from username@example.com
Switch(cfg-call-home)# sender reply-to username@example.com
Switch(cfg-call-home)# exit
Switch(config)#
```

Enabling Call Home

To enable or disable the Call Home feature, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters configuration mode.
Step 2	Switch(config)# service call-home	Enables the Call Home feature.

Testing Call Home Communications

You can test Call Home communications by sending messages manually using two command types.

- To send a user-defined Call Home test message, use the **call-home test** command.
- To send a specific alert group message, use the **call-home send** command.

Sending a Call Home Test Message Manually

To manually send a Call Home test message, perform this task:

	Command	Purpose
Step 1	Switch# call-home test ["test-message"] profile name	Sends a test message to the specified destination profile. The user-defined test message text is optional, but must be enclosed in quotes (") if it contains spaces. If no user-defined message is configured, a default message will be sent.

This example shows how to manually send a Call Home test message:

```
Switch# call-home test "test of the day" profile Ciscotac1
```

Sending a Call Home Alert Group Message Manually

To manually trigger a Call Home alert group message, perform this task:

	Command	Purpose
Step 1	Switch# call-home send alert-group configuration [profile name]	Sends a configuration alert group message to one destination profile if specified, or to all subscribed destination profiles.
	Switch# call-home send alert-group diagnostic { module number slot/subslot slot/bay } [profile name]	Sends a diagnostic alert group message to the configured destination profile if specified, or to all subscribed destination profiles. You must specify the module or port whose diagnostic information should be sent.
	Switch# call-home send alert-group inventory [profile name]	Sends an inventory alert group message to one destination profile if specified, or to all subscribed destination profiles.

When manually sending Call Home alert group messages, note the following guidelines:

- You can only manually send the configuration, diagnostic, and inventory alert groups.
- When you manually trigger a configuration, diagnostic, or inventory alert group message and you specify a destination profile name, a message is sent to the destination profile regardless of the profile's active status, subscription status, or severity setting.
- When you manually trigger a configuration or inventory alert group message and do not specify a destination profile name, a message is sent to all active profiles that have either a normal or periodic subscription to the specified alert group.
- When you manually trigger a diagnostic alert group message and do not specify a destination profile name, the command will cause the following actions:
 - For any active profile that subscribes to diagnostic events with a severity level of less than minor, a message is sent regardless of whether the module or interface has observed a diagnostic event.
 - For any active profile that subscribes to diagnostic events with a severity level of minor or higher, a message is sent only if the specified module or interface has observed a diagnostic event of at least the subscribed severity level; otherwise, no diagnostic message is sent to the destination profile.

This example shows how to send the configuration alert-group message to the destination profile:

```
Switch# call-home send alert-group configuration
```

This example shows how to send the diagnostic alert-group message to the destination profile for a specific module, slot/subslot, or slot/bay number.

```
Switch# call-home send alert-group diagnostic module 3 5/2
```

This example shows how to send the diagnostic alert-group message to all destination profiles for a specific module, slot/subslot, or slot/bay number.

```
Switch# call-home send alert-group diagnostic module 3 5/2 profile Ciscotac1
```

This example shows how to send the inventory call-home message:

```
Switch# call-home send alert-group inventory
```

Sending a Request for an Analysis and Report

You can use the **call-home request** command to submit information about your system to Cisco in order to receive helpful information specific to your system. You can request a variety of reports, including security alerts, known bugs, best practices, and command references.

To submit a request for report and analysis information from the Cisco Output Interpreter tool, perform one of these tasks:

Command	Purpose
Switch# call-home request output-analysis "show-command" [profile name] [ccoid user-id]	Sends the output of the specified show command for analysis. The show command must be contained in quotes ("").
Switch# call-home request { config-sanity bugs-list command-reference product-advisory } [profile name] [ccoid user-id]	Sends the output of a predetermined set of commands for analysis such as show running-config all , show version , or show module commands. In addition, the call-home request product-advisory command includes all inventory alert group commands. The keyword specified after the call home request command specifies the type of report required.

When manually sending a Call Home report and analysis request, note the following guidelines:

- If you specify a **profile** name value, the request is sent to the profile. If you do not specify a profile name, the request is sent to the Cisco TAC profile. The recipient profile does not need to be enabled for the Call Home request. The profile should specify the e-mail address where the transport gateway is configured so that the request message can be forwarded to the Cisco TAC and you can receive the reply from the Smart Call Home service.
- The **ccoid** user-id value is the registered identifier of the Smart Call Home user. If you specify a user-id, the response is sent to the e-mail address of the registered user. If do not specify a user-id, the response is sent to the contact e-mail address of the device.
- Based on the keyword specifying the type of report requested, the following information is returned:
 - **config-sanity**—Information on best practices as related to the current running configuration
 - **bugs-list**—Known bugs in the running version and in the currently applied features
 - **command-reference**—Reference links to all commands in the running configuration
 - **product-advisory**—Product Security Incident Response Team (PSIRT) notices, End of Life (EOL) or End of Sales (EOS) notices, or field notices (FN) that may affect devices in your network

This example shows a request for analysis of a user-specified **show** command:

```
Switch# call-home request output-analysis "show diagnostic result module all" profile TG
```

Sending the Output of a Command

You can use the **call-home send** command to execute a CLI command and e-mail the command output to Cisco or to an e-mail address that you specify.

To execute a CLI command and e-mail the command output, perform this task:

Command	Purpose
Switch# call-home send " <i>command</i> " [email <i>email-addr</i>] [service-number <i>SR</i>]	Executes the specified CLI command and e-mails the output.

When sending the output of a command, note the following guidelines:

- The specified CLI command can be any run command, including commands for all modules. The command must be contained in quotes ("").
- If an e-mail address is specified, the command output will be sent to that address. If no e-mail address is specified, the output will be sent to the Cisco TAC (attach@cisco.com). The e-mail will be sent in long text format with the service number, if specified, in the subject line.
- The service number is required only if no e-mail address is specified, or if a Cisco TAC e-mail address is specified.

This example shows how to send the output of a CLI command to a user-specified e-mail address:

```
Switch# call-home send "show diagnostic result module all" email support@example.com
```

Configuring and Enabling Smart Call Home

For application and configuration information of the Cisco Smart Call Home service, see the “FastStart” section of the *Smart Call Home User Guide* at this location:

<http://www.cisco.com/go/smartcall/>

The user guide includes configuration examples for sending Smart Call Home messages directly from your device or through a transport gateway (TG) aggregation point. You can use a TG aggregation point in cases requiring support for multiple devices or in cases where security requirements mandate that your devices may not be connected directly to the Internet.

Because the Smart Call Home service uses HTTPS as the transport method, you must also configure its CA as a trustpoint, as described in the *Smart Call Home User Guide*.

Displaying Call Home Configuration Information

To display the configured Call Home information, perform these tasks:

Command	Purpose
Switch# show call-home	Displays the Call Home configuration in summary.
Switch# show call-home detail	Displays the Call Home configuration in detail.
Switch# show call-home alert-group	Displays the available alert groups and their status.

Command	Purpose
Switch# show call-home mail-server status	Checks and displays the availability of the configured e-mail server(s).
Switch# show call-home profile {all name}	Displays the configuration of the specified destination profile. Use the keyword all to display the configuration of all destination profiles.
Switch# show call-home statistics	Displays the statistics of Call Home events.

Examples 79-1 to 79-7 show the results when using different options of the **show call-home** command.

Example 79-1 Configured Call Home Information

```
Switch# show call-home
call home feature : disable
  call home message's from address: switch@example.com
  call home message's reply-to address: support@example.com

vrf for call-home messages: Not yet set up

contact person's email address: technical@example.com

contact person's phone number: +1-408-555-1234
street address: 1234 Picaboo Street, Any city, Any state, 12345
customer ID: ExampleCorp
contract ID: X123456789
site ID: SantaClara
source ip address: Not yet set up
source interface: Not yet set up
Mail-server[1]: Address: smtp.example.com Priority: 1
Mail-server[2]: Address: 192.168.0.1 Priority: 2
Rate-limit: 20 message(s) per minute

Available alert groups:
  Keyword                State   Description
  -----
  configuration           Disable configuration info
  diagnostic              Disable diagnostic info
  environment             Disable environmental info
  inventory               Enable  inventory info
  syslog                  Disable syslog info

Profiles:
  Profile Name: campus-noc
  Profile Name: CiscoTAC-1

Switch#
```

Example 79-2 Configured Call Home Information in Detail

```
Switch# show call-home detail
Current call home settings:
call home feature : disable
  call home message's from address: switch@example.com
  call home message's reply-to address: support@example.com

vrf for call-home messages: Not yet set up

contact person's email address: technical@example.com
```

```

contact person's phone number: +1-408-555-1234
street address: 1234 Picaboo Street, Any city, Any state, 12345
customer ID: ExampleCorp
contract ID: X123456789
site ID: SantaClara
source ip address: Not yet set up
source interface: Not yet set up
Mail-server[1]: Address: smtp.example.com Priority: 1
Mail-server[2]: Address: 192.168.0.1 Priority: 2
Rate-limit: 20 message(s) per minute

```

Available alert groups:

Keyword	State	Description
configuration	Disable	configuration info
diagnostic	Disable	diagnostic info
environment	Disable	environmental info
inventory	Enable	inventory info
syslog	Disable	syslog info

Profiles:

Profile Name: campus-noc

```

Profile status: ACTIVE
Preferred Message Format: long-text
Message Size Limit: 3145728 Bytes
Transport Method: email
Email address(es): noc@example.com
HTTP address(es): Not yet set up

```

Alert-group	Severity
inventory	normal

Syslog-Pattern	Severity
N/A	N/A

Profile Name: CiscoTAC-1

```

Profile status: ACTIVE
Preferred Message Format: xml
Message Size Limit: 3145728 Bytes
Transport Method: email
Email address(es): callhome@cisco.com
HTTP address(es): https://tools.cisco.com/its/service/oddce/services/DDCEService

```

Periodic configuration info message is scheduled every 1 day of the month at 09:27

Periodic inventory info message is scheduled every 1 day of the month at 09: 12

Alert-group	Severity
diagnostic	minor
environment	warning
inventory	normal

Syslog-Pattern	Severity
.*	major

Switch#

Example 79-3 Available Call Home Alert Groups

Switch# **show call-home alert-group**

Available alert groups:

Keyword	State	Description
---------	-------	-------------

```

-----
configuration          Disable configuration info
diagnostic             Disable diagnostic info
environment            Disable environmental info
inventory              Enable inventory info
syslog                 Disable syslog info

```

Switch#

Example 79-4 E-Mail Server Status Information

```
Switch# show call-home mail-server status
Please wait. Checking for mail server status ...
```

```
Translating "smtp.example.com"
Mail-server[1]: Address: smtp.example.com Priority: 1 [Not Available]
Mail-server[2]: Address: 192.168.0.1 Priority: 2 [Not Available]
```

Switch#

Example 79-5 Information for All Destination Profiles (Predefined and User-Defined)

```
Switch# show call-home profile all
```

```
Profile Name: campus-noc
  Profile status: ACTIVE
  Preferred Message Format: long-text
  Message Size Limit: 3145728 Bytes
  Transport Method: email
  Email address(es): noc@example.com
  HTTP address(es): Not yet set up

Alert-group          Severity
-----
inventory            normal

Syslog-Pattern       Severity
-----
N/A                  N/A

Profile Name: CiscoTAC-1
  Profile status: ACTIVE
  Preferred Message Format: xml
  Message Size Limit: 3145728 Bytes
  Transport Method: email
  Email address(es): callhome@cisco.com
  HTTP address(es): https://tools.cisco.com/its/service/oddce/services/DDCEService

Periodic configuration info message is scheduled every 1 day of the month at 09:27

Periodic inventory info message is scheduled every 1 day of the month at 09:12

Alert-group          Severity
-----
diagnostic           minor
environment           warning
inventory            normal

Syslog-Pattern       Severity
-----
.*                   major
```

Switch#

Example 79-6 Information for a User-Defined Destination Profile

```
Switch# show call-home profile CiscoTAC-1
Profile Name: CiscoTAC-1
  Profile status: INACTIVE
  Preferred Message Format: xml
  Message Size Limit: 3145728 Bytes
  Transport Method: email
  Email address(es): callhome@cisco.com
  HTTP address(es): https://tools.cisco.com/its/service/oddce/services/DDCEService

Periodic configuration info message is scheduled every 11 day of the month at 11:25

Periodic inventory info message is scheduled every 11 day of the month at 11:10

Alert-group              Severity
-----
diagnostic               minor
environment              warning
inventory                normal

Syslog-Pattern           Severity
-----
.*                       major
```

Example 79-7 Call Home Statistics

```
Switch# show call-home statistics
```

Message Types	Total	Email	HTTP

Total Success	0	0	0
Config	0	0	0
Diagnostic	0	0	0
Environment	0	0	0
Inventory	0	0	0
SysLog	0	0	0
Test	0	0	0
Request	0	0	0
Send-CLI	0	0	0
Total In-Queue	0	0	0
Config	0	0	0
Diagnostic	0	0	0
Environment	0	0	0
Inventory	0	0	0
SysLog	0	0	0
Test	0	0	0
Request	0	0	0
Send-CLI	0	0	0
Total Failed	0	0	0
Config	0	0	0
Diagnostic	0	0	0
Environment	0	0	0
Inventory	0	0	0
SysLog	0	0	0
Test	0	0	0
Request	0	0	0
Send-CLI	0	0	0

```

Total Ratelimit
  -dropped 0 0 0
Config 0 0 0
Diagnostic 0 0 0
Environment 0 0 0
Inventory 0 0 0
SysLog 0 0 0
Test 0 0 0
Request 0 0 0
Send-CLI 0 0 0

```

Last call-home message sent time: n/a

Call Home Default Settings

Table 79-2 lists the default Call Home settings.

Table 79-2 Default Call Home Settings

Parameters	Default
Call Home feature status	Disabled
User-defined profile status	Active
Predefined Cisco TAC profile status	Inactive
Transport method	E-mail
Message format type	XML
Destination message size for a message sent in long text, short text, or XML format	3,145,728
Alert group status	Enabled
Call Home message severity threshold	1 (normal)
Message rate limit for messages per minute	20

Alert Group Trigger Events and Commands

Call Home trigger events are grouped into alert groups, with each alert group assigned CLI commands to execute when an event occurs. The CLI command output is included in the transmitted message.

Table 79-3 lists the trigger events included in each alert group, including the severity level of each event and the executed CLI commands for the alert group.

Table 79-3 Call Home Alert Groups, Events, and Actions

Alert Group	Call Home Trigger Event	Syslog Event	Severity	Description and CLI Commands Executed
Syslog				Event logged to syslog. (Only sent to TAC if syslog level 0, 1, or 2) CLI commands executed: show logging show inventory
	SYSLOG	LOG_EMERG	7	System is unusable.

Table 79-3 Call Home Alert Groups, Events, and Actions (continued)

Alert Group	Call Home Trigger Event	Syslog Event	Severity	Description and CLI Commands Executed
	SYSLOG	LOG_ALERT	6	Action must be taken immediately.
	SYSLOG	LOG_CRIT	5	Critical conditions.
	SYSLOG	LOG_ERR	4	Error conditions.
	SYSLOG	LOG_WARNING	3	Warning conditions.
	SYSLOG	LOG_NOTICE	2	Normal but signification condition.
	SYSLOG	LOG_INFO	1	Informational.
	SYSLOG	LOG_DEBUG	0	Debug-level messages.
Environmental				<p>Events related to power, fan, and environment sensing elements, such as temperature alarms. (Sent to TAC.)</p> <p>CLI commands executed:</p> <p>show module show environment show logging show power show inventory</p>
	TEMP_FAILURE	TempHigh	5	The temperature of the chassis is above the normal threshold.
	TEMP_FAILURE	Critical Temp	5	The temperature of the chassis has risen above the critical threshold.
	TEMP_FAILURE	Shutdown Temp	5	The temperature of the chassis is very high and the system will be shut down.
	TEMP_FAILURE	Some Temp Sensors Failed	3	Some of the temperature sensors have failed.
	TEMP_FAILURE	All Temp Sensors Failed	5	All temperature sensors have failed.
	TEMP_RECOVER	TempOk	5	The temperature of the chassis is normal.
	POWER_FAILURE	PowerSupplyBad	5	A power supply has failed or has been turned off.
	POWER_RECOVERY	PowerSupplyGood	5	A failed power supply has been fixed.
	POWER_FAILURE	PowerSupplyFanBad	3	A power supply fan has failed.
	POWER_RECOVERY	PowerSupplyFanGood	3	A failed power supply fan has been fixed.
	POWER_RECOVERY	PowerSupplyOutputIncreased	3	A power supply output has increased.
	POWER_FAILURE	PowerSupplyOutputDecreased	3	A power supply output has decreased.

Table 79-3 Call Home Alert Groups, Events, and Actions (continued)

Alert Group	Call Home Trigger Event	Syslog Event	Severity	Description and CLI Commands Executed
	POWER_FAIL URE	InlinePowerSupplyBad	3	Inline power source from a power supply has failed or turned off.
	POWER_FAIL URE	MixedPowerSupplyInC hassis	3	Mixed power supplies have been detected in the chassis.
	POWER_FAIL URE	NotEnoughPowerChass is	6	There is insufficient power to support the system. The system might shut down.
	POWER_ RECOVERY	InlinePowerSupplyGoo d	3	A failed source for inline power has been fixed.
	FANTRAY_FA ILURE	FanTrayPartialFailure	3	Either a fan or thermistors in system fan tray has failed.
	FANTRAY_FA ILURE	FanTrayMismatch	3	The fantray, supervisor, chassis combination is disallowed.
	FANTRAY_FA ILURE	FanTrayBad	5	Fan tray has failed.
	FANTRAY_ RECOVERY	FanTrayGood	3/5	Failed fan tray has been fixed. The severity of the notification depends on the failure which has been recovered from.
	FANTRAY_ FAILURE	InsufficientFantray	6	There are not enough FanTray to support the system. This may be followed by a system shut down.
	CLOCK_ALA RM	ClockSwitchover	2	Clock module has switched over to another clock.
	CLOCK_ALA RM	Clock Faulty	3	The clock module has been found to be faulty.
Inventory				Inventory status should be provided whenever a unit is cold-booted, or when FRUs are inserted or removed. it is considered a noncritical event, and the information is used for status and entitlement. CLI commands executed: show module show version show inventory oid show idprom all show power
	INSERTION	Module	1	A line card or supervisor engine has been inserted into a slot.
	REMOVAL	Module	1	A line card or supervisor engine has been removed from a slot.

Table 79-3 Call Home Alert Groups, Events, and Actions (continued)

Alert Group	Call Home Trigger Event	Syslog Event	Severity	Description and CLI Commands Executed
Diagnostic Failure			1/3/4/5	Events related to standard or intelligent line cards. CLI commands executed: show module show version show inventory show buffers show logging show diagnostic result module x detail show diagnostic result module all
Test	TEST		1	User-generated test message. CLI commands executed: show module show version show inventory
Configuration			1	User-generated request for configuration. CLI commands executed: show module show inventory show version show running-config all show startup-config

Message Contents

The following tables display the content formats of alert group messages:

- [Table 79-4](#) describes the content fields of a short text message.
- [Table 79-5](#) describes the content fields that are common to all long text and XML messages. The fields specific to a particular alert group message are inserted at a point between the common fields. The insertion point is identified in the table.
- [Table 79-6](#) describes the inserted content fields for reactive messages (system failures that require a TAC case) and proactive messages (issues that might result in degraded system performance).
- [Table 79-7](#) describes the inserted content fields for an inventory message.

Table 79-4 Format for a Short Text Message

Data Item	Description
Device identification	Configured device name
Date/time stamp	Time stamp of the triggering event
Error isolation message	Plain English description of triggering event
Alarm urgency level	Error level such as that applied to a system message

Table 79-5 Common Fields for All Long Text and XML Messages

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Time stamp	Date/timestamp of event in ISO time notation: YYYY-MM-DDTHH:MM:SS Note the T in between date and time, and note that the timezone/dst offset from UTC has already been added or subtracted.	CallHome/EventTime
Message name	Name of message.	For short text message only
Message type	Specifically "Call Home".	CallHome/Event/Type
Message group	Specifically "reactive". Optional in this case because default is "reactive".	CallHome/Event/SubType
Severity level	Severity level of message.	Body/Block/Severity
Source ID	This field is used to identify the product type for routing using the workflow engine. it is typically the product family name.	For long test mmessage only
Device ID	Unique Device Identifier (UDI) for end device generating message. This field should empty if the message is nonspecific to a fabric switch. Format: type@Sid@serial Where @:Separator character <ul style="list-style-type: none"> Type: If WS-C4503-E, product model number from backplane SEEPROM. Sid: "C" identifying serial ID as a chassis serial number. Serial: The serial number as identified by the Sid field. Example: "WS-C4503-E@C@SPE4465329F"	CallHome/Customer Data/ ContractData/ DeviceId
Customer ID	Optional user-configurable field used for contract information or other ID by any support service.	CallHome/Customer Data/ ContractData/ CustomerId
Contract ID	Optional user-configurable field used for contract information or other ID by any support service.	/CallHome/ Customer Data/ ContractData/ ContractId
Site ID	Optional user-configurable field used for Cisco-supplied site ID or other data meaningful to alternate support service.	CallHome/ CustomerData/ ContractData/ SiteId

Table 79-5 Common Fields for All Long Text and XML Messages (continued)

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Server ID	<p>If message is generated from fabric switch, UDI of switch. If message is proxied or originated by MA, the MA should overwrite this field with the MA UDI.</p> <p>Format is type@Sid@serial</p> <p>Where</p> <p>@: Separator character</p> <ul style="list-style-type: none"> Type: If WS-C4510R, product model number from backplane SEEPROM. Sid: "C" identifying serial ID as a chassis serial number Serial: The serial number as identified by the Sid field. <p>Example: "WS-C4510R@C@CAT234765XR"</p>	For long text message only
Message description	Short text describing the error.	CallHome/ MessageDescription
Device name	Node that experienced the event. it is the host name of the device.	CallHome/ CustomerData/ SystemInfo/ Name
Contact name	Name of person to contact for issues associated with the node experiencing the event.	CallHome/CustomerData/Syst emInfo/Contact
Contact e-mail	E-mail address of person identified as contact for this unit.	CallHome/CustomerData/Syst emInfo/ContactEmail
Contact phone number	Phone number of the person identified as the contact for this unit.	CallHome/CustomerData/Syst emInfo/ContactPhoneNumber
Street address	Optional field containing street address for RMA part shipments associated with this unit.	CallHome/CustomerData/Syst emInfo/StreetAddress
Model name	Model name of the unit (such as WS-C4503). it is the specific model as part of a product family name.	CallHome/ Device/ Cisco_Chassis/Model
Serial number	Chassis serial number of the unit.	CallHome/ Device/ Cisco_Chassis/ SerialNumber
Chassis part number	Top assembly number of the chassis as read from SEEPROM (such as WS-C4503 = 73-10558).	CallHome/Device/Cisco_Chas sis/AdditionalInformation/ AD@name="PartNumber"

Fields specific to a particular alert group message are inserted here.

The following fields may be repeated if multiple CLI commands are executed for this alert group.

Command output name	The exact command that was run (such as the show running-config command).	/aml/attachments/attachment/ name
---------------------	--	--------------------------------------

Table 79-5 Common Fields for All Long Text and XML Messages (continued)

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Attachment type	Specifically command output.	/aml/Attachments/attachment@type
MIME type	Normally text/plain or encoding type.	/aml/Attachments/Attachment/Data@encoding

Table 79-6 Inserted Fields for a Reactive or Proactive Event Message

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Chassis hardware version	Hardware version of chassis	CallHome/Device/Cisco_Chassis/HardwareVersion
Supervisor module software version	Top-level software version	CallHome/Device/Cisco_Chassis/AdditionalInformation/AD@name="SoftwareVersion"
Affected FRU name	Name of the affected FRU generating the event message	CallHome/Device/Cisco_Chassis/Cisco_Card/Model
Affected FRU serial number	Serial number of affected FRU	CallHome/Device/Cisco_Chassis/Cisco_Card/SerialNumber
Affected FRU part number	Part number of affected FRU	CallHome/Device/Cisco_Chassis/Cisco_Card/PartNumber
FRU slot	Slot number of FRU generating the event message	CallHome/Device/Cisco_Chassis/Cisco_Card/LocationWithinContainer
FRU hardware version	Hardware version of affected FRU	CallHome/Device/Cisco_Chassis/Cisco_Card/HardwareVersion
FRU software version	Software version(s) running on affected FRU	CallHome/Device/Cisco_Chassis/Cisco_Card/SoftwareIdentity/VersionString

Table 79-7 Inserted Fields for an Inventory Event Message

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Chassis hardware version	Hardware version of chassis	CallHome/Device/Cisco_Chassis/HardwareVersion
Supervisor module software version	Top-level software version	CallHome/Device/Cisco_Chassis/AdditionalInformation/AD@name="SoftwareVersion"
Affected FRU name	Name of the affected FRU generating the event message	CallHome/Device/Cisco_Chassis/Cisco_Card/Model

Table 79-7 *Inserted Fields for an Inventory Event Message (continued)*

Data Item (Plain Text and XML)	Description (Plain Text and XML)	XML Tag (XML Only)
Affected FRU s/n	Serial number of affected FRU	CallHome/Device/Cisco_Chassis/Cisco_Card/SerialNumber
Affected FRU part number	Part number of affected FRU	CallHome/Device/Cisco_Chassis/Cisco_Card/PartNumber
FRU slot	Slot number of FRU generating the event message	CallHome/Device/Cisco_Chassis/Cisco_Card/LocationWithinContainer
FRU hardware version	Hardware version of affected FRU	CallHome/Device/Cisco_Chassis/Cisco_Card/HardwareVersion
FRU software version	Software version(s) running on affected FRU	CallHome/Device/Cisco_Chassis/Cisco_Card/SoftwareIdentity/VersionString

Syslog Alert Notification in Long-Text Format Example

```

TimeStamp : 2009-02-06 12:57 GMT+00:00
Message Name : syslog
Message Type : Call Home
Message Group : reactive
Severity Level : 2
Source ID : Cat4500/4900
Device ID : WS-C4510R@C@1234567
Customer ID :
Contract ID :
Site ID :
Server ID : WS-C4510R@C@1234567
Event Description : *Feb 6 12:57:54.121: %CLEAR-5-COUNTERS: Clear counter on all
interfaces by console
System Name : Router
Contact Email : abc@example.com
Contact Phone :
Street Address :
Affected Chassis : WS-C4510R
Affected Chassis Serial Number : 1234567
Affected Chassis Part No : 12-3456-78
Affected Chassis Hardware Version : 1.1
Supervisor Software Version : 12.2(20090204:112419)
Command Output Name : show logging
Attachment Type : command output
MIME Type : text/plain
Command Output Text :
Syslog logging: enabled (0 messages dropped, 1 messages rate-limited, 0 flushes, 0
overruns, xml disabled, filtering disabled)

No Active Message Discriminator.

No Inactive Message Discriminator.
```

```

Console logging: level debugging, 95 messages logged, xml disabled,
                  filtering disabled
Monitor logging: level debugging, 0 messages logged, xml disabled,
                  filtering disabled
Buffer logging:   level debugging, 95 messages logged, xml disabled,
                  filtering disabled
Exception Logging: size (8192 bytes)
Count and timestamp logging messages: disabled
Persistent logging: disabled

```

No active filter modules.

```

Trap logging: level informational, 118 message lines logged

```

```

Log Buffer (4096 bytes):
00:59:54.379: %CALL_HOME-3-HTTP_REQUEST_FAILED: failed to send HTTP request to :
              https://172.17.46.17/its/service/oddce/services/DDCEService
              (ERR 107 : Bad parameters)
*Feb  6 00:59:55.379: %CALL_HOME-3-HTTP_REQUEST_FAILED: failed to send HTTP request to :
              https://172.17.46.17/its/service/oddce/services/DDCEService
              (ERR 107 : Bad parameters)
*Feb  6 01:04:37.903: %SYS-5-CONFIG_I: Configured from console by console
*Feb  6 01:04:51.783: %C4K_IOSMODPORTMAN-4-POWERSUPPLYREMOVED: Power supply 1 has been
removed
*Feb  6 01:04:56.047: %CALL_HOME-3-SMTP_SEND_FAILED: Unable to send notification using all
SMTP servers (ERR 6, error in reply from SMTP server)
*Feb  6 01:05:01.823: %C4K_IOSMODPORTMAN-6-POWERSUPPLYINSERTEDDETAILED: Power supply 1
(PWR-C45-1300ACV S/N: DTM123900VH Hw: 5.2) has been inserted
*Feb  6 01:05:01.823: %C4K_IOSMODPORTMAN-4-POWERSUPPLYBAD: Power supply 1 has failed or
been turned off
*Feb  6 01:05:01.823: %C4K_CHASSIS-3-MIXINPOWERDETECTED: Power supplies in the chassis are
of different types (AC/DC) or wattage
*Feb  6 01:05:51.827: %C4K_IOSMODPORTMAN-4-POWERSUPPLYREMOVED: Power supply 1 has been
removed
*Feb  6 01:05:56.087: %CALL_HOME-3-SMTP_SEND_FAILED: Unable to send notification using all
SMTP servers (ERR 6, error in reply from SMTP server)
*Feb  6 01:05:56.867: %C4K_IOSMODPORTMAN-6-POWERSUPPLYINSERTEDDETAILED: Power supply 1
(PWR-C45-1300ACV S/N: DTM123900VH Hw: 5.2) has been inserted
*Feb  6 01:05:56.867: %C4K_IOSMODPORTMAN-4-POWERSUPPLYBAD: Power supply 1 has failed or
been turned off
*Feb  6 01:05:56.867: %C4K_CHASSIS-3-MIXINPOWERDETECTED: Power supplies in the chassis are
of different types (AC/DC) or wattage
*Feb  6 01:06:31.871: %C4K_IOSMODPORTMAN-4-POWERSUPPLYREMOVED: Power supply 2 has been
removed
*Feb  6 01:06:31.871: %C4K_CHASSIS-3-INSUFFICIENTPOWERSUPPLIESDETECTED: Insufficient power
supplies present for specified configuration
*Feb  6 01:06:31.871: %C4K_CHASSIS-2-INSUFFICIENTPOWERDETECTED: Insufficient power
available for the current chassis configuration
*Feb  6 01:06:36.907: %C4K_IOSMODPORTMAN-6-POWERSUPPLYINSERTEDDETAILED: Power supply 2
(PWR-C45-1400AC S/N: AZS11260B3M Hw: 2.3) has been inserted
*Feb  6 01:08:06.911: %C4K_IOSMODPORTMAN-4-POWERSUPPLYREMOVED: Power supply 1 has been
removed
*Feb  6 01:08:11.171: %CALL_HOME-3-SMTP_SEND_FAILED: Unable to send notification using all
SMTP servers (ERR 6, error in reply from SMTP server)
*Feb  6 01:08:11.951: %C4K_IOSMODPORTMAN-6-POWERSUPPLYINSERTEDDETAILED: Power supply 1
(PWR-C45-1300ACV S/N: DTM123900VH Hw: 5.2) has been inserted
*Feb  6 01:08:11.951: %C4K_IOSMODPORTMAN-4-POWERSUPPLYBAD: Power supply 1 has failed or
been turned off
*Feb  6 01:08:11.951: %C4K_CHASSIS-3-MIXINPOWERDETECTED: Power supplies in the chassis are
of different types (AC/DC) or wattage
*Feb  6 01:10:35.371: %SYS-5-CONFIG_I: Configured from console by console
*Feb  6 01:12:06.955: %C4K_IOSMODPORTMAN-4-POWERSUPPLYREMOVED: Power supply 1 has been
removed

```



```
*Feb 6 01:12:11.995: %C4K_IOSMODPORTMAN-6-POWERSUPPLYINSERTEDDETAILED: Power supply 1
(PWR-C45-1300ACV S/N: DTM123900VH Hw: 5.2) has been inserted
*Feb 6 01:12:11.995: %C4K_IOSMODPORTMAN-4-POWERSUPPLYBAD: Power supply 1 has failed or
been turned off
*Feb 6 01:12:11.995: %C4K_CHASSIS-3-MIXINPOWERDETECTED: Power supplies in the chassis are
of different types (AC/DC) or wattage
*Feb 6 01:13:06.999: %C4K_IOSMODPORTMAN-4-POWERSUPPLYREMOVED: Power supply 2 has been
removed
*Feb 6 01:13:06.999: %C4K_CHASSIS-3-INSUFFICIENTPOWERSUPPLIESDETECTED: Insufficient power
supplies present for specified configuration
*Feb 6 01:13:06.999: %C4K_CHASSIS-2-INSUFFICIENTPOWERDETECTED: Insufficient power
available for the current chassis configuration
*Feb 6 01:13:12.035: %C4K_IOSMODPORTMAN-6-POWERSUPPLYINSERTEDDETAILED: Power supply 2
(PWR-C45-1400AC S/N: AZS11260B3M Hw: 2.3) has been inserted
*Feb 6 01:36:04.079: %SYS-5-CONFIG_I: Configured from console by console
*Feb 6 12:51:46.001: %SYS-5-CONFIG_I: Configured from console by console
*Feb 6 12:54:15.905: %SYS-5-CONFIG_I: Configured from console by console
Switch#
Command Output Name : show inventory
Attachment Type : command output
MIME Type : text/plain
Command Output Text : NAME: "Switch System", DESCR: "Cisco Systems, Inc. WS-C4510R 10 slot
switch "
PID: WS-C4510R          , VID: V06   , SN: 1234567

NAME: "Clock Module", DESCR: "Clock Module"
PID: WS-X4K-CLOCK       , VID: V04   , SN: 12345671

NAME: "Mux Buffer 3 ", DESCR: "Mux Buffers for Redundancy Logic"
PID: WS-X4590           , VID: V04   , SN: 12345672

NAME: "Mux Buffer 4 ", DESCR: "Mux Buffers for Redundancy Logic"
PID: WS-X4590           , VID: V04   , SN: 12345673

NAME: "Mux Buffer 5 ", DESCR: "Mux Buffers for Redundancy Logic"
PID: WS-X4590           , VID: V04   , SN: 12345674

NAME: "Mux Buffer 6 ", DESCR: "Mux Buffers for Redundancy Logic"
PID: WS-X4590           , VID: V04   , SN: 12345675

NAME: "Mux Buffer 7 ", DESCR: "Mux Buffers for Redundancy Logic"
PID: WS-X4590           , VID: V04   , SN: 12345676

NAME: "Mux Buffer 8 ", DESCR: "Mux Buffers for Redundancy Logic"
PID: WS-X4590           , VID: V04   , SN: 12345677

NAME: "Mux Buffer 9 ", DESCR: "Mux Buffers for Redundancy Logic"
PID: WS-X4590           , VID: V04   , SN: 12345678

NAME: "Mux Buffer 10 ", DESCR: "Mux Buffers for Redundancy Logic"
PID: WS-X4590           , VID: V04   , SN: 12345679

NAME: "Linecard(slot 2)", DESCR: "Supervisor V-10GE with 2 10GE X2 ports, and 4 1000BaseX
SFP ports"
PID: WS-X4516-10GE      , VID: V07   , SN: 1234567A

NAME: "Linecard(slot 3)", DESCR: "10/100/1000BaseT (RJ45)V with 48 10/100/1000 baseT voice
power ports (Cisco/IEEE)"
PID: WS-X4548-GB-RJ45V , VID: V08   , SN: 1234567B

NAME: "Linecard(slot 4)", DESCR: "10/100/1000BaseT (RJ45)V with 48 10/100/1000 baseT voice
power ports (Cisco/IEEE)"
PID: WS-X4548-GB-RJ45V , VID: V08   , SN: 1234567C
```

```

NAME: "Linecard(slot 5)", DESCR: "10/100BaseTX (RJ45) with 32 10/100 baseT and 4 100FX
daughtercard ports"
PID: WS-X4232-RJ-XX      , VID: V05      , SN: 1234567D

NAME: "Fan", DESCR: "FanTray"
PID: WS-X4582            , VID: V03      , SN: 1234567E

NAME: "Power Supply 1", DESCR: "Power Supply ( AC 1300W )"
PID: PWR-C45-1300ACV     , VID: V05      , SN: 1234567F

NAME: "Power Supply 2", DESCR: "Power Supply ( AC 1400W )"
PID: PWR-C45-1400AC      , VID: V04      , SN: 1234567G

```

Syslog Alert Notification in XML Format Example

```

<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope xmlns:soap-env="http://www.w3.org/2003/05/soap-envelope">
<soap-env:Header>
<aml-session:Session xmlns:aml-session="http://www.cisco.com/2004/01/aml-session"
soap-env:mustUnderstand="true"
soap-env:role="http://www.w3.org/2003/05/soap-envelope/role/next">
<aml-session:To>http://tools.cisco.com/neddce/services/DDCEService</aml-session:To>
<aml-session:Path>
<aml-session:Via>http://www.cisco.com/appliance/uri</aml-session:Via>
</aml-session:Path>
<aml-session:From>http://www.cisco.com/appliance/uri</aml-session:From>
<aml-session:MessageId>M44:1234567:abcd</aml-session:MessageId>
</aml-session:Session>
</soap-env:Header>
<soap-env:Body>
<aml-block:Block xmlns:aml-block="http://www.cisco.com/2004/01/aml-block">
<aml-block:Header>
<aml-block:Type>http://www.cisco.com/2005/05/callhome/syslog</aml-block:Type>
<aml-block:CreationDate>2009-02-06 12:58:31 GMT+00:00</aml-block:CreationDate>
<aml-block:Builder>
<aml-block:Name>Cat4500/4900</aml-block:Name>
<aml-block:Version>2.0</aml-block:Version>
</aml-block:Builder>
<aml-block:BlockGroup>
<aml-block:GroupId>G45:1234567:abcd</aml-block:GroupId>
<aml-block:Number>0</aml-block:Number>
<aml-block:IsLast>true</aml-block:IsLast>
<aml-block:IsPrimary>true</aml-block:IsPrimary>
<aml-block:WaitForPrimary>false</aml-block:WaitForPrimary>
</aml-block:BlockGroup>
<aml-block:Severity>2</aml-block:Severity>
</aml-block:Header>
<aml-block:Content>
<ch:CallHome xmlns:ch="http://www.cisco.com/2005/05/callhome" version="1.0">
<ch:EventTime>2009-02-06 12:58:30 GMT+00:00</ch:EventTime>
<ch:MessageDescription>*Feb 6 12:58:30.293: %CLEAR-5-COUNTERS: Clear counter on all
interfaces by console</ch:MessageDescription>
<ch:Event>
<ch:Type>syslog</ch:Type>
<ch:SubType></ch:SubType>
<ch:Brand>Cisco Systems</ch:Brand>
<ch:Series>Cat4k Series Switches</ch:Series>
</ch:Event>
<ch:CustomerData>
<ch:UserData>
<ch:Email>abc@example.com</ch:Email>
</ch:UserData>

```

```

<ch:ContractData>
<ch:CustomerId></ch:CustomerId>
<ch:SiteId></ch:SiteId>
<ch:ContractId></ch:ContractId>
<ch:DeviceId>WS-C4510R@C@1234567</ch:DeviceId>
</ch:ContractData>
<ch:SystemInfo>
<ch:Name>Switch</ch:Name>
<ch:Contact></ch:Contact>
<ch:ContactEmail>abc@example.com</ch:ContactEmail>
<ch:ContactPhoneNumber></ch:ContactPhoneNumber>
<ch:StreetAddress></ch:StreetAddress>
</ch:SystemInfo>
<ch:CCOID></ch:CCOID>
</ch:CustomerData>
<ch:Device>
<rme:Chassis xmlns:rme="http://www.cisco.com/rme/4.0">
<rme:Model>WS-C4510R</rme:Model>
<rme:HardwareVersion>1.1</rme:HardwareVersion>
<rme:SerialNumber>1234567</rme:SerialNumber>
<rme:AdditionalInformation>
<rme:AD name="PartNumber" value="12-3456-05" />
<rme:AD name="SoftwareVersion" value="12.2(20090204:112419)" />
<rme:AD name="SystemObjectId" value="1.2.3.4.5.6.7.537" />
<rme:AD name="SystemDescription" value="Cisco IOS Software, Catalyst 4500 L3 Switch
Software (cat4500-ENTSERVICES-M), Experimental Version 12.2(20090204:112419) Copyright (c)
1986-2009 by Cisco Systems, Inc.
Compiled Fri 06-Feb-09 15:22 by abc" />
</rme:AdditionalInformation>
</rme:Chassis>
</ch:Device>
</ch:CallHome>
</aml-block:Content>
<aml-block:Attachments>
<aml-block:Attachment type="inline">
<aml-block:Name>show logging</aml-block:Name>
<aml-block:Data encoding="plain">
<![CDATA[
Syslog logging: enabled (0 messages dropped, 1 messages rate-limited, 0 flushes, 0
overruns, xml disabled, filtering disabled)

```

No Active Message Discriminator.

No Inactive Message Discriminator.

```

Console logging: level debugging, 97 messages logged, xml disabled,
filtering disabled
Monitor logging: level debugging, 0 messages logged, xml disabled,
filtering disabled
Buffer logging: level debugging, 97 messages logged, xml disabled,
filtering disabled
Exception Logging: size (8192 bytes)
Count and timestamp logging messages: disabled
Persistent logging: disabled

```

No active filter modules.

```

Trap logging: level informational, 120 message lines logged

```

```

Log Buffer (4096 bytes):
107 : Bad parameters)

```

```

*Feb  6 00:59:55.379: %CALL_HOME-3-HTTP_REQUEST_FAILED: failed to send HTTP request to :
    https://172.17.46.17/its/service/odce/services/DDCEService
    (ERR 107 : Bad parameters)
*Feb  6 01:04:37.903: %SYS-5-CONFIG_I: Configured from console by console
*Feb  6 01:04:51.783: %C4K_IOSMODPORTMAN-4-POWERSUPPLYREMOVED: Power supply 1 has been
removed
*Feb  6 01:04:56.047: %CALL_HOME-3-SMTP_SEND_FAILED: Unable to send notification using all
SMTP servers (ERR 6, error in reply from SMTP server)
*Feb  6 01:05:01.823: %C4K_IOSMODPORTMAN-6-POWERSUPPLYINSERTEDDETAILED: Power supply 1
(PWR-C45-1300ACV S/N: DTM123900VH Hw: 5.2) has been inserted
*Feb  6 01:05:01.823: %C4K_IOSMODPORTMAN-4-POWERSUPPLYBAD: Power supply 1 has failed or
been turned off
*Feb  6 01:05:01.823: %C4K_CHASSIS-3-MIXINPOWERDETECTED: Power supplies in the chassis are
of different types (AC/DC) or wattage
*Feb  6 01:05:51.827: %C4K_IOSMODPORTMAN-4-POWERSUPPLYREMOVED: Power supply 1 has been
removed
*Feb  6 01:05:56.087: %CALL_HOME-3-SMTP_SEND_FAILED: Unable to send notification using all
SMTP servers (ERR 6, error in reply from SMTP server)
*Feb  6 01:05:56.867: %C4K_IOSMODPORTMAN-6-POWERSUPPLYINSERTEDDETAILED: Power supply 1
(PWR-C45-1300ACV S/N: DTM123900VH Hw: 5.2) has been inserted
*Feb  6 01:05:56.867: %C4K_IOSMODPORTMAN-4-POWERSUPPLYBAD: Power supply 1 has failed or
been turned off
*Feb  6 01:05:56.867: %C4K_CHASSIS-3-MIXINPOWERDETECTED: Power supplies in the chassis are
of different types (AC/DC) or wattage
*Feb  6 01:06:31.871: %C4K_IOSMODPORTMAN-4-POWERSUPPLYREMOVED: Power supply 2 has been
removed
*Feb  6 01:06:31.871: %C4K_CHASSIS-3-INSUFFICIENTPOWERSUPPLIESDETECTED: Insufficient power
supplies present for specified configuration
*Feb  6 01:06:31.871: %C4K_CHASSIS-2-INSUFFICIENTPOWERDETECTED: Insufficient power
available for the current chassis configuration
*Feb  6 01:06:36.907: %C4K_IOSMODPORTMAN-6-POWERSUPPLYINSERTEDDETAILED: Power supply 2
(PWR-C45-1400AC S/N: AZS11260B3M Hw: 2.3) has been inserted
*Feb  6 01:08:06.911: %C4K_IOSMODPORTMAN-4-POWERSUPPLYREMOVED: Power supply 1 has been
removed
*Feb  6 01:08:11.171: %CALL_HOME-3-SMTP_SEND_FAILED: Unable to send notification using all
SMTP servers (ERR 6, error in reply from SMTP server)
*Feb  6 01:08:11.951: %C4K_IOSMODPORTMAN-6-POWERSUPPLYINSERTEDDETAILED: Power supply 1
(PWR-C45-1300ACV S/N: DTM123900VH Hw: 5.2) has been inserted
*Feb  6 01:08:11.951: %C4K_IOSMODPORTMAN-4-POWERSUPPLYBAD: Power supply 1 has failed or
been turned off
*Feb  6 01:08:11.951: %C4K_CHASSIS-3-MIXINPOWERDETECTED: Power supplies in the chassis are
of different types (AC/DC) or wattage
*Feb  6 01:10:35.371: %SYS-5-CONFIG_I: Configured from console by console
*Feb  6 01:12:06.955: %C4K_IOSMODPORTMAN-4-POWERSUPPLYREMOVED: Power supply 1 has been
removed
*Feb  6 01:12:11.995: %C4K_IOSMODPORTMAN-6-POWERSUPPLYINSERTEDDETAILED: Power supply 1
(PWR-C45-1300ACV S/N: DTM123900VH Hw: 5.2) has been inserted
*Feb  6 01:12:11.995: %C4K_IOSMODPORTMAN-4-POWERSUPPLYBAD: Power supply 1 has failed or
been turned off
*Feb  6 01:12:11.995: %C4K_CHASSIS-3-MIXINPOWERDETECTED: Power supplies in the chassis are
of different types (AC/DC) or wattage
*Feb  6 01:13:06.999: %C4K_IOSMODPORTMAN-4-POWERSUPPLYREMOVED: Power supply 2 has been
removed
*Feb  6 01:13:06.999: %C4K_CHASSIS-3-INSUFFICIENTPOWERSUPPLIESDETECTED: Insufficient power
supplies present for specified configuration
*Feb  6 01:13:06.999: %C4K_CHASSIS-2-INSUFFICIENTPOWERDETECTED: Insufficient power
available for the current chassis configuration
*Feb  6 01:13:12.035: %C4K_IOSMODPORTMAN-6-POWERSUPPLYINSERTEDDETAILED: Power supply 2
(PWR-C45-1400AC S/N: AZS11260B3M Hw: 2.3) has been inserted
*Feb  6 01:36:04.079: %SYS-5-CONFIG_I: Configured from console by console
*Feb  6 12:51:46.001: %SYS-5-CONFIG_I: Configured from console by console
*Feb  6 12:54:15.905: %SYS-5-CONFIG_I: Configured from console by console
*Feb  6 12:57:54.121: %CLEAR-5-COUNTERS: Clear counter on all interfaces by console
*Feb  6 12:58:24.093: %SYS-5-CONFIG_I: Configured from console by console

```

```

Switch#]]></aml-block:Data>
</aml-block:Attachment>
<aml-block:Attachment type="inline">
<aml-block:Name>show inventory</aml-block:Name>
<aml-block:Data encoding="plain">
<![CDATA[NAME: "Switch System", DESCR: "Cisco Systems, Inc. WS-C4510R 10 slot switch "
PID: WS-C4510R          , VID: V06  , SN: 1234567

NAME: "Clock Module", DESCR: "Clock Module"
PID: WS-X4K-CLOCK      , VID: V04  , SN: 12345671

NAME: "Mux Buffer 3 ", DESCR: "Mux Buffers for Redundancy Logic"
PID: WS-X4590          , VID: V04  , SN: 12345672

NAME: "Mux Buffer 4 ", DESCR: "Mux Buffers for Redundancy Logic"
PID: WS-X4590          , VID: V04  , SN: 12345673

NAME: "Linecard(slot 2)", DESCR: "Supervisor V-10GE with 2 10GE X2 ports, and 4 1000BaseX
SFP ports"
PID: WS-X4516-10GE     , VID: V07  , SN: 12345674

NAME: "Linecard(slot 3)", DESCR: "10/100/1000BaseT (RJ45)V with 48 10/100/1000 baseT voice
power ports (Cisco/IEEE)"
PID: WS-X4548-GB-RJ45V , VID: V08  , SN: 12345675

NAME: "Linecard(slot 4)", DESCR: "10/100/1000BaseT (RJ45)V with 48 10/100/1000 baseT voice
power ports (Cisco/IEEE)"
PID: WS-X4548-GB-RJ45V , VID: V08  , SN: 12345676

NAME: "Linecard(slot 5)", DESCR: "10/100BaseTX (RJ45) with 32 10/100 baseT and 4 100FX
daughtercard ports"
PID: WS-X4232-RJ-XX    , VID: V05  , SN: 12345677

NAME: "Fan", DESCR: "FanTray"
PID: WS-X4582          , VID: V03  , SN: 12345678

NAME: "Power Supply 1", DESCR: "Power Supply ( AC 1300W )"
PID: PWR-C45-1300ACV   , VID: V05  , SN: 12345679

NAME: "Power Supply 2", DESCR: "Power Supply ( AC 1400W )"
PID: PWR-C45-1400AC    , VID: V04  , SN: 1234567A

Switch#]]></aml-block:Data>
</aml-block:Attachment>
</aml-block:Attachments>
</aml-block:Block>
</soap-env:Body>
</soap-env:Envelope>

```




Configuring Cisco IOS IP SLA Operations

This chapter describes how to use Cisco IOS IP Service Level Agreements (SLAs) on Catalyst 4500 series switch. Cisco IP SLAs is a part of Cisco IOS software that allows Cisco customers to analyze IP service levels for IP applications and services by using active traffic monitoring—the generation of traffic in a continuous, reliable, and predictable manner—for measuring network performance. With Cisco IOS IP SLAs, service provider customers can measure and provide service level agreements, and enterprise customers can verify service levels, verify outsourced service level agreements, and understand network performance. Cisco IOS IP SLAs can perform network assessments, verify quality of service (QoS), ease the deployment of new services, and assist with network troubleshooting.

The Catalyst 4500 series switch Base or Enterprise Services The switch also supports the Built-in Traffic Simulator using Cisco IOS IP SLAs video operations to generate synthetic traffic for a variety of video applications, such as Telepresence, IPTV and IP video surveillance camera. You can use the simulator tool:

- for network assessment before deploying applications that have stringent network performance requirements.
- along with the Cisco Mediatrace for post-deployment troubleshooting for any network related performance issues.

The traffic simulator includes a sophisticated scheduler that allows the user to run several tests simultaneously or periodically and over extended time periods (Supported only on switches running the Enterprise Services feature set).

This chapter consists of these sections:

- [Understanding Cisco IOS IP SLAs, page 80-2](#)
- [Configuring IP SLAs Operations, page 80-6](#)
- [Monitoring IP SLAs Operations, page 80-12](#)



Note

For information on configuring this feature, see the *Configuring Cisco IOS IP SLAs Video Operations* document at:

http://www.cisco.com/en/US/docs/ios-xml/ios/ipsla/configuration/12-2se/sla_video.html

For more information about IP SLAs, see the *Cisco IOS IP SLAs Configuration Guides*:

<http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipsla/configuration/xe-3e/sla-xe-3e-book.html>

For command syntax information, see the command reference:

http://www.cisco.com/en/US/docs/ios/ipsla/command/reference/sla_book.html

Understanding Cisco IOS IP SLAs

Cisco IOS IP SLAs sends data across the network to measure performance between multiple network locations or across multiple network paths. It simulates network data and IP services and collects network performance information in real time. Cisco IOS IP SLAs generates and analyzes traffic either between Cisco IOS devices or from a Cisco IOS device to a remote IP device such as a network application server. Measurements provided by the various Cisco IOS IP SLAs operations can be used for troubleshooting, for problem analysis, and for designing network topologies.

Depending on the specific Cisco IOS IP SLAs operation, various network performance statistics are monitored within the Cisco device and stored in both command-line interface (CLI) and Simple Network Management Protocol (SNMP) MIBs. IP SLAs packets have configurable IP and application layer options such as source and destination IP address, User Datagram Protocol (UDP)/TCP port numbers, a type of service (ToS) byte (including Differentiated Services Code Point [DSCP] and IP Prefix bits), Virtual Private Network (VPN) routing/forwarding instance (VRF), and URL web address.

Because Cisco IP SLAs is Layer 2 transport independent, you can configure end-to-end operations over disparate networks to best reflect the metrics that an end user is likely to experience. IP SLAs collects a unique subset of these performance metrics:

- Delay (both round-trip and one-way)
- Jitter (directional)
- Packet loss (directional)
- Packet sequencing (packet ordering)
- Path (per hop)
- Connectivity (directional)
- Server or website download time

Because Cisco IOS IP SLAs is SNMP-accessible, it can also be used by performance-monitoring applications like CiscoWorks Internetwork Performance Monitor (IPM) and other third-party Cisco partner performance management products. You can find more details about network management products that use Cisco IOS IP SLAs:

<http://www.cisco.com/go/ipsla>

Using IP SLAs can provide these benefits:

- Service-level agreement monitoring, measurement, and verification.
- Network performance monitoring
 - Measures the jitter, latency, or packet loss in the network.
 - Provides continuous, reliable, and predictable measurements.
- IP service network health assessment to verify that the existing QoS is sufficient for new IP services.
- Edge-to-edge network availability monitoring for proactive verification and connectivity testing of network resources (for example, shows the network availability of an NFS server used to store business critical data from a remote site).
- Troubleshooting of network operation by providing consistent, reliable measurement that immediately identifies problems and saves troubleshooting time.
- Multiprotocol Label Switching (MPLS) performance monitoring and network verification (if the switch supports MPLS)

This section includes this information about IP SLAs functionality:

- [Using Cisco IOS IP SLAs to Measure Network Performance, page 80-3](#)
- [IP SLAs Responder and IP SLAs Control Protocol, page 80-4](#)
- [Response Time Computation for IP SLAs, page 80-4](#)
- [IP SLAs Operation Scheduling, page 80-5](#)
- [IP SLAs Operation Threshold Monitoring, page 80-5](#)

Using Cisco IOS IP SLAs to Measure Network Performance

You can use IP SLAs to monitor the performance between any area in the network—core, distribution, and edge—without deploying a physical probe. It uses generated traffic to measure network performance between two networking devices. [Figure 80-1](#) shows how IP SLAs begins when the source device sends a generated packet to the destination device. After the destination device receives the packet, depending on the type of IP SLAs operation, it responds with time-stamp information for the source to make the calculation on performance metrics. An IP SLAs operation performs a network measurement from the source device to a destination in the network using a specific protocol such as UDP.

Figure 80-1 *Cisco IOS IP SLAs Operation*

To implement IP SLAs network performance measurement, you need to perform these tasks:

1. Enable the IP SLAs responder, if required.
2. Configure the required IP SLAs operation type.
3. Configure any options available for the specified operation type.
4. Configure threshold conditions, if required.
5. Schedule the operation to run, then let the operation run for a period of time to gather statistics.
6. Display and interpret the results of the operation using the Cisco IOS CLI or a network management system (NMS) system with SNMP.

For more information about IP SLAs operations, see the operation-specific chapters in the *Cisco IOS IP SLAs Configuration Guide*:

http://www.cisco.com/en/US/products/ps6441/products_installation_and_configuration_guides_list.html

The switch does not support Voice over IP (VoIP) service levels using the gatekeeper registration delay operations measurements. Before configuring any IP SLAs application, you can use the **show ip sla application** privileged EXEC command to verify that the operation type is supported on your software image.

IP SLAs Responder and IP SLAs Control Protocol

The IP SLAs responder is a component embedded in the destination Cisco device that allows the system to anticipate and respond to IP SLAs request packets. The responder provides accurate measurements without the need for dedicated probes. The responder uses the Cisco IOS IP SLAs Control Protocol to provide a mechanism through which it can be notified on which port it should listen and respond. Only a Cisco IOS device can be a source for a destination IP SLAs Responder.



Note

The IP SLAs responder can be a Cisco IOS Layer 2, responder-configurable switch, such as a Catalyst 2960 switch. The responder does not need to support full IP SLAs functionality.

Figure 80-1 shows where the Cisco IOS IP SLAs responder fits in the IP network. The responder listens on a specific port for control protocol messages sent by an IP SLAs operation. Upon receipt of the control message, it enables the specified UDP or TCP port for the specified duration. During this time, the responder accepts the requests and responds to them. It disables the port after it responds to the IP SLAs packet, or when the specified time expires. MD5 authentication for control messages is available for added security.

You do not need to enable the responder on the destination device for all IP SLAs operations. For example, a responder is not required for services that are already provided by the destination router (such as Telnet or HTTP). You cannot configure the IP SLAs responder on non-Cisco devices and Cisco IOS IP SLAs can send operational packets only to services native to those devices.

Response Time Computation for IP SLAs

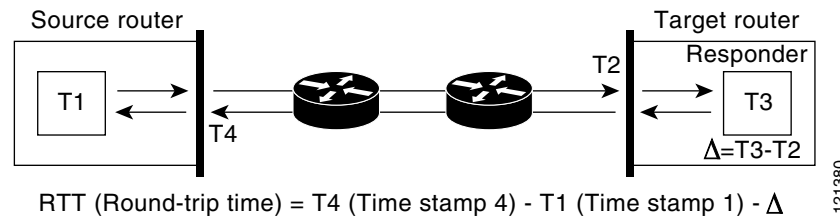
Switches and routers can take tens of milliseconds to process incoming packets due to other high priority processes. This delay affects the response times because the test-packet reply might be in a queue while waiting to be processed. In this situation, the response times would not accurately represent true network delays. IP SLAs minimizes these processing delays on the source device as well as on the target device (if the responder is being used) to determine true round-trip times. IP SLAs test packets use time stamping to minimize the processing delays.

When the IP SLAs responder is enabled, it allows the target device to take time stamps when the packet arrives on the interface at interrupt level and again just as it is leaving, eliminating the processing time. This time stamping is made with a granularity of sub-milliseconds (ms).

Figure 80-2 demonstrates how the responder works. Four time stamps are taken to make the calculation for round-trip time. At the target router, with the responder functionality enabled, time stamp 2 (TS2) is subtracted from time stamp 3 (TS3) to produce the time spent processing the test packet as represented

by delta. This delta value is then subtracted from the overall round-trip time. Notice that the same principle is applied by IP SLAs on the source router where the incoming time stamp 4 (TS4) is also taken at the interrupt level to allow for greater accuracy.

Figure 80-2 Cisco IOS IP SLAs Responder Time Stamping



An additional benefit of the two time stamps at the target device is the ability to track one-way delay, jitter, and directional packet loss. Because much network behavior is asynchronous, it is critical to have these statistics. However, to capture one-way delay measurements, you must configure both the source router and target router with Network Time Protocol (NTP) so that the source and target are synchronized to the same clock source. One-way jitter measurements do not require clock synchronization.

IP SLAs Operation Scheduling

When you configure an IP SLAs operation, you must schedule the operation to begin capturing statistics and collecting error information. You can schedule an operation to start immediately or to start at a certain month, day, and hour. You can use the pending option to set the operation to start at a later time. The pending option is an internal state of the operation that is visible through SNMP. The pending state is also used when an operation is a reaction (threshold) operation waiting to be triggered. You can schedule a single IP SLAs operation or a group of operations at one time.

You can schedule several IP SLAs operations by using a single command through the Cisco IOS CLI or the CISCO RTTMON-MIB. Scheduling the operations to run at evenly distributed times allows you to control the amount of IP SLAs monitoring traffic. This distribution of IP SLAs operations helps minimize the CPU utilization and thus improves network scalability.

For more details about the IP SLAs multioperations scheduling functionality, see the “IP SLAs—Multiple Operation Scheduling” chapter of the *Cisco IOS IP SLAs Configuration Guide*:

http://www.cisco.com/en/US/products/ps6441/products_installation_and_configuration_guides_list.html

IP SLAs Operation Threshold Monitoring

To support successful service level agreement monitoring, you must have mechanisms that notify you immediately of any possible violation. IP SLAs can send SNMP traps that are triggered by events such as these:

- Connection loss
- Timeout
- Round-trip time threshold
- Average jitter threshold

- One-way packet loss
- One-way jitter
- One-way mean opinion score (MOS)
- One-way latency

An IP SLAs threshold violation can also trigger another IP SLAs operation for further analysis. For example, the frequency could be increased or an ICMP path echo or ICMP path jitter operation could be initiated for troubleshooting.

Determining the type of threshold and the level to set can be complex, and depends on the type of IP service being used in the network.

http://www.cisco.com/en/US/products/ps6441/products_installation_and_configuration_guides_list.html

Configuring IP SLAs Operations

This section does not include configuration information for all available operations as the configuration information details are included in the *Cisco IOS IP SLAs Configuration Guide*. It does include several operations as examples, including configuring the responder, configuring UDP jitter operation, which requires a responder, and configuring ICMP echo operation, which does not require a responder.

For details about configuring other operations, see the *Cisco IOS IP SLAs Configuration Guide*:

http://www.cisco.com/en/US/products/ps6441/products_installation_and_configuration_guides_list.html

This section includes this information:

- [IP SLA Default Configuration, page 80-6](#)
- [IP SLA Configuration Guidelines, page 80-6](#)
- [Configuring the IP SLAs Responder, page 80-7](#)
- [Analyzing IP Service Levels by Using the UDP Jitter Operation, page 80-8](#)
- [Analyzing IP Service Levels by Using the ICMP Echo Operation, page 80-11](#)

IP SLA Default Configuration

By default, no IP SLAs operations are configured.

IP SLA Configuration Guidelines

For information on the IP SLAs commands, see the *Cisco IOS IP SLAs Command Reference, Release 12.4T* command reference:

http://www.cisco.com/en/US/docs/ios/ipsla/command/reference/sla_book.html

For detailed descriptions and configuration procedures, see the *Cisco IOS IP SLAs Configuration Guide, Release 12.4T* L:

http://www.cisco.com/en/US/products/ps6441/products_installation_and_configuration_guides_list.html

Note that not all of the IP SLAs commands or operations described in this guide are supported on the switch. The switch supports IP service level analysis by using UDP jitter, UDP echo, HTTP, TCP connect, ICMP echo, ICMP path echo, ICMP path jitter, FTP, DNS, and DHCP, as well as multiple operation scheduling and proactive threshold monitoring. It does not support VoIP service levels using the gatekeeper registration delay operations measurements.

Before configuring any IP SLAs application, you can use the **show ip sla application** privileged EXEC command to verify that the operation type is supported on your software image. This is an example of the output from the command:

```
Switch# show ip sla application
      IP SLAs
Version: 2.2.0 Round Trip Time MIB, Infrastructure Engine-II
Time of last change in whole IP SLAs: 22:17:39.117 UTC Fri Jun
Estimated system max number of entries: 15801

Estimated number of configurable operations: 15801
Number of Entries configured      : 0
Number of active Entries         : 0
Number of pending Entries        : 0
Number of inactive Entries       : 0

      Supported Operation Types
Type of Operation to Perform: 802.1agEcho
Type of Operation to Perform: 802.1agJitter
Type of Operation to Perform: dhcp
Type of Operation to Perform: dns
Type of Operation to Perform: echo
Type of Operation to Perform: ftp
Type of Operation to Perform: http
Type of Operation to Perform: jitter
Type of Operation to Perform: pathEcho
Type of Operation to Perform: pathJitter
Type of Operation to Perform: tcpConnect
Type of Operation to Perform: udpEcho

IP SLAs low memory water mark: 21741224
```

Configuring the IP SLAs Responder

The IP SLAs responder is available only on Cisco IOS software-based devices, including some Layer 2 switches that do not support full IP SLAs functionality, such as a Catalyst 2960 switch.

To configure the IP SLAs responder on the target device (the operational target), perform this task:

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.
Step 2	<code>ip sla responder {tcp-connect udp-echo} ipaddress ip-address port port-number</code>	Configures the switch as an IP SLAs responder. The keywords have these meanings: <ul style="list-style-type: none"> • tcp-connect—Enable the responder for TCP connect operations. • udp-echo—Enable the responder for User Datagram Protocol (UDP) echo or jitter operations. • ipaddress ip-address—Enter the destination IP address. • port port-number—Enter the destination port number. <p>Note The IP address and port number must match those configured on the source device for the IP SLAs operation.</p>
Step 3	<code>end</code>	Returns to privileged EXEC mode.
Step 4	<code>show ip sla responder</code>	Verifies the IP SLAs responder configuration on the device.
Step 5	<code>copy running-config startup-config</code>	(Optional) Saves your entries in the configuration file.

To disable the IP SLAs responder, enter the **no ip sla responder** global configuration command. This example shows how to configure the device as a responder for the UDP jitter IP SLAs operation in the next procedure:

```
Switch(config)# ip sla responder udp-echo 172.29.139.134 5000
```

Analyzing IP Service Levels by Using the UDP Jitter Operation

Jitter means interpacket delay variance. When multiple packets are sent consecutively 10 ms apart from source to destination, if the network is behaving correctly, the destination should receive them 10 ms apart. But if there are delays in the network (like queuing, arriving through alternate routes, and so on) the arrival delay between packets might be more than or less than 10 ms with a positive jitter value meaning that the packets arrived more than 10 ms apart. If the packets arrive 12 ms apart, positive jitter is 2 ms; if the packets arrive 8 ms apart, negative jitter is 2 ms. For delay-sensitive networks, positive jitter values are undesirable, and a jitter value of 0 is ideal.

In addition to monitoring jitter, the IP SLAs UDP jitter operation can be used as a multipurpose data gathering operation. The packets IP SLAs generates carry packet sending and receiving sequence information and sending and receiving time stamps from the source and the operational target. Based on these, UDP jitter operations measure this data:

- Per-direction jitter (source to destination and destination to source)
- Per-direction packet-loss
- Per-direction delay (one-way delay)
- Round-trip delay (average round-trip time)

To configure UDP jitter operation on the source device, perform this task:

	Command	Purpose
Step 1	configure terminal	Enters global configuration mode.
Step 2	ip sla operation-number	Creates an IP SLAs operation, and enter IP SLAs configuration mode.
Step 3	udp-jitter { <i>destination-ip-address</i> <i>destination-hostname</i> } <i>destination-port</i> [source-ip { <i>ip-address</i> <i>hostname</i> }] [source-port <i>port-number</i>] [control { enable disable }] [num-packets <i>number-of-packets</i>] [interval <i>interpacket-interval</i>]	Configures the IP SLAs operation as a UDP jitter operation, and enter UDP jitter configuration mode. <ul style="list-style-type: none"> • <i>destination-ip-address</i> <i>destination-hostname</i>—Specify the destination IP address or hostname. • <i>destination-port</i>—Specify the destination port number in the range from 1 to 65535. • (Optional) source-ip {<i>ip-address</i> <i>hostname</i>}—Specify the source IP address or hostname. When a source IP address or hostname is not specified, IP SLAs chooses the IP address nearest to the destination • (Optional) source-port <i>port-number</i>—Specify the source port number in the range from 1 to 65535. When a port number is not specified, IP SLAs chooses an available port. • (Optional) control—Enable or disable sending of IP SLAs control messages to the IP SLAs responder. By default, IP SLAs control messages are sent to the destination device to establish a connection with the IP SLAs responder • (Optional) num-packets <i>number-of-packets</i>—Enter the number of packets to be generated. The range is 1 to 6000; the default is 10. • (Optional) interval <i>inter-packet-interval</i>—Enter the interval between sending packets in milliseconds. The range is 1 to 6000; the default value is 20 ms.
Step 4	frequency <i>seconds</i>	(Optional) Sets the rate at which a specified IP SLAs operation repeats. The range is from 1 to 604800 seconds; the default is 60 seconds.
Step 5	exit	Exits UDP jitter configuration mode, and return to global configuration mode.

Step 4	frequency <i>seconds</i>	(Optional) Sets the rate at which a specified IP SLAs operation repeats. The range is from 1 to 604800 seconds; the default is 60 seconds.
Step 5	exit	Exits UDP jitter configuration mode, and return to global configuration mode.

To disable the IP SLAs operation, enter the no **ip sla operation-number** global configuration command. This example shows how to configure a UDP jitter IP SLAs operation:

```
Switch(config)# ip sla 10
Switch(config-ip-sla)# udp-jitter 172.29.139.134 5000
Switch(config-ip-sla-jitter)# frequency 30
Switch(config-ip-sla-jitter)# exit
Switch(config)# ip sla schedule 5 start-time now life forever
Switch(config)# end
Switch# show ip sla configuration 10
IP SLAs, Infrastructure Engine-II.
```

```
Entry number: 10
Owner:
Tag:
Type of operation to perform: udp-jitter
Target address/Source address: 10.1.1.1/0.0.0.0
Target port/Source port: 2/0
Request size (ARR data portion): 32
Operation timeout (milliseconds): 5000
Packet Interval (milliseconds)/Number of packets: 20/10
```



```

Type Of Service parameters: 0x0
Verify data: No
Vrf Name:
Control Packets: enabled
Schedule:
    Operation frequency (seconds): 30
    Next Scheduled Start Time: Pending trigger
    Group Scheduled : FALSE
    Randomly Scheduled : FALSE
    Life (seconds): 3600
    Entry Ageout (seconds): never
    Recurring (Starting Everyday): FALSE
    Status of entry (SNMP RowStatus): notInService
Threshold (milliseconds): 5000
Distribution Statistics:
    Number of statistic hours kept: 2
    Number of statistic distribution buckets kept: 1
    Statistic distribution interval (milliseconds): 20
Enhanced History:

```

Analyzing IP Service Levels by Using the ICMP Echo Operation

The ICMP echo operation measures end-to-end response time between a Cisco device and any devices using IP. Response time is computed by measuring the time taken between sending an ICMP echo request message to the destination and receiving an ICMP echo reply. Many customers use IP SLAs ICMP-based operations, in-house ping testing, or ping-based dedicated probes for response time measurements between the source IP SLAs device and the destination IP device. The IP SLAs ICMP echo operation conforms to the same specifications as ICMP ping testing, and the two methods result in the same response times.



Note

This operation does not require the IP SLAs responder to be enabled.

To disable the IP SLAs operation, enter the **no ip sla operation-number** global configuration command. This example shows how to configure an ICMP echo IP SLAs operation:

```

Switch(config)# ip sla 12
Switch(config-ip-sla)# icmp-echo 172.29.139.134
Switch(config-ip-sla-echo)# frequency 30
Switch(config-ip-sla-echo)# exit
Switch(config)# ip sla schedule 5 start-time now life forever
Switch(config)# end
Switch# show ip sla configuration 22
IP SLAs, Infrastructure Engine-II.

```

```

Entry number: 12
Owner:
Tag:
Type of operation to perform: echo
Target address: 2.2.2.2
Source address: 0.0.0.0
Request size (ARR data portion): 28
Operation timeout (milliseconds): 5000
Type Of Service parameters: 0x0
Verify data: No
Vrf Name:
Schedule:
    Operation frequency (seconds): 60
    Next Scheduled Start Time: Pending trigger

```

```
Group Scheduled : FALSE
Randomly Scheduled : FALSE
Life (seconds): 3600
Entry Ageout (seconds): never
Recurring (Starting Everyday): FALSE
Status of entry (SNMP RowStatus): notInService
Threshold (milliseconds): 5000
Distribution Statistics:
  Number of statistic hours kept: 2
  Number of statistic distribution buckets kept: 1
  Statistic distribution interval (milliseconds): 20
History Statistics:
  Number of history Lives kept: 0
  Number of history Buckets kept: 15
  History Filter Type: None
Enhanced History:
```

Monitoring IP SLAs Operations

To display IP SLAs operations configuration

Command	Purpose
<code>show ip sla authentication</code>	Displays IP SLAs authentication information.
<code>show ip sla responder</code>	Displays information about the IP SLAs responder.



Configuring RMON

This chapter describes how to configure Remote Network Monitoring (RMON) on your Catalyst 4500 series switch. RMON is a standard monitoring specification that defines a set of statistics and functions that can be exchanged between RMON-compliant console systems and network probes. RMON provides you with comprehensive network-fault diagnosis, planning, and performance-tuning information.

This chapter consists of these sections:

- [About RMON, page 81-1](#)
- [Configuring RMON, page 81-2](#)
- [Displaying RMON Status, page 81-5](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

About RMON

RMON is an Internet Engineering Task Force (IETF) standard monitoring specification that allows various network agents and console systems to exchange network monitoring data. You can use the RMON feature with the Simple Network Management Protocol (SNMP) agent in the switch to monitor all the traffic flowing among switches on all connected LAN segments.

Figure 81-1 Remote Monitoring Example

The switch supports these RMON groups (defined in RFC 1757):

- Statistics (RMON group 1)—Collects Ethernet, Fast Ethernet, and Gigabit Ethernet statistics on an interface.
- History (RMON group 2)—Collects a history group of statistics on Ethernet, Fast Ethernet, and Gigabit Ethernet interfaces for a specified polling interval.
- Alarm (RMON group 3)—Monitors a specific MIB object for a specified interval, triggers an alarm at a specified value (rising threshold), and resets the alarm at another value (falling threshold). Alarms can be used with events; the alarm triggers an event, which can generate a log entry or an SNMP trap.
- Event (RMON group 9)—Determines the action to take when an event is triggered by an alarm. The action can be to generate a log entry or an SNMP trap.

Because switches supported by Cisco IOS Release 12.2(31)SG use hardware counters for RMON data processing, the monitoring is more efficient, and little processing power is required.

Configuring RMON

This section describes how to configure RMON on your switch. It contains this configuration information:

- [Default RMON Configuration, page 81-2](#)
- [Configuring RMON Alarms and Events, page 81-2](#)
- [Configuring RMON Collection on an Interface, page 81-4](#)

Default RMON Configuration

RMON is disabled by default; no alarms or events are configured.

Only RMON 1 is supported on the switch.

Configuring RMON Alarms and Events

You can configure your switch for RMON by using the command-line interface (CLI) or an SNMP-compatible network management station. We recommend that you use a generic RMON console application on the network management station (NMS) to take advantage of RMON's network management capabilities. You must also configure SNMP on the switch to access RMON MIB objects. For more information, see [Chapter 75, "Configuring SNMP."](#)

To enable RMON alarms and events, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# rmon alarm <i>number variable interval</i> { absolute delta } rising-threshold <i>value</i> [<i>event-number</i>] falling-threshold <i>value</i> [<i>event-number</i>] [owner <i>string</i>]	Sets an alarm on a MIB object. <ul style="list-style-type: none"> For <i>number</i>, specify the alarm number. The range is 1 to 65535. For <i>variable</i>, specify the MIB object to monitor. For <i>interval</i>, specify the time in seconds the alarm monitors the MIB variable. The range is 1 to 4294967295 seconds. Specify the absolute keyword to test each MIB variable directly; specify the delta keyword to test the change between samples of a MIB variable. For <i>value</i>, specify a number at which the alarm is triggered and one for when the alarm is reset. The range for the rising threshold and falling threshold <i>values</i> is -2147483648 to 2147483647. (Optional) For <i>event-number</i>, specify the event number to trigger when the rising or falling threshold exceeds its limit. (Optional) For owner <i>string</i>, specify the owner of the alarm.
Step 3	Switch(config)# rmon event <i>number</i> [description <i>string</i>] [log] [owner <i>string</i>] [trap <i>community</i>]	Adds an event in the RMON event table that is associated with an RMON event number. <ul style="list-style-type: none"> For <i>number</i>, assign an event number. The range is 1 to 65535. (Optional) For description <i>string</i>, specify a description of the event. (Optional) Use the log keyword to generate an RMON log entry when the event is triggered. (Optional) For owner <i>string</i>, specify the owner of this event. (Optional) For <i>community</i>, enter the SNMP community string used for this trap.
Step 4	Switch(config)# end	Returns to privileged EXEC mode.
Step 5	Switch# show running-config	Verifies your entries.
Step 6	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To disable an alarm, use the **no rmon alarm number** global configuration command on each alarm you configured. You cannot disable at once all the alarms that you configured. To disable an event, use the **no rmon event number** global configuration command. To learn more about alarms and events and how they interact with each other, see RFC 1757.

You can set an alarm on any MIB object. The following example configures RMON alarm number 10 by using the **rmon alarm** command. The alarm monitors the MIB variable ifEntry.20.1 once every 20 seconds until the alarm is disabled and checks the change in the variable's rise or fall. If the ifEntry.20.1 value shows a MIB counter increase of 15 or more, such as from 100000 to 100015, the alarm is triggered. The alarm in turn triggers event number 1, which is configured with the **rmon event** command. Possible events can include a log entry or an SNMP trap. If the ifEntry.20.1 value changes by 0, the alarm is reset and can be triggered again.

```
Switch(config)# rmon alarm 10 ifEntry.20.1 20 delta rising-threshold 15 1
falling-threshold 0 owner jjohnson
```

The following example creates RMON event number 1 by using the **rmon event** command. The event is defined as High ifOutErrors and generates a log entry when the event is triggered by the alarm. The user jjones owns the row that is created in the event table by this command. This example also generates an SNMP trap when the event is triggered.

```
Switch(config)# rmon event 1 log trap eventtrap description "High ifOutErrors" owner
jjones
```

Configuring RMON Collection on an Interface

You must first configure RMON alarms and events to display collection information.

To collect group history statistics on an interface, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>interface-id</i>	Specifies the interface on which to collect history, and enter interface configuration mode.
Step 3	Switch(config-if)# rmon collection history <i>index</i> [buckets <i>bucket-number</i>] [interval <i>seconds</i>] [owner <i>ownername</i>]	Enables history collection for the specified number of buckets and time period. <ul style="list-style-type: none"> For <i>index</i>, identify the RMON group of statistics. The range is 1 to 65535. (Optional) For buckets <i>bucket-number</i>, specify the maximum number of buckets desired for the RMON collection history group of statistics. The range is 1 to 65535. The default is 50 buckets. (Optional) For interval <i>seconds</i>, specify the number of seconds in each polling cycle. (Optional) For owner <i>ownername</i>, enter the name of the owner of the RMON group of statistics. To disable history collection, use the no rmon collection history index interface configuration command.
Step 4	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 5	Switch# show running-config	Verifies your entries.

	Command	Purpose
Step 6	Switch# show rmon history	Displays the contents of the switch history table.
Step 7	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

To collect group Ethernet statistics on an interface, perform this task:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface interface-id	Specifies the interface on which to collect statistics, and enter interface configuration mode.
Step 3	Switch(config-if)# rmon collection stats index [owner ownername]	<p>Enables RMON statistic collection on the interface.</p> <ul style="list-style-type: none"> For <i>index</i>, specify the RMON group of statistics. The range is from 1 to 65535. (Optional) For owner ownername, enter the name of the owner of the RMON group of statistics. <p>To disable the collection of group Ethernet statistics, use the no rmon collection stats index interface configuration command.</p>
Step 4	Switch(config-if)# end	Returns to privileged EXEC mode.
Step 5	Switch# show running-config	Verifies your entries.
Step 6	Switch# show rmon statistics	Displays the contents of the switch statistics table.
Step 7	Switch# copy running-config startup-config	(Optional) Saves your entries in the configuration file.

Displaying RMON Status

To display the RMON status, use one or more of the following privileged EXEC commands:

Command	Purpose
show rmon	Displays general RMON statistics.
show rmon alarms	Displays the RMON alarm table.
show rmon events	Displays the RMON event table.
show rmon history	Displays the RMON history table.
show rmon statistics	Displays the RMON statistics table.



Performing Diagnostics

You can use diagnostics to test and verify the functionality of the hardware components of your system (chassis, supervisor engines, modules, and ASICs) while your Catalyst 4500 series switch is connected to a live network. Diagnostics consists of packet-switching tests that test hardware components and verify the data path and control signals.

Online diagnostics are categorized as bootup, on-demand, schedule, or health-monitoring diagnostics. Bootup diagnostics run during bootup; on-demand diagnostics run from the CLI; scheduled diagnostics run at user-designated intervals or specified times when the switch is connected to a live network; and health-monitoring runs in the background.



Note

Diagnostic shell mode is not supported on Supervisor Engine 9-E, 8-E, and 7-E, in wired mode.

This chapter consists of these sections:

- [Configuring Online Diagnostics, page 82-1](#)
- [Performing Diagnostics, page 82-3](#)
- [Power-On Self-Test Diagnostics, page 82-10](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

Configuring Online Diagnostics

These sections describe how to configure online diagnostics:

- [Configuring On-Demand Online Diagnostics, page 82-2](#)
- [Scheduling Online Diagnostics, page 82-2](#)

Configuring On-Demand Online Diagnostics

You can run on-demand online diagnostic tests from the CLI. You can set the execution action to either stop or continue the test when a failure is detected, or to stop the test after a specific number of failures occur with the failure count setting. The iteration setting allows you to configure a test to run multiple times.

To schedule online diagnostics, perform this task:

Command	Purpose
Switch# diagnostic ondemand {iteration iteration_count} {action-on-error {continue stop} [error_count]}	Configures on-demand diagnostic tests to run, how many times to run (iterations), and what action to take when errors are found.

This example shows how to set the on-demand testing iteration count:

```
Switch# diagnostic ondemand iterations 3
Switch#
```

This example shows how to set the execution action when an error is detected:

```
Switch# diagnostic ondemand action-on-error continue 2
Switch#
```

Scheduling Online Diagnostics

You can schedule online diagnostics to run at a designated time of day or on a daily, weekly, or monthly basis. You can schedule tests to run only once or to repeat at an interval. Use the **no** form of this command to remove the scheduling.

To configure online diagnostics, perform this task:

Command	Purpose
Switch(config)# diagnostic schedule module number test {test_id test_id_range all} [port {num num_range all} {on mm dd yyyy hh:mm} {daily hh:mm} {weekly day_of_week hh:mm}]	Schedules on-demand diagnostic tests on the specified module for a specific date and time, how many times to run (iterations), and what action to take when errors are found.

This example shows how to schedule diagnostic testing on a specific date and time for a specific port on module 6:

```
Switch(config)# diagnostic schedule module 6 test 2 port 3 on may 23 2009 23:32
Switch(config)#
```

This example shows how to schedule diagnostic testing to occur daily:

```
Switch(config)# diagnostic schedule module 6 test 2 port 3 daily 12:34
Switch(config)#
```

This example shows how to schedule diagnostic testing to occur weekly:

```
Switch(config)# diagnostic schedule module 6 test 2 port 3 weekly friday 09:23
Switch(config)#
```

Performing Diagnostics

After you configure online diagnostics, you can start or stop diagnostic tests or display the test results. You can also see which tests are configured and what diagnostic tests have already run.

These sections describe how to run online diagnostic tests after they have been configured:

- [Starting and Stopping Online Diagnostic Tests, page 82-3](#)
- [Displaying Online Diagnostic Tests and Test Results, page 82-4](#)
- [Displaying Data Path Online Diagnostics Test Results, page 82-7](#)
- [Line Card Online Diagnostics, page 82-8](#)
- [Troubleshooting with Online Diagnostics, page 82-8](#)



Note

Before you enable any online diagnostics tests, enable the logging console or monitor to observe all warning messages.



Note

When running disruptive tests, only run them when you are connected using the console. When disruptive tests complete, a warning message on the console will recommend that you reload the system to return to normal operation. Strictly follow this warning.

Starting and Stopping Online Diagnostic Tests

After you configure diagnostic tests, you can use the **start** and **stop** keywords to begin or end a test.

To start or stop an online diagnostic command, perform one of these tasks:

Command	Purpose
Switch# diagnostic start module <i>number test {test_id </i> <i>test_id_range minimal complete</i> <i> basic per-port </i> <i>non-disruptive all} [port {num </i> <i>port#_range all}]</i>	Starts a diagnostic test on a port or range of ports on the specified module.
Switch# diagnostic stop module <i>number</i>	Stops a diagnostic test on the specified module.

This example shows how to start a diagnostic test on module 6:

```
Switch# diagnostic start module 6 test 2
Diagnostic[module 6]: Running test(s) 2 Run interface level cable diags
Diagnostic[module 6]: Running test(s) 2 may disrupt normal system operation
Do you want to continue? [no]: yes
Switch#
*May 14 21:11:46.631: %DIAG-6-TEST_RUNNING: module 6: Running online-diag-tdr{ID=2} ...
*May 14 21:11:46.631: %DIAG-6-TEST_OK: module 6: online-diag-tdr{ID=2} has completed
successfully
Switch#
```

This example shows how to stop a diagnostic test on module 6:

```
Switch# diagnostic stop module 6
Diagnostic[module 6]: Diagnostic is not active.
```

The message indicates no active diagnostic on module 6

Displaying Online Diagnostic Tests and Test Results

You can display the configured online diagnostic tests and check the results of the tests with the **show diagnostic** command.

To display the configured diagnostic tests, perform this task:

Command	Purpose
Switch# show diagnostic {bootup cns content [module num] description [module num] events [module num] [event-type event-type] ondemand result [module num] [detail] schedule [module num] simulation status}	Displays the test results of online diagnostics and lists supported test suites.

This example shows how to display the online diagnostics configured on module 1:

```
Switch# show diagnostic content module 6
module 6:
Diagnostics test suite attributes:
  M/C/* - Minimal bootup level test / Complete bootup level test / NA
  B/* - Basic ondemand test / NA
  P/V/* - Per port test / Per device test / NA
  D/N/* - Disruptive test / Non-disruptive test / NA
  S/* - Only applicable to standby unit / NA
  X/* - Not a health monitoring test / NA
  F/* - Fixed monitoring interval test / NA
  E/* - Always enabled monitoring test / NA
  A/I - Monitoring is active / Monitoring is inactive
  cable-tdr/* - Interface cable diags / NA
  o/* - Ongoing test, always active / NA

ID   Test Name                               Attributes      Test Interval  Thre-
====  =====                               =====
1) linecard-online-diag -----> M**D***I**      not configured  n/a
2) online-diag-tdr -----> **PD***Icable- not configured  n/a
3) stub-rx-errors -----> ***N***A**      000 00:01:00.00 n/a
4) supervisor-rx-errors -----> ***N***A**      000 00:01:00.00 n/a
```

This example shows how to display the test description for a given test on a module:

```
Switch# show diagnostic description module 6 test 1

linecard-online-diag :
  Linecard online-diagnostics run after the system boots up but
  before it starts passing traffic. Each linecard port is placed in
  loopback, and a few packets are injected into the switching fabric
  from the cpu to the port. If the packets are successfully
```

received by the cpu, the port passes the test. Sometimes one port or a group of ports sharing common components fail. The linecard is then placed in partial faulty mode. If no ports can loop back traffic, the board is placed in faulty state.

Switch#

This example shows how to display the online diagnostic results for module 6:

Switch# **show diagnostic result module 6**

Current bootup diagnostic level: minimal

module 6: SerialNo : JAB0815059L

Overall Diagnostic Result for module 6 : PASS

Diagnostic level at card bootup: minimal

Test results: (. = Pass, F = Fail, U = Untested)

- 1) linecard-online-diag -----> .
- 2) online-diag-tdr:

Port	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
	U	U	.	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U

- 3) stub-rx-errors -----> .
- 4) supervisor-rx-errors -----> .

Switch#

This example shows how to display the online diagnostic results details for module 6:

Switch# **show diagnostic result module 6 detail**

Current bootup diagnostic level: minimal

module 6: SerialNo : JAB0815059L

Overall Diagnostic Result for module 6 : PASS

Diagnostic level at card bootup: minimal

Test results: (. = Pass, F = Fail, U = Untested)

- 1) linecard-online-diag -----> .

```

Error code -----> 0 (DIAG_SUCCESS)
Total run count -----> 1
Last test testing type -----> n/a
Last test execution time ----> Jun 01 2009 11:19:36
First test failure time -----> n/a
Last test failure time -----> n/a
Last test pass time -----> Jun 01 2009 11:19:36
Total failure count -----> 0
Consecutive failure count ---> 0

```

Slot	Ports	Card Type	Diag Status	Diag Details
6	24	10/100/1000BaseT (RJ45)V, Cisco/IEEE	Passed	None

Ports	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Ports	17	18	19	20	21	22	23	24								

```
Error code -----> 0 (DIAG_SUCCESS)
Total run count -----> 1
Last test testing type -----> OnDemand
Last test execution time -----> Jun 03 2009 05:39:00
First test failure time -----> n/a
Last test failure time -----> n/a
Last test pass time -----> Jun 03 2009 05:39:00
Total failure count -----> 0
Consecutive failure count ----> 0
```

```
Error code -----> 3 (DIAG_SUCCESS)
Total run count -----> 4
Last test testing type -----> Health Monitoring
Last test execution time -----> Dec 20 2009 22:30:41
First test failure time -----> n/a
Last test failure time -----> n/a
Last test pass time -----> Dec 20 2009 22:30:41
Total failure count -----> 0
Consecutive failure count -----> 0
```

```
Error code -----> 3 (DIAG_SUCCESS)
Total run count -----> 4
Last test testing type -----> Health Monitoring
Last test execution time -----> Dec 20 2009 22:30:41
First test failure time -----> n/a
```

```

Last test failure time -----> n/a
Last test pass time -----> Dec 20 2009 22:30:41
Total failure count -----> 0
Consecutive failure count ---> 0

```

Switch#

Displaying Data Path Online Diagnostics Test Results

A data path online diagnostic test verifies that the data paths between the supervisor engine and the linecards (defined as a number of stub ASICs) are functioning correctly. There is a direct connection between each stub ASIC on a line card and the supervisor engine. Error counters on the supervisor engine (supervisor-rx-trends) and each stub ASIC on a line card (stub-rx-trends) are monitored periodically. Error counters that continually increase indicate malfunctioning hardware in the data path and cause the test to fail. Data path online diagnostic tests are non-destructive and the error counters are polled every minute.

Errors on the stub end of the data path are reported as errors in traffic egressing to the line card from the supervisor engine switching ASICs. Some initial errors might be revealed as links are brought up, but they should not increase. An increasing count indicates a poor connection between the supervisor engine and a line card. If only one line card is affected, the cause is likely an incorrectly seated or faulty line card. The error counts include idle frames, so detection can occur when traffic is not flowing.

Errors on the supervisor end of the data path are reported as errors in traffic ingressing to the supervisor engine from linecards. The error counts should not increase and the detection includes idle frames. If the error counts increase for more than one line card, the likely cause is a faulty supervisor engine or chassis. If only one stub or line card is affected, the likely cause is a faulty line card or a defective mux buffer (for a redundant chassis).

In addition to running periodically, data path online diagnostics can be also be invoked on-demand in the following way:

```

Switch# diagnostic start module 1 test stub-rx-errors
*Apr 1 09:25:14.211: %DIAG-6-TEST_RUNNING: module 1: Running stub-rx-errors{ID=3} ...
*Apr 1 09:25:14.211: %DIAG-6-TEST_OK: module 1: stub-rx-errors{ID=3} has completed
Switch# diagnostic start module 1 test supervisor-rx-errors
*Apr 1 09:25:26.503: %DIAG-6-TEST_RUNNING: module 1: Running supervisor-rx-errors{ID=4}
...
*Apr 1 09:25:26.503: %DIAG-6-TEST_OK: module 1: supervisor-rx-errors{ID=4} has completed
successfully

```

Detailed information about the test results can be viewed as follows:

```
Switch# show diagnostic result module 1 test stub-rx-errors detail
```

Current bootup diagnostic level: minimal

Test results: (. = Pass, F = Fail, U = Untested)

```

3) stub-rx-errors -----> .

Error code -----> 0 (DIAG_SUCCESS)
Total run count -----> 7
Last test testing type -----> OnDemand
Last test execution time ----> Apr 01 2010 09:25:14
First test failure time -----> n/a
Last test failure time -----> n/a

```

```

Last test pass time -----> Apr 01 2010 09:25:14
Total failure count -----> 0
Consecutive failure count ---> 0

```

```
Switch# show diagnostic result module 1 test supervisor-rx-errors detail
```

```
Current bootup diagnostic level: minimal
```

```
Test results: (. = Pass, F = Fail, U = Untested)
```

```

4) supervisor-rx-errors -----> .

Error code -----> 0 (DIAG_SUCCESS)
Total run count -----> 4
Last test testing type -----> OnDemand
Last test execution time ----> Apr 01 2010 09:25:26
First test failure time ----> n/a
Last test failure time -----> n/a
Last test pass time -----> Apr 01 2010 09:25:26
Total failure count -----> 0
Consecutive failure count ---> 0

```

```
Switch#
```

Line Card Online Diagnostics

A line card online diagnostic test verifies that all ports on a line card are working correctly. The test can detect whether the path to the front panel port on the line card is broken. The test cannot indicate where along the path that the problem occurred.



Note

This test is run only for line cards that have stub chips.

Line card online diagnostics runs only once, when the line cards boot. This situation can happen when you insert a line card or power up a chassis.

Line card online diagnostics are performed by sending a packet from the CPU to every port on the line card. Because this packet is marked loopback, the CPU expects to see this packet return from the port. The packet first traverses the ASICs on the supervisor engine card, then travels by using the chassis backplane and the stub chip on the line cards to the PHYs. The PHY sends it back down the same path.



Note

The packet does not reach or exit the front panel port.

Troubleshooting with Online Diagnostics

A faulty line card occurs if any of the following conditions occurs.

- All ports fail
- All ports on a stub chip fail
- Only one port fails

For all of these situations, the output of the **show module** command would display the status of the line card as faulty:

```
Switch# show mod
Chassis Type : WS-C4507R
Power consumed by backplane : 40 Watts
```

Mod	Ports	Card Type	Model	Serial No.
1	6	Sup II+10GE 10GE (X2), 1000BaseX (SFP)	WS-X4013+10GE	JAB091502G0
2	6	Sup II+10GE 10GE (X2), 1000BaseX (SFP)	WS-X4013+10GE	JAB091502FC
3	48	100BaseX (SFP)	WS-X4248-FE-SFP	JAB093305RP
4	48	10/100BaseTX (RJ45)V	WS-X4148-RJ45V	JAE070717E5
5	48	10/100BaseTX (RJ45)V	WS-X4148-RJ45V	JAE061303U3
6	48	10/100BaseTX (RJ45)V	WS-X4148-RJ45V	JAE061303WJ
7	24	10/100/1000BaseT (RJ45)V, Cisco/IEEE	WS-X4524-GB-RJ45V	JAB0815059Q

M	MAC addresses	Hw	Fw	Sw	Status
1	000b.5f27.8b80 to 000b.5f27.8b85	0.2	12.2(27r)SG(12.2(37)SG	Ok
2	000b.5f27.8b86 to 000b.5f27.8b8b	0.2	12.2(27r)SG(12.2(37)SG	Ok
3	0005.9a80.6810 to 0005.9a80.683f	0.4			Ok
4	000c.3016.aae0 to 000c.3016.ab0f	2.6			Ok
5	0008.a3a3.4e70 to 0008.a3a3.4e9f	1.6			Ok
6	0008.a3a3.3fa0 to 0008.a3a3.3fcf	1.6			Faulty
7	0030.850e.3e78 to 0030.850e.3e8f	1.0			Ok

Mod	Redundancy role	Operating mode	Redundancy status
1	Active Supervisor	SSO	Active
2	Standby Supervisor	SSO	Standby hot

To troubleshoot a faulty line card, follow these steps:

Step 1 Enter the command **show diagnostic result module 3**.

If a faulty line card was inserted in the chassis, it will fail the diagnostics and the output will be similar to the following:

```
Current bootup diagnostic level: minimal

module 3:   SerialNo : JAB093305RP

Overall Diagnostic Result for module 3 : MAJOR ERROR
Diagnostic level at card bootup: minimal

Test results: (. = Pass, F = Fail, U = Untested)

1) linecard-online-diag -----> F
```

Switch#

Issue an RMA for the line card, contact TAC, and skip steps 2 and 3.

The output may display the following:

```
module 3:

Overall diagnostic result: PASS

Test results: (. = Pass, F = Fail, U = Untested)

1) linecard-online-diag -----> .
```

The message indicates that the line card passed online diagnostics either when it was inserted into the chassis the last time or when the switch was powered up (as reported by the “.”). You need to obtain additional information to determine the cause.

Step 2 Insert a different supervisor engine card and reinsert the line card.

If the line card passes the test, it suggests that the supervisor engine card is defective.

Issue an RMA for the supervisor engine, contact TAC, and skip step 3.

Because online diagnostics are not run on the supervisor engine card, you cannot use the **#show diagnostic module 1** command to test whether the supervisor engine card is faulty.

Step 3 Reinsert the line card in a different chassis.

If the line card passes the test, the problem is associated with the chassis.

Issue an RMA for the chassis and contact TAC.

Power-On Self-Test Diagnostics

The following topics are discussed:

- [Overview of Power-On Self-Test Diagnostics, page 82-10](#)
- [POST Result Example, page 82-11](#)
- [Power-On Self-Test Results, page 82-12](#)
- [Troubleshooting the Test Failures, page 82-18](#)

Overview of Power-On Self-Test Diagnostics

All Catalyst 4500 series switches have power-on self-test (POST) diagnostics that run whenever a supervisor engine boots. POST tests the basic hardware functionality for the supervisor switching engine, its associated packet memory and other on-board hardware components. The results of the POST impacts how the switch boots, because the health of the supervisor engine is critical to the operation of the switch. The switch might boot in a marginal or faulty state.

POST is currently supported on the following supervisor engines:

- WS-X4014
- WS-X4515
- WS-X4516
- WS-X4516-10GE
- WS-X4013+
- WS-X4013+TS
- WS-X4013+10GE
- WS-X45-SUP6L-E
- WS-X45-SUP7-E
- WS-X45-SUP7L-E
- WS-X45-SUP8-E

- WS-X45-SUP8L-E
- WS-X45-SUP9-E

The POST results are indicated with a period (.) or a Pass for Pass, an F for a Fail and a U for Untested.

POST Result Example

For all the supervisor engines, POST performs CPU, traffic, system, system memory, and feature tests.

For CPU tests, POST verifies appropriate activity of the supervisor engine SEEPROM, temperature sensor, and Ethernet end-of-band channel (EOBC), when used.

The following example illustrates the output of a CPU subsystem test:

```
[...]
Cpu Subsystem Tests ...
seeprom: . temperature_sensor: . eobc: .
[...]
```

For traffic tests, the POST sends packets from the CPU to the switch. These packets loop several times within the switch core and validate the switching, the Layer 2 and the Layer 3 functionality. To isolate the hardware failures accurately, the loop back is done both inside and outside the switch ports.

Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, and 7-EChecking digital signature
/nfs/gsg-sw/interim/flo_gsbu8/newest_image/iosxe/prod/cat4500e-universal.bin: Digitally
Signed Development Software with key version A

```
Rommon reg: 0x00004FA8
Reset2Reg: 0x00000F00
```

```
Image load status: 0x00000000
#####
Snowtrooper 220 controller 0x04324CF8..0x044EDFA6 Size:0x0058B0C1 Program Done!
#####
Linux version 2.6.24.4.3.3.k10 (priypras@gsg-lnx-bld6) (gcc version 4.2.1 p4 (Cisco
c4.2.1-p4)) #1 SMP Mon Jul 18 02:35:13 PDT 2011
Starting System Services
```

```
diagsk10-post version 4.1.7.4
```

```
prod: WS-X45-SUP7-E part: 73-12064-08 serial: CAT1418L05H
```

```
Power-on-self-test for Module 1: WS-X45-SUP7-E
Test Status: (. = Pass, F = Fail, U = Untested)
```

```
CPU Subsystem Tests ...
seeprom: Pass
```

```
Traffic: L3 Loopback ...
Test Results: Pass
```

```
Traffic: L2 Loopback ...
Test Results: Pass
post done
Exiting to ios...
```

Power-On Self-Test Results

The following topics are discussed:

- [Sample Display of the POST on an Active Supervisor Engine, page 82-12](#)
- [Sample Display of the POST on a Standby Supervisor Engine, page 82-15](#)

Sample Display of the POST on an Active Supervisor Engine

Examples include:

- [Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, and 7-E, page 82-12](#)
- [Supervisor Engine 6-E and Supervisor Engine 6L-E, page 82-13](#)

Supervisor Engine 9-E, 8L-E, 8-E, 7-LE, and 7-E

```
Switch# show diagnostic result module 3 detail

Current bootup diagnostic level: minimal

module 3:   SerialNo : CAT1450L1QU

Overall Diagnostic Result for module 3 : PASS
Diagnostic level at card bootup: minimal

Test results: (. = Pass, F = Fail, U = Untested)

-----

1) supervisor-bootup -----> .

      Error code -----> 0 (DIAG_SUCCESS)
      Total run count -----> 1
      Last test testing type -----> n/a
      Last test execution time ----> Jul 21 2011 20:16:56
      First test failure time -----> n/a
      Last test failure time -----> n/a
      Last test pass time -----> Jul 21 2011 20:16:56
      Total failure count -----> 0
      Consecutive failure count ---> 0

Power-On-Self-Test Results for ACTIVE Supervisor

prod: WS-X45-SUP7-E part: 73-12064-08 serial: CAT1450L1QU

Power-on-self-test for Module 3: WS-X45-SUP7-E
Test Status: (. = Pass, F = Fail, U = Untested)

CPU Subsystem Tests ...
  seeprom: Pass

Traffic: L3 Loopback ...
  Test Results: Pass

Traffic: L2 Loopback ...
  Test Results: Pass

Module 3 Passed
```

```
2) linecard-online-diag -----> .
```

```
Error code -----> 0 (DIAG_SUCCESS)
Total run count -----> 1
Last test testing type -----> n/a
Last test execution time ----> Jul 21 2011 20:16:56
First test failure time ----> n/a
Last test failure time -----> n/a
Last test pass time -----> Jul 21 2011 20:16:56
Total failure count -----> 0
Consecutive failure count ---> 0
```

Slot	Ports	Card Type	Diag Status	Diag Details
3	4	Sup 7-E 10GE (SFP+), 1000BaseX (SFP)	Skipped	Packet memory

Detailed Status

```
-----
. = Pass                U = Unknown
L = Loopback failure    S = Stub failure
P = Port failure
E = EEPROM failure      G = GBIC integrity check failure
```

```
Ports 1  2  3  4
      .  .  .  .
```

```
3) stub-rx-errors -----> .
```

```
Error code -----> 0 (DIAG_SUCCESS)
Total run count -----> 2
Last test testing type -----> Health Monitoring
Last test execution time ----> Jul 21 2011 20:18:57
First test failure time ----> n/a
Last test failure time -----> n/a
Last test pass time -----> Jul 21 2011 20:18:57
Total failure count -----> 0
Consecutive failure count ---> 0
```

```
4) supervisor-rx-errors -----> .
```

```
Error code -----> 0 (DIAG_SUCCESS)
Total run count -----> 2
Last test testing type -----> Health Monitoring
Last test execution time ----> Jul 21 2011 20:18:57
First test failure time ----> n/a
Last test failure time -----> n/a
Last test pass time -----> Jul 21 2011 20:18:57
Total failure count -----> 0
Consecutive failure count ---> 0
```

Switch#

Supervisor Engine 6-E and Supervisor Engine 6L-E

Switch# **show diagnostic result module 5 detail**

Current bootup diagnostic level: minimal

module 5: SerialNo : JAE1213CK36

Overall Diagnostic Result for module 5 : PASS
Diagnostic level at card bootup: minimal

Test results: (. = Pass, F = Fail, U = Untested)

1) supervisor-bootup -----> .

Error code -----> 0 (DIAG_SUCCESS)
Total run count -----> 1
Last test testing type -----> n/a
Last test execution time ----> Jul 21 2011 13:35:55
First test failure time -----> n/a
Last test failure time -----> n/a
Last test pass time -----> Jul 21 2011 13:35:55
Total failure count -----> 0
Consecutive failure count ---> 0

Power-On-Self-Test Results for ACTIVE Supervisor

prod: WS-X45-SUP6-E part: 73-10597-06 serial: JAE1213CK36

Power-on-self-test for Module 5: WS-X45-SUP6-E

CPU Subsystem Tests ...

seeprom: Pass

Traffic: L3 Looopback ...

Test Results: Pass

Traffic: L2 Loopback ...

Test Results: Pass

Switching Subsystem Memory ...

Packet Memory Test Results: Pass

Module 5 Passed

Remote TenGigabitPort status: Untested

2) linecard-online-diag -----> .

Error code -----> 0 (DIAG_SUCCESS)
Total run count -----> 1
Last test testing type -----> n/a
Last test execution time ----> Jul 21 2011 13:35:55
First test failure time -----> n/a
Last test failure time -----> n/a
Last test pass time -----> Jul 21 2011 13:35:55
Total failure count -----> 0
Consecutive failure count ---> 0

Slot	Ports	Card Type	Diag Status	Diag Details
5	6	Sup 6-E 10GE (X2), 1000BaseX (SFP)	Skipped	Packet memory

Detailed Status

```
. = Pass          U = Unknown
L = Loopback failure  S = Stub failure
P = Port failure
E = SEEPROM failure   G = GBIC integrity check failure
```

```
Ports 1  2  3  4  5  6
      .  .  .  .  .  .
```

```
3) stub-rx-errors -----> .
```

```
Error code -----> 0 (DIAG_SUCCESS)
Total run count -----> 1
Last test testing type -----> Health Monitoring
Last test execution time ----> Jul 21 2011 13:36:57
First test failure time ----> n/a
Last test failure time -----> n/a
Last test pass time -----> Jul 21 2011 13:36:57
Total failure count -----> 0
Consecutive failure count ----> 0
```

```
4) supervisor-rx-errors -----> .
```

```
Error code -----> 0 (DIAG_SUCCESS)
Total run count -----> 1
Last test testing type -----> Health Monitoring
Last test execution time ----> Jul 21 2011 13:36:57
First test failure time ----> n/a
Last test failure time -----> n/a
Last test pass time -----> Jul 21 2011 13:36:57
Total failure count -----> 0
Consecutive failure count ----> 0
```

```
Switch#
```

Sample Display of the POST on a Standby Supervisor Engine

Example include:

- [Supervisor Engine 7-E and Supervisor Engine 7L-E, page 82-15](#)
- [Supervisor Engine 6-E and Supervisor Engine 6L-E, page 82-17](#)

Supervisor Engine 7-E and Supervisor Engine 7L-E

```
Switch# show diagnostic result module 4 detail
```

```
Current bootup diagnostic level: minimal
```

```
module 4: SerialNo :
```

```
Overall Diagnostic Result for module 4 : PASS
Diagnostic level at card bootup: minimal
```

```
Test results: (. = Pass, F = Fail, U = Untested)
```

```
1) supervisor-bootup -----> .
```

```
Error code -----> 0 (DIAG_SUCCESS)
```

```

Total run count -----> 1
Last test testing type -----> n/a
Last test execution time ----> Jul 21 2011 20:16:56
First test failure time -----> n/a
Last test failure time -----> n/a
Last test pass time -----> Jul 21 2011 20:16:56
Total failure count -----> 0
Consecutive failure count ---> 0

```

Power-On-Self-Test Results for STANDBY Supervisor

Power-On-Self-Test utility did not run during last boot session

```

2) linecard-online-diag -----> .

```

```

Error code -----> 0 (DIAG_SUCCESS)
Total run count -----> 1
Last test testing type -----> n/a
Last test execution time ----> Jul 21 2011 20:16:56
First test failure time -----> n/a
Last test failure time -----> n/a
Last test pass time -----> Jul 21 2011 20:16:56
Total failure count -----> 0
Consecutive failure count ---> 0

```

Slot	Ports	Card Type	Diag Status	Diag Details
4	4	Sup 7-E 10GE (SFP+), 1000BaseX (SFP)	Passed	None

Detailed Status

```

-----
. = Pass          U = Unknown
L = Loopback failure  S = Stub failure
P = Port failure
E = SEEPROM failure  G = GBIC integrity check failure

```

```

Ports 1   2   3   4
      .   .   .   .

```

```

3) stub-rx-errors -----> .

```

```

Error code -----> 0 (DIAG_SUCCESS)
Total run count -----> 1
Last test testing type -----> Health Monitoring
Last test execution time ----> Jul 21 2011 20:25:20
First test failure time -----> n/a
Last test failure time -----> n/a
Last test pass time -----> Jul 21 2011 20:25:20
Total failure count -----> 0
Consecutive failure count ---> 0

```

```

4) supervisor-rx-errors -----> .

```

```

Error code -----> 0 (DIAG_SUCCESS)
Total run count -----> 1
Last test testing type -----> Health Monitoring
Last test execution time ----> Jul 21 2011 20:25:20

```



```

First test failure time ----> n/a
Last test failure time -----> n/a
Last test pass time -----> Jul 21 2011 20:25:20
Total failure count -----> 0
Consecutive failure count ---> 0

```

**Note**

To ensure that the maximum number of ports are tested, ensure that both supervisor engines are present on power-up.

Supervisor Engine 6-E and Supervisor Engine 6L-E

```
Switch# show diagnostic result module 6 detail
```

```
Current bootup diagnostic level: minimal
```

```
module 6: SerialNo :
```

```
Overall Diagnostic Result for module 6 : PASS
```

```
Diagnostic level at card bootup: minimal
```

```
Test results: (. = Pass, F = Fail, U = Untested)
```

```
1) supervisor-bootup -----> .
```

```

Error code -----> 0 (DIAG_SUCCESS)
Total run count -----> 1
Last test testing type -----> n/a
Last test execution time ----> Jul 21 2011 13:35:55
First test failure time -----> n/a
Last test failure time -----> n/a
Last test pass time -----> Jul 21 2011 13:35:55
Total failure count -----> 0
Consecutive failure count ---> 0

```

```
Power-On-Self-Test Results for STANDBY Supervisor
```

```
prod: WS-X45-SUP6-E part: 73-10597-04 serial: JAE1132SXQL
```

```
Power-on-self-test for Module 6: WS-X45-SUP6-E
```

```
CPU Subsystem Tests ...
```

```
seeprom: Pass
```

```
Traffic: L3 Loopback ...
```

```
Test Results: Pass
```

```
Traffic: L2 Loopback ...
```

```
Test Results: Pass
```

```
Switching Subsystem Memory ...
```

```
Packet Memory Test Results: Pass
```

```
Module 6 Passed
```

```
Remote TenGigabitPort status: Untested
```

```
2) linecard-online-diag -----> .
```

```

Error code -----> 0 (DIAG_SUCCESS)
Total run count -----> 1
Last test testing type -----> n/a
Last test execution time ----> Jul 21 2011 13:35:55
First test failure time -----> n/a
Last test failure time -----> n/a
Last test pass time -----> Jul 21 2011 13:35:55
Total failure count -----> 0
Consecutive failure count ---> 0

```

Slot	Ports	Card Type	Diag Status	Diag Details
6	6	Sup 6-E 10GE (X2), 1000BaseX (SFP)	Passed	None

```

Detailed Status
-----
. = Pass          U = Unknown
L = Loopback failure  S = Stub failure
P = Port failure
E = SEEPROM failure  G = GBIC integrity check failure

```

```

Ports 1  2  3  4  5  6
      .  .  .  .  .  .

```

```

3) stub-rx-errors -----> .

Error code -----> 0 (DIAG_SUCCESS)
Total run count -----> 3
Last test testing type -----> Health Monitoring
Last test execution time ----> Jul 21 2011 13:39:06
First test failure time -----> n/a
Last test failure time -----> n/a
Last test pass time -----> Jul 21 2011 13:39:06
Total failure count -----> 0
Consecutive failure count ---> 0

```

```

4) supervisor-rx-errors -----> .

Error code -----> 0 (DIAG_SUCCESS)
Total run count -----> 3
Last test testing type -----> Health Monitoring
Last test execution time ----> Jul 21 2011 13:39:06
First test failure time -----> n/a
Last test failure time -----> n/a
Last test pass time -----> Jul 21 2011 13:39:06
Total failure count -----> 0
Consecutive failure count ---> 0

```

Switch#

Troubleshooting the Test Failures

A failure of any of the POST tests reflects a problem with the hardware on the supervisor engine. Cisco IOS boots the supervisor engine with limited functionality, allowing you to evaluate and display the diagnostic test results. To determine the failure cause, do one of the following:

- Evaluate whether the hardware failure is persistent by power cycling the supervisor engine to rerun the POST tests.

- Remove and reinsert the supervisor engine into the chassis to ensure that the seating is correct. Contact Cisco Systems customer support team for more information.

**Note**

On a redundant chassis, concurrent POST is supported on supervisor engines that are already inserted. However, if a second supervisor engine is inserted while the first one is loading, you might boot the first supervisor engine in a faulty Cisco IOS state (POST will abort, and some of the POST's tests will be bypassed). This situation only happens during concurrent bootup of the supervisor engines. You should not insert any additional supervisor engines in the empty supervisor engine slot while an already seated supervisor engine is running POST. The POST sequence is completed when the "Exiting to ios..." message is displayed.



Configuring WCCP Version 2 Services

This chapter describes how to configure the Catalyst 4500 Series Switches to redirect traffic to content engines (web caches) using Web Cache Communication Protocol (WCCP) Version 2



Note

Throughout this chapter, WCCP refers to WCCP Version 2. Version 1 is *not* supported.

This chapter consists of these sections:

- [Understanding WCCP, page 83-1](#)
- [Restrictions for WCCP, page 83-5](#)
- [Configuring WCCP, page 83-5](#)
- [Verifying and Monitoring WCCP Configuration Settings, page 83-9](#)
- [WCCP Configuration Examples, page 83-9](#)



Note

The tasks in this chapter assume that you have already configured content engines on your network. For specific information on hardware and network planning associated with Cisco Content Engines and WCCP, see the product literature and documentation links available on Cisco.com:

http://www.cisco.com/en/US/docs/ios/12_2/configfun/configuration/guide/fcf018_ps1835_TSD_Products_Configuration_Guide_Chapter.html

and

http://www.cisco.com/en/US/tech/tk122/tk717/tsd_technology_support_protocol_home.html

Understanding WCCP

These sections describe WCCP:

- [Overview, page 83-2](#)
- [Hardware Acceleration, page 83-2](#)
- [Understanding WCCP Configuration, page 83-3](#)
- [WCCP Features, page 83-3](#)

Overview

The Cisco IOS WCCP feature allows use of Cisco Content Engines (or other content engines running WCCP) to localize web traffic patterns in the network, enabling content requests to be fulfilled locally. Traffic localization reduces transmission costs and download time.

WCCP enables Cisco IOS routing platforms to transparently redirect content requests. The main benefit of transparent redirection of HTTP/non-HTTP requests is that users need not configure their browsers to use a web proxy. Instead, they can use the target URL to request content, and have their requests automatically redirected to a content engine. The word “transparent” in this case means that the end user does not know that a requested file (such as a web page) came from the content engine instead of from the originally specified server.

When a content engine receives a request, it attempts to service it from its own local content. If the requested information is not present, the content engine issues its own request to the originally targeted server to get the required information. When the content engine retrieves the requested information, it forwards it to the requesting client and caches it to fulfill future requests, thus maximizing download performance and substantially reducing transmission costs.

WCCP enables a series of content engines, called a *content engine cluster*, to provide content to a router or multiple routers. Network administrators can easily scale their content engines to handle heavy traffic loads using these clustering capabilities. Cisco clustering technology enables each content member to work in parallel, resulting in linear scalability. Clustering content engines greatly improves the scalability, redundancy, and availability of your caching solution. You can cluster up to 32 content engines to scale to your desired capacity.

Beginning in Cisco IOS XE Release 3.7.1E, WCCPv2 is supported on Virtual Switching System (VSS).

Beginning in Cisco IOS XE Release 3.8.0E, WCCPv2 supports traffic redirection to and from Virtual Routing and Forwarding (VRF) interfaces. Ensure that you configure the content engine running WCCP such that the forward and return traffic, to and from the content engine, is from interfaces that are a part of the same VRF. The VRF used for WCCP on an interface should match the VRF configured on that interface.

The feature is supported on Cisco Catalyst 4500E Series Switches with Supervisor Engine 9-E, 8-E, 7-E and Cisco Catalyst 4500-X switches, on the IP Base image and the Enterprise Services image.

Hardware Acceleration

Hardware Acceleration is enabled by default on Catalyst 4500 series switches. Layer 2 rewrite forwarding and Layer 2 return method are supported in hardware.

When the switch exhausts hardware (TCAM) or software resources, traffic is redirected in software. GRE return method is supported only in software.

Configure a directly connected content engine to negotiate use of the WCCP Layer 2 Redirection feature (with load balancing) based on the mask assignment table. The **show ip wccp web-cache detail** command displays the redirection method in use for each cache.

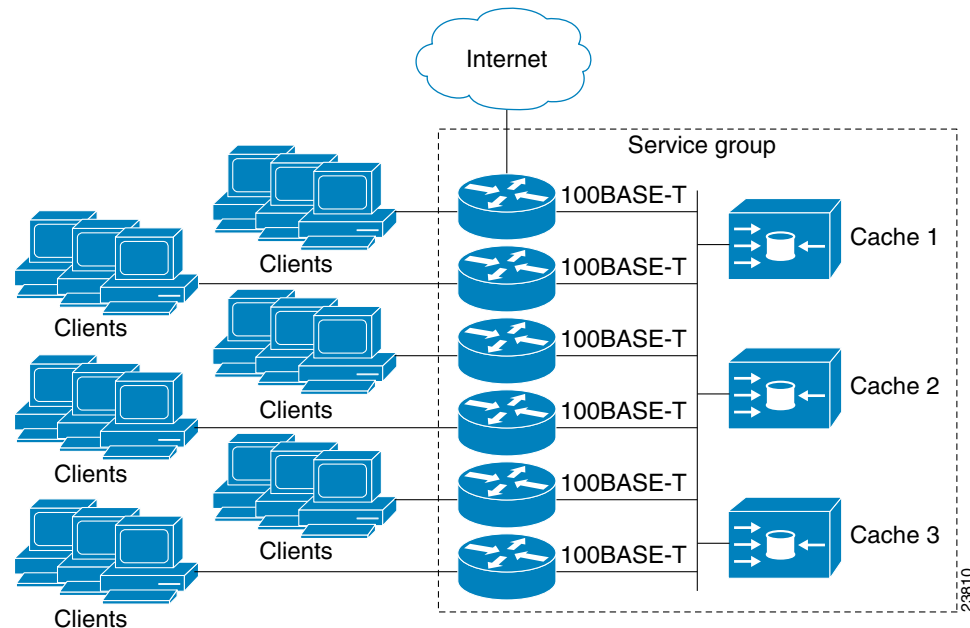
**Note**

You can configure the Cisco Content Engine Release 2.2 or later to use the WCCP Layer 2 Redirection feature with the mask assignment table.

Understanding WCCP Configuration

Multiple routers can use WCCP to service a cache cluster. [Figure 83-1](#) illustrates a sample configuration using multiple routers.

Figure 83-1 Cisco Content Engine Network Configuration Using WCCP



The subset of content engines within a cluster and routers connected to the cluster that are running the same service is known as a *service group*. Available services include TCP and User Datagram Protocol (UDP) redirection.

WCCP requires that each content engine be aware of all the routers in the service group. You must a list of IP addresses for each of the routers in the group configured on each content engine. The address of each router in the group must be explicitly specified for each content engine during configuration.

The following sequence of events describe how WCCP works:

1. Each WCCP client (content engine) is configured with a list of WCCP servers (routers).
2. Each content engine announces its presence with a `Here I Am` message and a list of routers with which it has established communication. The routers reply with their view (list) of content engines in the service group through `I See You` messages.
3. If the view is consistent across all content engines in the cluster, one content engine is designated as the lead and sets the policy that the switches deploy in redirecting traffic.

WCCP Features

These sections describe WCCP features:

- [HTTP and Non-HTTP Services Support](#)
- [Multiple Routers Support](#)
- [MD5 Security](#)

- [Web Content Packet Return](#)

HTTP and Non-HTTP Services Support

WCCP enables redirection of HTTP traffic (TCP port 80 traffic), as well as non-HTTP traffic (TCP and UDP). WCCP supports the redirection of packets intended for other ports, including those used for proxy-web cache handling, File Transfer Protocol (FTP) caching, FTP proxy handling, web caching for ports other than 80, and real audio, video, and telephony applications.

To accommodate the various types of services available, WCCP introduces the concept of multiple *service groups*. Service information is specified in the WCCP configuration commands using dynamic services identification numbers (such as 98) or a predefined service keywords (such as web-cache). This information is used to validate that service group members are all using or providing the same service.



Note

The Catalyst 4500 series switch supports up to eight service groups.

For information on supported WCCP version 2 services with ACNS version 5.2 software, refer to the *Release Notes for Cisco ACNS Software, Release 5.2.3*.

The content engines in service group specify traffic to be redirected by protocol (TCP or UDP) and port (source or destination). Each service group has a priority level assigned to it. Packets are matched against service groups in priority order and redirected by the highest priority service group that matches traffic characteristics.

Multiple Routers Support

WCCP enables you to attach multiple routers to a cluster of cache engines. The use of multiple routers in a service group enables redundancy, interface aggregation, and distribution of the redirection load.

MD5 Security

WCCP provides optional authentication that enables you to control which routers and content engines become part of the service group using passwords and the HMAC MD5 standard. Shared-secret MD5 one-time authentication (set using the **ip wccp [password [0-7] password]** global configuration command) enables messages to be protected against interception, inspection, and replay.

Web Content Packet Return

If a content engine is unable to provide a requested object it has cached due to error or overload, the content engine returns the request to the router for onward transmission to the originally specified destination server. WCCP verifies which requests have been returned from the content engine unserved. Using this information, the router can then forward the request to the originally targeted server (rather than attempting to resend the request to the content cluster). This provides error handling transparency to clients.

Typical reasons why a content engine would reject packets and initiate the packet return feature include the following:

- Instances when the content engine is overloaded and has no room to service the packets.
- Instances when the content engine is filtering for certain conditions that make caching packets counterproductive (such as, when IP authentication has been turned on).

Restrictions for WCCP

The following limitations apply to WCCP:

- Time To Live (TTL) value of Layer 3 switches servicing a cluster must be 15 seconds or less.
- A service group can comprise up to 32 content engines and 32 devices.
- All the content engines in a cluster must be configured to communicate with all the devices servicing the cluster.
- A total of eight active IPv4 and IPv6 service groups are supported on a switch. If used in pairs, up to four service-group pairs can be configured simultaneously.
- The Layer 2 rewrite forwarding method is supported in the hardware.
- The Layer 2 return method is supported in the hardware and is recommended.
- The content engine must be directly connected to the device.
- Input/output redirection configuration is not supported on content engines facing interfaces.
- WCCPv2 supports up to 256 distinct masks. However, a Catalyst 4500 series switch only supports a single mask.

Configuring WCCP

The following configuration tasks assume that you have already installed and configured the content engines you want to include in your network. You must configure the content engines in the cluster before configuring WCCP on your device. Refer to the [Cisco Content Engine User Guide](#) for content engine configuration and setup tasks.

IP must be configured on the device interface connected to the cache engines. Examples of device configuration tasks follow this section. For complete descriptions of the command syntax, refer to the *Cisco IOS Configuration Fundamentals Command Reference, Cisco IOS Release 12.3*.

These sections describe how to configure WCCP:

- [Configuring a Service Group Using WCCP, page 83-5](#) (Required)
- [Using Access Lists for a WCCP Service Group, page 83-8](#) (Optional)
- [Setting a Password for a Switch and Cache Engine, page 83-8](#) (Optional)

Configuring a Service Group Using WCCP

WCCP uses service groups based on logical redirection services. The standard service is the content engine, which intercepts TCP port 80 (HTTP) traffic and redirects that traffic to the content engines. This service is referred to as a well-known service, because the characteristics of the web cache service are known by both the router and content engines. A description of a well-known service is not required beyond a service identification (the command line interface (CLI) provides a **web-cache** keyword in the command syntax).

For information on supported WCCP services with ACNS version 5.2 software, refer to the *Release Notes for Cisco ACNS Software, Release 5.2.3*.

In addition to the web cache service, there can be up to seven dynamic services running concurrently on the switch.

**Note**

More than one service can run on a switch at the same time, and routers and content engines can be part of multiple service groups at the same time.

The dynamic services are defined by the content engines; the content engine instructs the router which protocol or ports to intercept, and how to distribute the traffic. The router itself does not have information on the characteristics of the dynamic service group's traffic, because this information is provided by the first content engine to join the group. In a dynamic service, up to eight ports can be specified within a single protocol TCP or UDP).

Cisco Content Engines, for example, use dynamic service 99 to specify a reverse-proxy service. However, other content engines may use this service number for some other service. The following configuration information deals with enabling general services on Cisco routers. Refer to the content engine documentation for information on configuring services on content engines.

To enable a service on a Catalyst 4500 series switch, perform this task:

	Command	Purpose
Step 1	<pre>Switch(config)# ip wccp [vrf vrf-name] [group-address groupaddress] [redirect-list access-list] [group-list access-list] [password password]</pre> <p>For IPv6</p> <pre>Switch(config)# ipv6 wccp [vrf vrf-name] [group-address groupaddress] [redirect-list access-list] [group-list access-list]</pre>	<p>Specifies the following:</p> <ul style="list-style-type: none"> • A dynamic service to enable on the switch, • The IP multicast address used by the service group (optional) • The redirect access list to control the traffic to be redirected (optional) • The group list to use for content engine membership (optional) • Whether to use MD5 authentication (optional) <p>Enables the WCCP service.</p>
Step 2	<pre>Switch(config-if)# [no] ip wccp check services all</pre> <p>For IPv6</p> <pre>Switch(config-if)# [no] ipv6 wccp check services all</pre>	<p>If a service matches the packet and the service has a redirect access list configured, then the IP packet will be checked against the access list. If the packet is rejected by the access-list, the packet will not be passed down to lower priority services unless the ip wccp check services all command is configured. After the ip wccp check services all command is configured, WCCP will continue to attempt to match the packet against any remaining low priority services configured on the interface.</p>
Step 3	<pre>Switch(config)# interface type number</pre>	<p>Specifies the client interface to be configured and enters interface configuration mode.</p>
Step 4	<pre>Switch(config-if)# ip wccp [vrf vrf-name]{web-cache service-number} redirect {in out}</pre> <p>For IPv6</p> <pre>Switch(config-if)# ipv6 wccp [vrf vrf-name] redirect in</pre>	<p>For IPv4, enables WCCP redirection for ingress or egress traffic on the specified client interface.</p> <p>For IPv6, enables WCCP redirection for ingress traffic on the specified client interface.</p>

	Command	Purpose
Step 5	Switch(config)# interface <i>type number</i>	Specifies the interface to be configured for egress redirection exclusion
Step 6	Switch(config-if)# ip wccp redirect exclude in	Specifies that packets received on this interface be excluded from egress redirection. This command MUST be configured on the content engine interface if Layer 2 return method is used by the content engine and egress redirection is configured on the server interface.

Specifying a Web Cache Service

To configure a web cache service and ingress redirection for IPv4 perform this task:

	Command	Purpose
Step 1	Switch(config)# ip wccp [vrf <i>vrf-name</i>] web-cache	Enables the web cache service on the switch.
Step 2	Switch(config)# interface <i>type number</i>	Targets a client interface number for which the web cache service runs, and enters interface configuration mode.
Step 3	Switch(config-if)# ip wccp [vrf <i>vrf-name</i>] web-cache redirect in	Enables the verification on packets to determine if they qualify to be redirected to a content engine, using the client interface specified in Step 2.

To configure a web cache service and egress redirection for IPv4 traffic, perform this task:

	Command	Purpose
Step 1	Switch(config)# ip wccp [vrf <i>vrf-name</i>] web-cache	Enables the web cache service on the switch.
Step 2	Switch(config)# interface <i>type number</i>	Targets a server interface number for the web cache service, and enters interface configuration mode.
Step 3	Switch(config-if)# ip wccp web-cache redirect out	Enables the verification on packets to determine if they qualify to be redirected to a content engine, using the client interface specified in Step 2.
Step 4	Switch(config)# interface <i>type number</i>	Specifies the content engine interface number, and enters interface configuration mode.
Step 5	Switch(config-if)# ip wccp redirect exclude in	Specifies that packets received on this interface be excluded from egress redirection. This prevents the packets returned by the content engine through the L2-return method or the packets generated by the content engine from being redirected back to the content engine.

Using Access Lists for a WCCP Service Group

A Catalyst 4500 series switch can use an access list to restrict the content engines that can join a service group.

To restrict a content engine, perform this task:

	Command	Purpose
Step 1	Switch(config)# access-list <i>access-list</i> permit ip host <i>host-address</i> [<i>destination-address</i> <i>destination-host</i> any]	Creates an access list based on the unicast address of the content engines.
Step 2	Switch(config)# ip wccp web-cache redirect-list <i>access-list</i> For IPv6 Switch(config)# ipv6 wccp [vrf <i>vrf-name</i>]{ <i>service-number</i> } redirect-list <i>access-list</i>	Indicates to the switch which content engines are allowed or disallowed to form a service group.

Setting a Password for a Switch and Cache Engine

MD5 password security requires that each content engine and Catalyst 4500 series switch that wants to join a service group be configured with the service group password. The password can consist of up to seven characters. Each content engine or Catalyst 4500 series switch in the service group authenticates the security component in a received WCCP packet immediately after validating the WCCP message header. Packets failing authentication are discarded.

To configure an MD5 password for use by the Catalyst 4500 series switch in WCCP communications, perform this task:

Command	Purpose
Switch(config)# ip wccp web-cache password <i>password</i>	Sets an MD5 password on the Catalyst 4500 series switch.

Verifying and Monitoring WCCP Configuration Settings

To verify and monitor the configuration settings for WCCP, use the following commands in EXEC mode:

Command	Purpose
Switch# show ip wccp [vrf vrf-name] [web-cache service-number] For IPv6 Switch# show ipv6 wccp [vrf vrf-name] Switch# show ip wccp [vrf vrf-name] { web-cache service-number } detail For IPv6 Switch# show ipv6 wccp [vrf vrf-name] detail Switch# show ip interface	Displays global information related to WCCP, including the protocol version that is currently running, the number of content engines in the routers service group, the content engine group is allowed to connect to the device, and the access list being used. Queries the device for information on which content engines of a specific service group that the device has detected. The information can be displayed for either the web cache service or for the specified dynamic service. Displays the status about whether redirection commands are configured on a client interface. For example, Web Cache Redirect is enabled / disabled.
Switch# show ip wccp [vrf vrf-name] { web-cache service-number } view For IPV6 Switch# show ipv6 wccp [vrf vrf-name] view	Displays the devices in a particular service group that have been detected and the content engines that are not visible to all other devices to which the current device is connected. The view keyword indicates a list of addresses of the service group. The information can be displayed for either the web cache service or the specified dynamic service. Note For further troubleshooting information, use the show ip wccp { web-cache service number } service command.

WCCP Configuration Examples

This section provides the following configuration examples:

- [Example: Performing a General WCCP Configuration, page 83-10](#)
- [Example: Running a Web Cache Service, page 83-10](#)
- [Example: Running a Reverse Proxy Service, page 83-10](#)
- [Example: Running TCP-Promiscuous Service, page 83-11](#)
- [Example: Running Redirect Access List, page 83-12](#)
- [Example: Using Access Lists, page 83-12](#)
- [Example: Setting a Password for a Switch and Content Engines, page 83-13](#)
- [Example: Verifying WCCP Settings, page 83-13](#)

Example: Performing a General WCCP Configuration

The following example shows a general WCCP configuration session. VLAN 20 is for the client interface. VLAN 50 is for the content engine interface.

```
Switch# configure terminal
Switch(config)# ip wccp web-cache group-address 224.1.1.100 password alaska1
Switch(config)# interface vlan 20
Switch(config-if)# ip wccp web-cache redirect in
Switch(config)# interface vlan 50
Switch(config-if)# ip wccp web-cache group-listen
```

The following example shows a general IPv6 WCCP configuration where GigabitEthernet 0/1/0 is the client interface and GigabitEthernet 0/2/0 is the content engine interface:

```
Switch# configure terminal
Switch(config)# ipv6 wccp interface GigabitEthernet 0/1/0
Switch(config)# ipv6 wccp check services all
Switch(config)# interface GigabitEthernet 0/1/0
Switch(config-if)# ipv6 wccp redirect in
Switch(config)# interface GigabitEthernet 0/2/0
```

Example: Running a Web Cache Service

The following example shows a web cache service configuration session with ingress redirection, for IPv4:

```
Switch# configure terminal
Switch(config)# ip wccp web-cache
Switch(config)# interface vlan 20
Switch(config-if)# ip wccp web-cache redirect in
Switch# copy running-config startup-config
Switch# show ip interface vlan 20 | include WCCP Redirect
```

```
WCCP Redirect inbound is enabled
WCCP Redirect exclude is disabled
```

The following example shows a web-cache service configuration session with ingress redirection, for an IPv4 VRF interface:

```
Switch# configure terminal
Switch(config)# ip wccp vrf test web-cache
Switch(config)# interface vlan 10
Switch(config-if)# vrf forwarding test
Switch(config-if)# ip wccp vrf test web-cache redirect in
Switch# copy running-config startup-config
```

Example: Running a Reverse Proxy Service

**Note**

The WCCP reverse proxy service is not supported for IPv6 traffic.

The following example assumes you are configuring a service group using Cisco Content Engines, which use dynamic service 99 to run a reverse proxy service. The following example illustrates how to configure egress redirection, where VLAN 40 reflects the server interface and VLAN 50 reflects the content engine interface:

```
Switch# configure terminal
Switch(config)# ip wccp 99
Switch(config)# interface vlan 40
Switch(config-if)# ip wccp 99 redirect in
Switch(config)# interface vlan 50
Switch(config-if)# ip wccp redirect exclude in
```

For IPv6

```
Switch# configure terminal
Switch(config)# ipv6 wccp 99
Switch(config)# interface vlan 40
Switch(config-if)# ipv6 wccp 99 redirect in
Switch(config)# interface vlan 50
```

Example: Running TCP-Promiscuous Service

The following example shows how to configure TCP promiscuous service, where VLAN 40 represents the server interface and VLAN 50 represents the content engine interface:

```
Switch# configure terminal
Switch(config)# ip wccp 61
Switch(config)# ip wccp 62
Switch(config)# interface vlan 30
Switch(config-if)# ip wccp 61 redirect in
Switch(config)# interface vlan 40
Switch(config-if)# ip wccp 62 redirect in
Switch(config)# interface vlan 50
Switch(config-if)# ip wccp redirect exclude in
```

For IPv6

```
Switch# configure terminal
Switch(config)# ipv6 wccp 51
Switch(config)# ipv6 wccp 52
Switch(config)# interface vlan 30
Switch(config-if)# ipv6 wccp 51 redirect in
Switch(config)# interface vlan 40
Switch(config-if)# ipv6 wccp 52 redirect in
```

The following example shows how to configure the TCP promiscuous service for IPv4 VRF interfaces, where VLAN 40 represents the server interface and VLAN 50 represents the content engine interface:

```
Switch# configure terminal
Switch(config)# ip wccp vrf abc 91
Switch(config)# ip wccp vrf abc 92
Switch(config)# interface vlan 30
Switch(config-if)# vrf forwarding abc s
Switch(config-if)# ip wccp vrf abc 91 redirect in
Switch(config)# interface vlan 40
Switch(config-if)# vrf forwarding abc
Switch(config-if)# ip wccp vrf abc 92 redirect in
Switch(config)# interface vlan 50
Switch(config-if)# vrf forwarding abc
```

Example: Running Redirect Access List

The following example shows how to redirect traffic only from subnet 10.1.1.0:

```
Switch(config)# ip access-list extended 100
Switch(config-ext-nacl)# permit ip 10.1.1.0 255.255.255.0 any
Switch(config-ext-nacl)# exit
Switch(config)# ip wccp web-cache redirect-list 100
Switch(config)# interface vlan 40
Switch(config-if)# ip wccp web-cache redirect in
Switch(config)# interface vlan 50
Switch(config-if)# ip wccp redirect exclude in
```

The following example shows how to redirect IPv6 traffic only from 2001::1/64 2004::1/64 eq www:

```
switch(config)# ipv6 access-list ACL_1
switch(config-ipv6-acl)# permit tcp 2001::1/64 2004::1/64 eq www
switch(config-ipv6-acl)# exit
switch(config)# ipv6 wccp 61 redirect-list ACL_1
switch(config)# interface vlan 40
switch(config-if)# ipv6 wccp 61 redirect in
```

Example: Using Access Lists

To achieve better security, you can use a standard access list to notify the Catalyst 4500 series switch to which IP addresses are valid for a content engine attempting to register with the current switch. The following example shows a standard access list configuration session where the access list number is 10 for some sample hosts:

```
switch(config)# access-list 10 permit host 11.1.1.1
switch(config)# access-list 10 permit host 11.1.1.2
switch(config)# access-list 10 permit host 11.1.1.3
switch(config)# ip wccp web-cache group-list 10
```

The following examples shows a standard access list configuration for IPv6:

```
switch(config)#ipv6 access-list ACL_1
switch(config-ipv6-acl)#permit tcp 2001::1/64 2004::1/64 eq www
switch(config)#ipv6 wccp 61 redirect-list ACL_1
```


Example: Setting a Password for a Switch and Content Engines

The following example shows a WCCP password configuration session where the password is `alaska1`:

```
Switch# configure terminal
Switch(config)# ip wccp web-cache password alaska1
```

Example: Verifying WCCP Settings

To verify your configuration changes, use the **more system:running-config EXEC** command. The following example shows that both the web cache service and dynamic service 99 are enabled on the Catalyst 4500 series switch:

WCCP Unicast Mode

```
Switch# more system:running-config

Building configuration...
Current configuration:
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service udp-small-servers
service tcp-small-servers
!
enable secret 5 $1$nSVy$faliJsVQXVPW.KuCxZNTh1
enable password alabama1
!
ip subnet-zero
ip wccp web-cache
ip wccp 99
!
interface Vlan200
ip address 10.3.1.2 255.255.255.0
ip wccp web-cache redirect in

interface Vlan300
ip address 10.4.1.1 255.255.255.0
ip wccp redirect exclude in

interface Vlan400
ip address 10.5.1 255.255.255.0
ip wccp 99 redirect out

ip default-gateway 10.3.1.1
ip classless
ip route 0.0.0.0 0.0.0.0 10.3.1.1
no ip http server
!
!
```




Configuring MIB Support

This chapter describes how to configure SNMP and MIB support for the Cisco 4500 series switch. It includes the following sections:

- [Determining MIB Support for Cisco IOS Releases, page 84-1](#)
- [Using Cisco IOS MIB Tools, page 84-1](#)
- [Downloading and Compiling MIBs, page 84-2](#)
- [Enabling SNMP Support, page 84-4](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference*, you can locate it in the [Cisco IOS Master Command List, All Releases](#).

Determining MIB Support for Cisco IOS Releases

To determine which MIBs are included in the Cisco IOS release running on the Cisco 4500 series switch, follow these steps:

- Step 1** Go to the Cisco MIBs Support page:
<http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml>
- Step 2** Under Cisco Access Products, select a **Cisco 4500 switch** to display a list of MIBs supported on the Cisco 4500 switches.
- Step 3** Scroll through the list to find the release you are interested in.

Using Cisco IOS MIB Tools

This section describes how to access the Cisco MIB tools page. The MIB Locator finds MIBs in Cisco IOS software releases. You can find general MIB information, instructions about how to use the SNMP Object Navigator which translates SNMP object identifiers (OIDs) into SNMP names, and how to load Cisco MIBs.

To access the Cisco IOS MIB tools site, follow these steps:

Step 1 Go to the Cisco Products and Services page:
<http://tools.cisco.com/ITDIT/MIBS/servlet/index>

Step 2 Click **MIB Locator** to launch the application.

The MIB Locator application allows you to find a MIB in the following three ways.

- From the MIB Locator page, you can:
 - Click the drop-down menu and select the desired Cisco IOS software release.
 - From the Platform Family menu, select the appropriate feature set: **CAT4500-SUP2-PLUS**, **CAT4500-SUP2-PLUS-TS**, **CAT4500-SUP3**, **CAT4500-SUP4**, **CAT4500-SUP5**, **CAT4500-SUP5-10gGE2**. If you select the platform first, the system displays only those releases and feature sets that apply to the Cisco 4500 series switch.
 - From the Feature Set menu, select **Service Provider W/VIP**.
- From the MIB Locator page, you can search by image name. For example, enter the following and click the **Submit** button:
 c7200-js56i-mz.12.0-1
- From the MIB Locator page, you can search for the MIB from the list of MIBs in the menu. You can select one, or for multiple selections, hold down the **CTRL** key, then click the **Submit** button.



Note After you make a selection, follow the links and instructions.

Downloading and Compiling MIBs

The following sections provide information about how to download and compile MIBs for the Cisco 4500 series switch:

- [Guidelines for Working with MIBs, page 84-2](#)
- [Downloading MIBs, page 84-3](#)
- [Compiling MIBs, page 84-4](#)

Guidelines for Working with MIBs

While working with MIBs, consider the following guidelines:

- Mismatches on datatype definitions might cause compiler errors or warning messages. Although Cisco MIB datatype definitions are not mismatched, some standard RFC MIBs do mismatch. For example:

```
MIB A defines: SomeDatatype ::= INTEGER(0..100)
MIB B defines: SomeDatatype ::= INTEGER(1..50)
```

This example is considered to be a trivial error and the MIB loads successfully with a warning message.

The next example is considered as a nontrivial error (even though the two definitions are essentially equivalent), and the MIB is not successfully parsed.

```
MIB A defines: SomeDatatype ::= DisplayString
MIB B defines: SomeDatatype ::= OCTET STRING (SIZE(0..255))
```

If your MIB compiler treats these as errors, or you want to delete the warning messages, edit one of the MIBs that define this same datatype so that the definitions match.

- Many MIBs import definitions from other MIBs. If your management application requires MIBs to be loaded, and you experience problems with undefined objects, you might want to load the following MIBs in this order:

```
SNMPv2-SMI.my
SNMPv2-TC.my
SNMPv2-MIB.my
RFC1213-MIB.my
IF-MIB.my
CISCO-SMI.my
CISCO-PRODUCTS-MIB.my
CISCO-TC.my
```

- For additional information and SNMP technical tips, go to the following URL:
http://www.cisco.com/en/US/tech/tk648/tk362/technologies_tech_note09186a0080094aa5.shtml
- For a list of SNMP OIDs assigned to MIB objects, go to the following URL and click on **SNMP Object Navigator** and follow the links:
<http://tools.cisco.com/ITDIT/MIBS/servlet/index>



Note You must have a Cisco CCO name and password to access the MIB Locator.

- For information about how to download and compile Cisco MIBs, go to the following URL:
http://www.cisco.com/en/US/tech/tk648/tk362/technologies_tech_note09186a00800b4cee.shtml

Downloading MIBs

to download the MIBs onto your system if they are not already present, follow these steps:

-
- Step 1** Review the guidelines in the previous section (“[Guidelines for Working with MIBs](#)”).
- Step 2** Go to one of the following Cisco URLs. If the MIB you want to download is not there, try the other URL; otherwise, go to one of the URLs in Step 5.
- <ftp://ftp.cisco.com/pub/mibs/v2>
- <ftp://ftp.cisco.com/pub/mibs/v1>
- Step 3** Click the link for a MIB to download that MIB to your system.
- Step 4** Select **File > Save** or **File > Save As** to save the MIB on your system.
- Step 5** You can download industry-standard MIBs from the following URL:
- <http://www.oidview.com/mibs/0/md-0-1.html>
-

Compiling MIBs

If you plan to integrate the Cisco 4500 series switch with an SNMP-based management application, then you must also compile the MIBs for that platform. For example, if you are running HP OpenView on a UNIX operating system, you must compile Cisco 4500 series switch MIBs with the HP OpenView Network Management System (NMS). For instructions, see the NMS documentation.

Enabling SNMP Support

The following procedure summarizes how to configure the Cisco 4500 series switch for SNMP support.

For detailed information about SNMP commands, see the following Cisco documents:

- *Cisco IOS Release 15.0 Configuration Guides*, available at the following URL:
http://www.cisco.com/en/US/products/ps10591/products_installation_and_configuration_guides_list.html
- *Cisco IOS Release 12.3 Configuration Fundamentals and Network Management Command Reference*, Part 3: System Management Commands, “Router and Network Configuration Commands” section, available at the following URL:
http://www.cisco.com/en/US/docs/ios/12_3/configfun/command/reference/fun_r.html

To configure the Cisco 4500 series switch for SNMP support, follow these steps:

-
- Step 1** Establish your basic SNMP configuration using the command line interface (CLI) on the router. Note that these basic configuration commands are issued for SNMP version 2c. For SNMP version 3, you must also set up SNMP users and groups. Refer to the preceding list of documents for command and set up information.
- Define SNMP read-only and read-write communities:

```
Router (config)# snmp-server community Read_Only_Community_Name ro
Router (config)# snmp-server community Read_Write_Community_Name rw
```
 - Configure SNMP views (to limit the range of objects accessible to different SNMP user groups):

```
Router (config)# snmp-server view view_name oid-tree {included | excluded}
```
- Step 2** Identify (by IP address) the host to receive SNMP notifications from the router:

```
Router (config)# snmp-server host host
```
- Step 3** Configure the router to generate notifications. You can use keywords to limit the number and types of messages generated.

```
Router (config)# snmp-server enable traps [notification-type] [notification-option]
```
- Step 4** (Optional) Configure the router to generate SNMP notifications released to field replaceable units (FRUs):

```
Router (config)# snmp-server enable traps fru-ctrl
```
- Step 5** (Optional) Configure the router to generate SNMP notifications related to environmental monitoring:

```
Router (config)# snmp-server enable traps envmon
```
-



Configuring Easy Virtual Networks

Easy Virtual Network (EVN) is an IP-based virtualization technology that provides end-to-end virtualization of two or more Layer-3 networks. You can use a single IP infrastructure to provide separate virtual networks whose traffic paths remain isolated from each other.

This chapter contains the following sections

- [Prerequisites for Configuring Easy Virtual Network, page 85-1](#)
- [Restrictions for EVN, page 85-1](#)
- [About Easy Virtual Network, page 85-2](#)
- [Configuring Easy Virtual Networks, page 85-14](#)
- [Configuration Examples for Configuring EVN, page 85-18](#)
- [Troubleshooting EVN Configuration, page 85-22](#)

Prerequisites for Configuring Easy Virtual Network

- Implementing EVN in a network requires a single IP infrastructure that you want to virtualize into two or more logical networks or L3VPNs. EVN provides path isolation for the traffic on the different virtual networks.
- You must have a functioning campus design in place before adding virtualization to a network.
- You should understand virtual routing and forwarding (VRF) instances and how they are used to maintain traffic separation across the network.

Restrictions for EVN

- EIGRP command inheritance is not supported on VNET interfaces.
- The **vnet tag** command does not support management VRFs.
- We recommend that you configure a value between 2 and 1000 as the VNET tag. Configuring a value above this range will conflict with the switch internal VLAN assignments.
- An EVN trunk is allowed on any interface that supports 802.1q encapsulation, such as Fast Ethernet, Gigabit Ethernet, and port channels.
- There are additional platform and line-card restrictions for an EVN trunk. Check Cisco Feature Navigator, for supported platforms and line cards.

- A single IP infrastructure can be virtualized to provide up to 32 virtual networks end-to-end.
- If an EVN trunk is configured on an interface, you cannot configure VRF-Lite on the same interface.
- OSPFv3 is not supported; OSPFv2 is supported.
- The following features are not supported by EVN:
 - IS-IS
 - RIP
 - Route replication is not supported with BGP
 - Certain SNMP set operations
- The following are not supported on an EVN trunk:
 - Access control lists (ACLs)
 - BGP interface commands are not inherited
 - IPv6, except on vnet global
 - Network address translation (NAT)
 - NetFlow
 - Web Cache Communication Protocol (WCCP)

About Easy Virtual Network

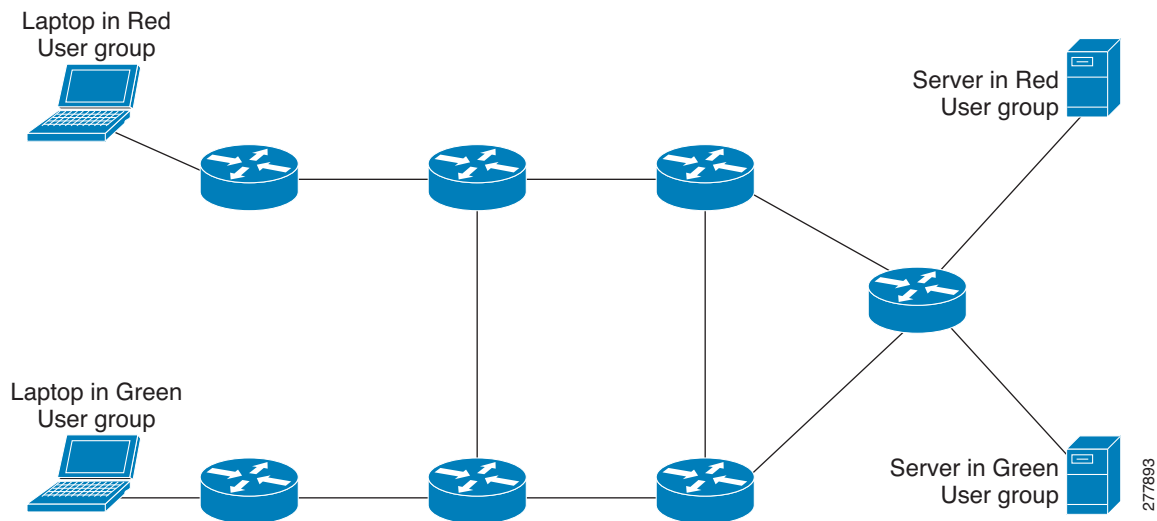
Easy Virtual Network (EVN) builds on the existing IP-based virtualization mechanism known as VRF-Lite. EVN provides enhancements in path isolation, simplified configuration and management, and improved shared service support. EVN is backward compatible with VRF-Lite to enable seamless network migration from VRF-Lite to EVN.

EVN supports IPv4, static routes, Open Shortest Path First version 2 (OSPFv2), and Protocol Independent Multicast (PIM) and Multicast Source Discovery Protocol (MSDP) for IPv4 Multicast routing. EVN also supports Cisco Express Forwarding (CEF) and Simple Network Management Protocol (SNMP).

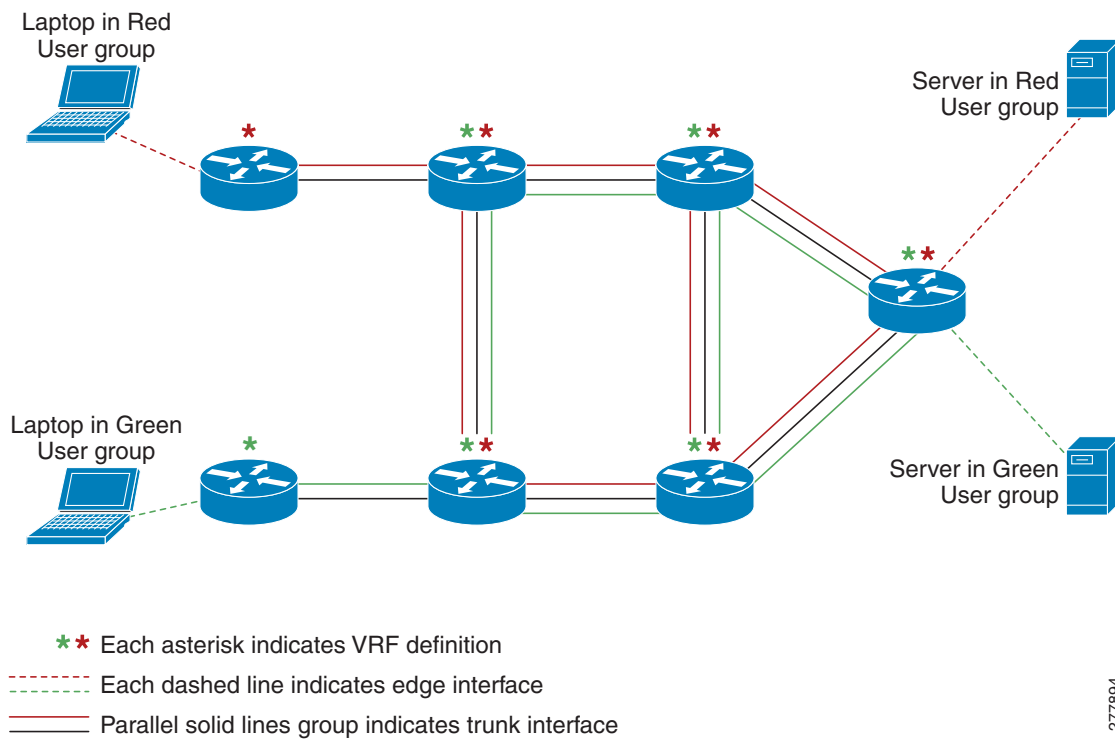
Virtual Network Tags Provide Path Isolation

It is not uncommon to have different user groups running on the same IP infrastructure. Various business reasons require traffic isolation between different groups. The figure below shows two user groups, Red and Green, running on the same network. Prior to network virtualization, there is no separation of traffic between the two groups. Users in the Red user group can access the server in the Green user group, and vice versa.

Without network virtualization, path isolation can be achieved by using access control, which is expensive to maintain, prone to error and does not support unique routing and forwarding tables per network.

Figure 85-1 Network without Virtualization

Virtual networks provide a coarse-grained segmentation of different user groups on one physical network. By configuring virtual networks, you can virtualize a single IP infrastructure to provide a number of virtual networks end to end. In the figure below, a single IP infrastructure is virtualized into two VPNs by creating two VRFs, Red and Green.

Figure 85-2 Network with Virtualization

In addition to utilizing VRFs to provide device-level separation, each virtual network has path isolation from the other. Path isolation is achieved by tagging the traffic so it carries the same tag value throughout the same virtual network. Each network device along the path uses the tags to provide separation among different VRFs. A single tag number ties VRF red, for example, on one device to VRF red on another device.

Virtual Network Tags

Each VPN and associated EVN has a tag value that you assign during configuration. The tag value is global, meaning that on each device, the same EVN must be assigned the same numerical tag value. Tag values range from 2 to 4094.

An EVN is allowed on any interface that supports 802.1q encapsulation, such as Fast Ethernet, Gigabit Ethernet, and port channels. To allow for backward compatibility with the VRF-Lite solution, the vLAN ID field in the 802.1q frame is used to carry the virtual network tag.

Traffic that carries a virtual network tag is called tagged traffic. Traffic that does not carry a virtual network tag is called untagged traffic.

Tags are illustrated in the following configuration with two VRFs, red and green:

```
! Define two VRFs, red and green.
vrf definition red
  vnet tag 101
!
  address-family ipv4
  exit-address-family
!
vrf definition green
  vnet tag 102
!
  address-family ipv4
  exit-address-family
!
```

A virtual network is defined as a VRF instance with a virtual network tag assigned.

vnet Global

A predefined EVN known as vnet global is on the device. It refers to the global routing context and it corresponds to the default RIB. In figure 2 and figure 3, vnet global is represented by a black line connecting devices. The vnet global carries untagged traffic. By default, interfaces belong to the vnet global. Furthermore, vnet global is always running on trunk interfaces. The vnet global is also known as the default routing table.

**Note**

IPv6 traffic is supported in vnet global only.

Edge Interfaces and EVN Trunk Interfaces

User devices are connected to a Layer 2 switch port, which is assigned to a VLAN. A VLAN can be thought of as a Layer 2 VPN. Customers will group all of the devices that need to be supported in a common Layer 3 VPN in a single VLAN. The point where data traffic is handed off between a VLAN and VRF is called an edge interface.

An edge interface connects a user device to the EVN and in effect defines the boundary of the EVN. Edge interfaces connect end devices such as hosts and servers that are not VRF-aware. Traffic carried over the edge interface is untagged. The edge interface classifies which EVN the received traffic belongs to. Each edge interface is configured to belong to only one EVN.

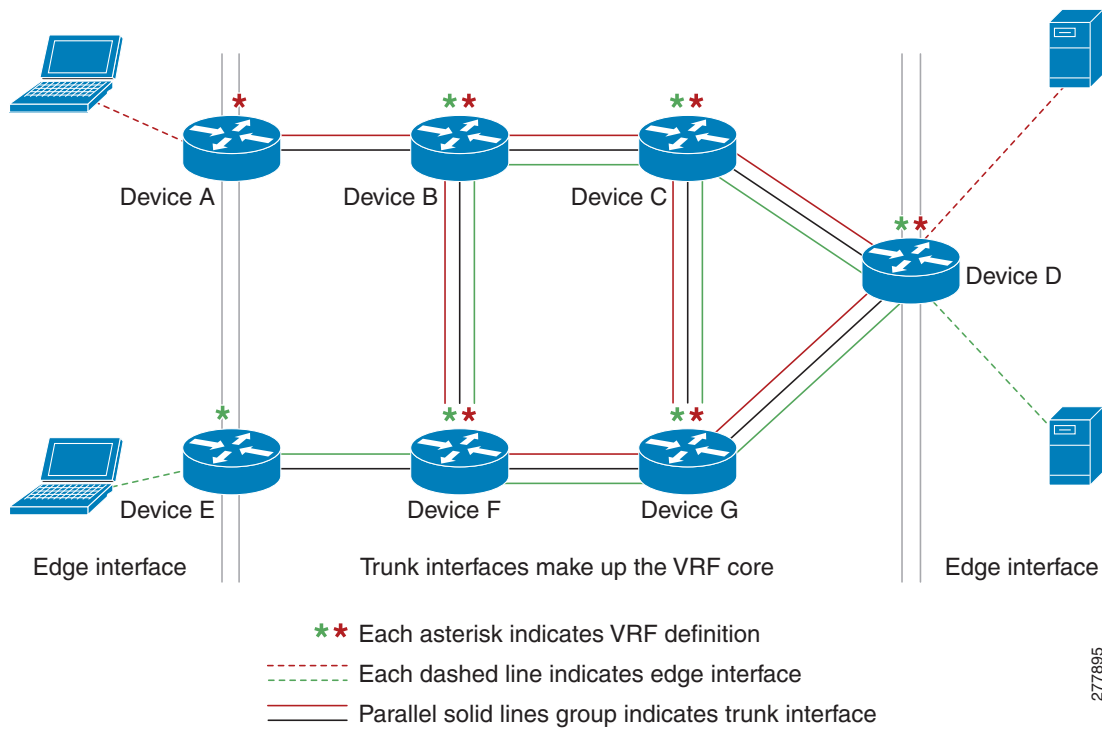
An EVN trunk interface connects VRF-aware devices together and provides the core with a means to transport traffic for multiple EVNs. Trunk interfaces carry tagged traffic. The tag is used to de-multiplex the packet into the corresponding EVN. A trunk interface has one subinterface for each EVN.

The **vnet trunk** command is used to define an interface as an EVN trunk interface.

An EVN interface uses two types of interfaces: edge interfaces and trunk interfaces. An interface can be an edge or trunk interface, but not both. Figure 3 illustrates devices A and D, which have edge interfaces that belong to VRF Red. Devices D and E have edge interfaces that belong to VRF Green.

Devices B, C, D, F, and G have trunk interfaces that make up the EVN core. These five devices have interfaces that belong to both VRF Red and VRF Green.

Figure 85-3 EVN Edge and EVN Trunk Interfaces



Identifying Trunk Interfaces in Display Output

Because a trunk interface carries multiple EVNs, sometimes it is not sufficient to display only the trunk interface name. When it is necessary to indicate that display output pertains to a particular EVN running on the trunk interface, the convention used is append a period and the virtual network tag, making the format interface.virtual-network-tag. Examples are gigabitethernet1/1/1.101 and gigabitethernet1/1/1.102.

By default, when a trunk interface is configured, all of the EVNs and associated virtual network tags are configured, and a virtual network subinterface is automatically created. As stated above, a period and the virtual network tag number are appended to the interface number.

In the following example, VRF red is defined with virtual network tag 3. Hence, the system created Fast Ethernet 0/0/0.3 (in VRF red).

```
Device# show running-config vrf red
```

```
Building configuration...
Current configuration : 1072 bytes
vrf definition red
  vnet tag 3
  !
  address-family ipv4
  exit-address-family
  !
```

You can display this hidden interface with the show derived-config command and see that all of the commands entered on Fast Ethernet 0/0/0 have been inherited by Fast Ethernet 0/0/0.3:

```
Device# show derived-config interface fastethernet0/0/0.3
```

```
Derived configuration : 478 bytes
!
interface FastEthernet0/0/0.3
  description Subinterface for VRF NG red
  vrf forwarding red
  encapsulation dot1Q 3
  ip address 10.1.1.1 255.255.255.0
end
```

Single IP Address on Trunk Interfaces

A trunk interface can carry traffic for multiple EVNs. To simplify the configuration process, all the subinterfaces and associated EVNs have the same IP address assigned. In other words, a trunk interface is identified by the same IP address in different EVN contexts. This is because each EVN has a unique routing and forwarding table, thereby enabling support for overlapping IP addresses across multiple EVNs.

Relationship Between VRFs Defined and VRFs Running on a Trunk Interface

By default, the trunk interfaces on a router will carry traffic for all VRFs defined by the vrf definition command. For example, in the following configuration, every VRF defined on the router is included on the interface:

```
interface FastEthernet 1/0/0 vnet trunk ip address 10.1.1.1 255.255.255.0
```

However, you might want to enable only a subset of VRFs over a certain trunk interface for traffic separation purposes. This is achieved by creating a VRF list, which is referenced in the **vnet trunk** command. When a trunk interface is enabled with a VRF list, only VRFs on the list are enabled on the interface. The exception is that vnet global is always enabled on the trunk interface.

In the following example, only the two specified VRFs on the list (red and green) are enabled on the interface:

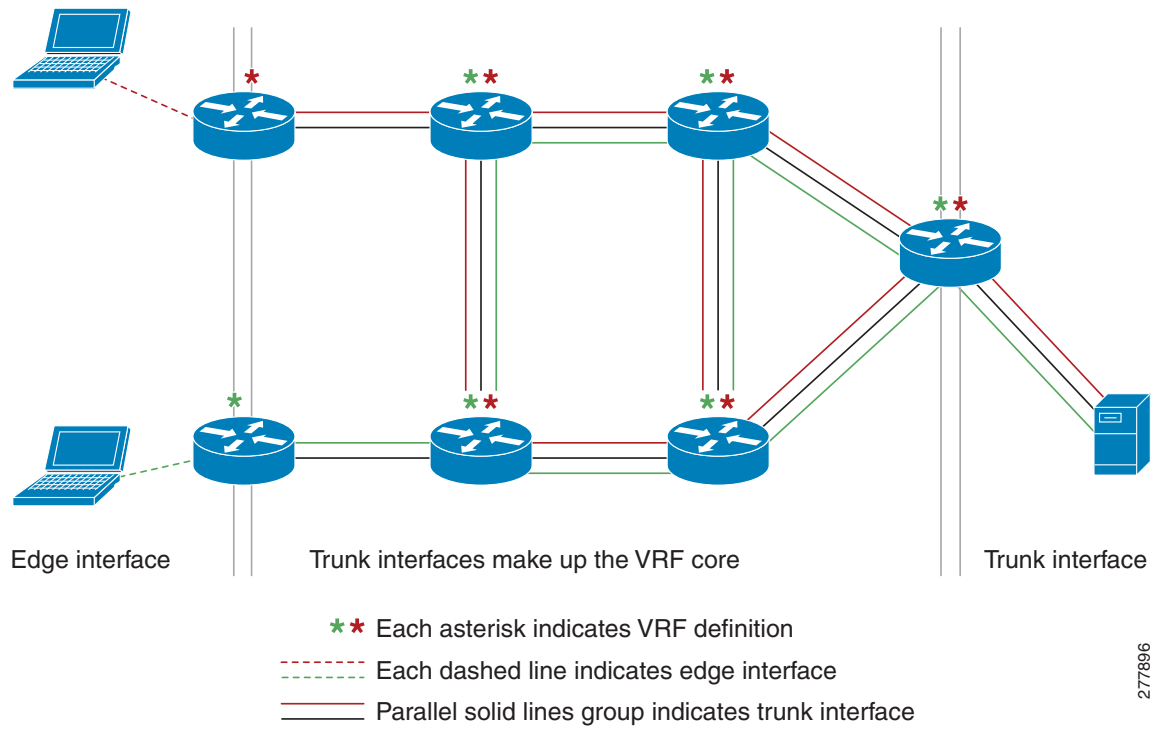
```
vrf list mylist
member red
member green
!
interface FastEthernet 1/0/0
vnet trunk list mylist
ip address 10.1.1.1 255.255.255.0
```

VRF Awareness

A device connected to a virtual network may not understand virtual network tags and can send and receive only untagged traffic. Such a device is referred to as VRF unaware. For example, a laptop computer is usually VRF unaware.

By contrast, a device that can send and receive tagged traffic and therefore takes the tag value into account when processing such traffic is known as VRF aware. For example, a VRF-aware server shared among different EVNs could use the virtual network tag to distinguish requests received and send responses. A VRF-aware device is connected to the EVN using a trunk interface, as shown in figure 4.

Figure 85-4 VRF Aware Server



277896

The term “VRF aware” can also be used to describe a software component running on the device. A software component is VRF aware if it can operate on different EVNs. For example, ping is VRF aware because it allows you to choose the EVN to which you want to send the ping packet.

Routing Protocols Supported by EVN

Each EVN runs a separate instance of a routing protocol. This allows each EVN to fine-tune its routing separately and also limits fate sharing. Different virtual networks may run different routing protocols concurrently.

EVN supports static routes, OSPFv2, and PIM, MSDP, and IGMP for multicast routing.

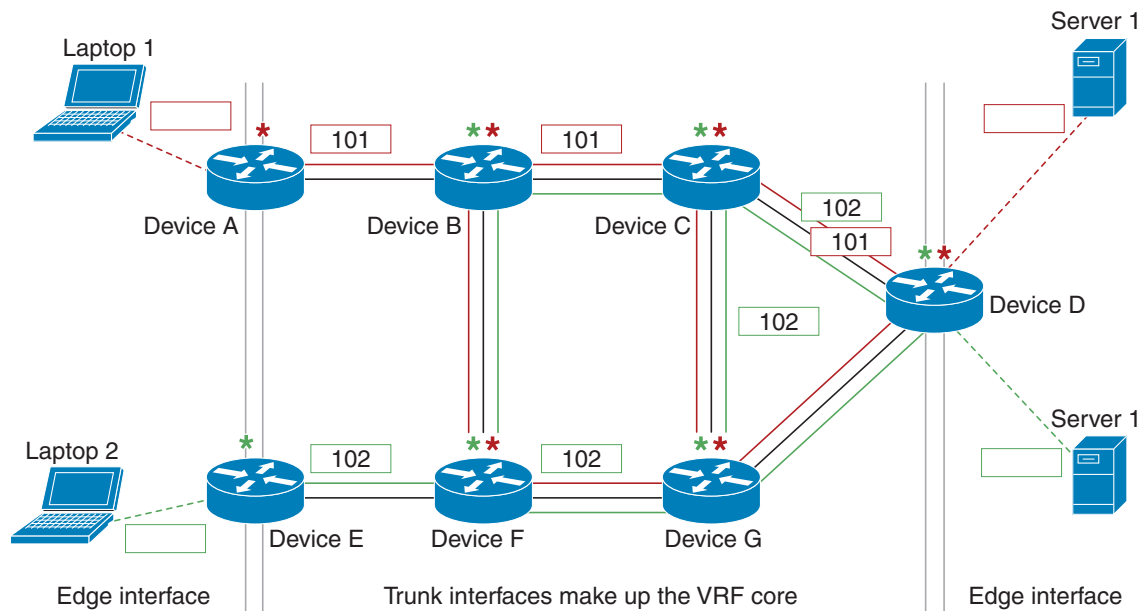
Packet Flow in a Virtual Network

Packets enter an EVN through an edge interface, traverse multiple trunk interfaces, and exit the virtual network through another edge interface. At the ingress edge interface, packets are mapped from a VLAN into a particular EVN. Once the packet is mapped to an EVN, it is tagged with the associated virtual network tag. The virtual network tag allows the trunk interface to carry packets for multiple EVNs. The packets remain tagged until they exit the EVN through the egress edge interface.

On the edge interface, the EVN associated with the interface is used for route lookup. On the trunk interface, the virtual network tag carried in the packet is used to locate the corresponding EVN for routing the packets.

If the egress interface is an edge interface, the packet is forwarded untagged. However, if the egress interface is a trunk interface, the packet is forwarded with the tag of the ingress EVN.

The figure below illustrates how traffic from two VRFs, red and green, can coexist on the same IP infrastructure, using the tags 101 and 102.

Figure 85-5 Packet Flow in a Virtual Network

* * Each asterisk indicates VRF definition

--- Each dashed line indicates edge interface

==== Parallel solid lines group indicates trunk interface

277897

The packet flow from Laptop 1 to Server 1 in VRF red occurs as follows:

1. Laptop 1 send an untagged packet to Server 1.
2. Device A receives the packet over an edge interface, which is associated with VRF red.
 - a. Device A does route lookup in VRF red and sees that the next hop is Device B through a trunk interface.
 - b. Device A encapsulates the packet with VRF red's tag (101) and sends it over the trunk interface.
3. Device B receives the packet over a trunk interface. Seeing virtual network tag 101, Device B identifies that the packet belongs to VRF red.
 - a. Device B does route lookup in VRF red and sees that the next hop is Device C through a trunk interface.
 - b. Device B encapsulates the packet with VRF red's tag (101) and sends it over the trunk interface.
4. Device C receives the packet over a trunk interface. Using virtual network tag 101, Device C identifies that the packet belongs to VRF red.
 - a. Device C does route lookup in VRF red and sees that the next hop is Device D through a trunk interface.
 - b. Device C encapsulates the packet with VRF red's tag (101) and sends it over the trunk interface.
5. Device D receives the packet over a trunk interface. Using virtual network tag 101, Device D identifies that the packet belongs to VRF red.
 - a. Device D does route lookup in VRF red and sees that the next hop is through an edge interface.
 - b. Device D sends the untagged packet over the edge interface to Server 1.
6. Server 1 receives the untagged packet originated from Laptop 1.

Command Inheritance on EVN Trunk Interfaces

One of the benefits of EVN is the ability to easily configure multiple EVNs on a common trunk interface without the need to configure each interface associated with an EVN individually. An EVN trunk interface takes advantage of the fact that the configuration requirements for different EVNs will be similar over a single trunk interface. When specific commands are configured on the trunk interface, they define default values that are inherited by all EVNs running over the same interface, including vnet global. If you feel that the settings are acceptable for all of the EVNs sharing an interface, then no individual configuration is necessary.

For example, the OSPF hello interval can be set for all EVNs over the trunk interface with the following configuration:

```
interface gigabitethernet1/1/1
 vnet trunk
 ip address 10.1.2.1 255.255.255.0
 ! set OSPF hello interval for all VRFs on this interface.
 ip ospf hello-interval 20
```

Overriding Command Inheritance Virtual Network Interface Mode

You can set up EVNs on the same trunk interface to have different configurations, by override inherited values using specific commands in virtual network interface mode for individual EVNs. In this mode, the command's settings override the Cisco default value or the value you set in interface configuration mode.

In interface configuration mode, entering the **vnet name** command causes the system to enter virtual network interface mode.

Beginning in Cisco IOS XE Release 3.9.1E, you can override the inherited IP address for subinterfaces. For more information, see [Changing the Inherited IP Address for Subinterfaces](#), page 85-17.

The following list displays the commands for which inherited values can be overridden:

Command	Values Inherited by EVNs on Interface?	Values Can Be Overridden in Virtual Network Interface Mode?
ip accounting	Yes	No
ip address	Yes	Yes
ip broadcast-address	Yes	No
ip directed broadcast	Yes	No
ip information-reply	Yes	No
ip irdp	Yes	No
ip load-sharing	Yes	No
ip mask-reply	Yes	No
ip mtu	Yes	No
ip proxy-arp	Yes	No
ip redirects	Yes	No

Command	Values Inherited by EVNs on Interface?	Values Can Be Overridden in Virtual Network Interface Mode?
ip unnumbered	Yes	No
ip unreachable	Yes	No
ip ospf process-id area	No	Yes
ip ospf authentication	Yes	Yes
ip ospf authentication-key	Yes	Yes
ip ospf cost	Yes	Yes
ip ospf database-filter	Yes	Yes
ip ospf dead-interval	Yes	Yes
ip ospf demand-circuit	Yes	Yes
ip ospf flood-reduction	Yes	Yes
ip ospf hello-interval	Yes	Yes
ip ospf ll	Yes	Yes
ip ospf message-digest-key	Yes	Yes
ip ospf mtu-ignore	Yes	Yes
ip ospf network	Yes	Yes
ip ospf priority	Yes	Yes
ip ospf resync-timeout	Yes	Yes
ip ospf shutdown	Yes	Yes
ip ospf transmit-delay	Yes	Yes
ip ospf transmit-interval	Yes	Yes
ip ospf ttl-security	Yes	Yes
ip ospf vnet area	No	No
ip igmp access-group	Yes	Yes
ip igmp explicit-tracking	Yes	Yes
ip igmp helper-address	Yes	Yes
ip igmp immediate-leave	Yes	Yes
ip igmp last-member-query-count	Yes	Yes
ip igmp last-member-query-interval	Yes	Yes
ip igmp limit	Yes	Yes
ip igmp mroute-proxy	Yes	Yes
ip igmp proxy-service	Yes	Yes
ip igmp querier-timeout	Yes	Yes
ip igmp query-interval	Yes	Yes
ip igmp query-max-response-time	Yes	Yes

Command	Values Inherited by EVNs on Interface?	Values Can Be Overridden in Virtual Network Interface Mode?
ip igmp tcn	Yes	Yes
ip igmp unidirectional-link	Yes	Yes
ip igmp v3lite	Yes	Yes
ip igmp version	Yes	Yes
ip multicast boundary	Yes	Yes
ip pim bidir-neighbor-filter	Yes	Yes
ip pim bsr-border	Yes	Yes
ip pim dense-mode	Yes	Yes
ip pim dr-priority	Yes	Yes
ip pim nbma-mode	Yes	Yes
ip pim neighbor-filter	Yes	Yes
ip pim passive	Yes	Yes
ip pim query-interval	Yes	Yes
ip pim sparse-dense-mode	Yes	Yes
ip pim sparse-mode	Yes	Yes
ip pim state-refresh	Yes	Yes
ip mfib cef	Yes	Yes
ip mfib forwarding	Yes	Yes

Removing Overrides and Restoring Values Inherited from EVN Trunk

The **no** and **default** keywords result in different outcomes, depending on whether they are used for a trunk interface or in virtual network interface mode. This section describes the different outcomes.

When the **no** or **default** keyword is entered before a command on a trunk interface, the trunk is restored to the system's default value for that command. (This is standard behavior resulting for the **no** or **default** keyword).

When the **default** keyword is entered before a command in virtual network interface mode, the override value is removed and the value that is inherited from the trunk is restored. The override value for the specific EVN is no longer in effect.

In the following example, the trunk interface is configured with an OSPF cost of 20, but VRF blue overrides that value with an OSPF cost of 30:

```
interface gigabitethernet 2/0/0
 vnet trunk
 ip address 10.1.1.1 255.255.255.0
 ! Set OSPF cost for all VRFs on this interface to 20.
 ip ospf cost 20
 vnet name blue
 ! Set OSPF cost for blue to 30.
 ip ospf cost 30
```

When the following commands are entered, the OSPF cost value is restored to 20, which is the cost inherited from the trunk interface. (Note that 20 is not the default value of the `ip ospf cost` command.)

```
Device(config-if)# vnet name blue
Device(config-if-vnet)# default ip ospf cost
```

Determining if No Form of Commands Appear in Configuration Files

If a command switches a feature on or off, the **no** form of the command appears in the configuration file when configured. Nonvolatile generation (NVGEN) overrides the setting from the EVN trunk, as shown in the following example:

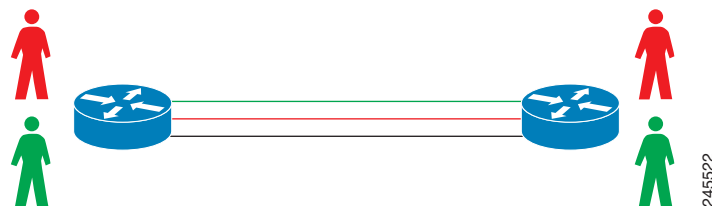
```
interface gigabitethernet 2/0/0
 vnet trunk
 vnet name red
  no ip pim sparse-mode
  no ip route-cache cef
 vnet global
  ip ospf cost 100
```

If a command takes an argument in its syntax, such as **ip ospf cost cost**, the **no** form of the command will remove the configuration, but does not appear in the configuration file. That is, it will not be NVGEN'ed because the user could enter **ip ospf cost default-value** to override the inherited value.

EVN Compatibility with VRF-Lite

EVN is wire compatible with VRF-Lite. In other words, on the outside, 802.1q, SNMP MIBs, and all the EVN infrastructure will look exactly the same as VRF-Lite.

In the figure below, both devices have VRFs defined. The device on the left uses VRF-Lite, and the device on the right uses an EVN trunk with tags. The two configurations follow the figure.



Example: VRF-Lite Subinterface Configuration EVN Trunk Configuration

```
interface TenGigabitEthernet1/1/1
 ip address 10.122.5.31 255.255.255.254
 ip pim query-interval 333 msec
 ip pim sparse-mode
 logging event link-status
interface TenGigabitEthernet1/1/1.101
 description Subinterface for Red VRF
```

```
interface TenGigabitEthernet 1/1/1
 vnet trunk
 ip address 10.122.5.32 255.255.255.254
 pim sparse-mode
 logging event link-status
Global Configuration:
 vrf definition red
```

```

encapsulation dot1Q 101
ip vrf forwarding Red
ip address 10.122.5.31 255.255.255.254
ip pim query-interval 333 msec
ip pim sparse-mode
logging event subif-link-status
interface TenGigabitEthernet1/1/1.102
description Subinterface for Green VRF
encapsulation dot1Q 102
ip vrf forwarding Green
ip address 10.122.5.31 255.255.255.254
ip pim query-interval 333 msec
ip pim sparse-mode
logging event subif-link-status

vnet tag 101
vrf definition green
vnet tag 102

```

SQoS and EVN

Quality of Service (QoS) configurations are applied to the main physical interface on an EVN trunk. The QoS policy affects all traffic that flows out the physical interface in all the VRFs at the same time. In other words, QoS and network virtualization are mutually independent. For example, traffic marked with the DSCP value specified for voice will be put into the voice queue if the packet is from the red VRF, blue VRF, or green VRF. The traffic for all the VRFs is queued together.

Configuring Easy Virtual Networks



Note

We recommend that you draw your network topology, indicating the interfaces on each router that belong to the EVNs. The diagram facilitates tracking the interfaces you are configuring as edge interfaces and the interfaces you are configuring as trunk interfaces.

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# vrf definition <i>vrf name</i>	Configures a VRF routing table instance and enters VRF configuration mode.
Step 3	Switch(config-vrf)# description <i>string</i>	(Optional) Describes a VRF to help a network administrator review the configuration files. to it.You can specify up to 3 control-plane IP addresses for the edge device.
Step 4	Switch(config-vrf)# address-family ipv4	Enters address family configuration mode to configure a routing session using standard IP version 4 address prefixes.
Step 5	Switch(config-vrf-af)# exit-address-family	Exits address family configuration mode.
Step 6	Switch(config-vrf)# exit	Exits to global configuration mode.
Step 7		Repeat steps 2 to 7 to configure another VRF instance and associate a vnet tag.
Step 8	Switch(config)# interface <i>interface number</i>	Configures an interface type and enters interface configuration mode.

	Command	Purpose
Step 9	Switch(config-if)# ip address <i>ip address mask</i>	Sets a primary IP address for the interface.
Step 10	Switch(config-if)# vrf trunk [<i>list vrf-list name</i>]	Defines a trunk interface. By default, all VRFs defined with the vrf definition command run on all trunk interfaces on the router. Therefore, VRF red and VRF blue are now running on this interface. Use the list vrf-list-name command elements to restrict VRFs running on a trunk interface.
Step 11	Switch(config-if)# no shutdown	Restarts an interface.
Step 12	Switch(config-if)# exit	Returns to global configuration mode.
Step 13	Switch(config)# router ospf <i>process ID</i>	Configures an Open Shortest Path First (OSPF) routing process and associates it with a VRF. This OSPF instance has no VRF, so it is vnet global .
Step 14	Switch(config-router)# network <i>ip address wildcard area area ID</i>	Defines the interfaces and associated area IDs on which OSPF runs, and the area ID for those interfaces.
Step 15	Switch(config-if)# exit	Returns to global configuration mode.
Step 16	Switch(config)# router ospf <i>process ID vrf vrf name</i>	Configures an OSPF routing process and associates it with a VRF. Specifies a different process-id for each VRF because they each need their own OSPF instance.
Step 17	Switch(config-router)# network <i>ip address wildcard area area ID</i>	Defines the interfaces and associated area IDs on which OSPF runs, and the area ID for those interfaces.
Step 18	Switch(config-router)# end	Ends the configuration session and returns to privileged EXEC mode.

Enabling a Subset of VRFs over a Trunk Interface

To create a VRF list and enable only a subset of VRFs over a trunk interface, enter the following commands:



Note

This task presumes that the VRF has already been configured.

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# vrf list <i>vrf list name</i>	Defines a list of VRFs and enters VRF list configuration mode. The vrf-list-name argument may contain up to 32 characters. Quotation marks, spaces, and * are not allowed.
Step 3	Switch(config-vrf-list)# member <i>vrf name</i>	Specifies an existing VRF as a member of a VRF list. The VRF must be defined before it can be added to a list.

	Command	Purpose
Step 4	Repeat Step 3 to add other VRFs to the list.	(Optional) If you want a trunk interface with one VRF, your list only needs one VRF.
Step 5	Switch(config)# interface <i>type number</i>	Configures an interface and enters interface configuration mode.
Step 6	Switch(config-if)# vnet trunk list <i>vrf list name</i>	Defines a trunk interface and enables the VRFs that are in the VRF list. Use the vrf list name you defined earlier in this task.
Step 7	Switch(config-if)# ip address <i>ip address mask</i>	Sets a primary IP address for the interface.
Step 8	Switch(config-if)# end	Ends the configuration session and returns to privileged EXEC mode.
Step 9	Switch# show vrf list [<i>vrf list name</i>]	Displays information about the specified VRF list.

Configuring EVN Edge Interfaces

Perform this task to configure an edge interface, which connects a user device to a virtual network. Traffic carried over an edge interface is untagged. The edge interface determines which virtual network the received traffic belongs to. Each edge interface is mapped to only one virtual network.

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface <i>type number</i>	Configures an interface type and enters interface configuration mode.
Step 3	Switch(config)# vrf forwarding <i>vrf name</i>	Defines an edge interface and determines the VRF to which the incoming traffic belongs. The vrf name must already be defined using the vrf definition command. Note Ensure that you are not on the trunk interface when you configure an edge interface.
Step 4	Switch(config-if)# ip address <i>ip address mask</i>	Sets a primary IP address for the interface.
Step 5	Switch(config-if)# end	Ends the configuration session and returns to privileged EXEC mode.

Verifying EVN Configuration

Enter the following commands to verify your configuration. All the existing VRF show commands are supported in virtual networks. If a device has a mix of VRFs and virtual networks, the various **show vrf** commands will include both VRFs and virtual networks in the output.

Command	Purpose
Switch# show vnet tag	Displays where each tag has been configured or used.
Switch# show running-config [<i>vrf vnet</i>] [<i>vrf-name</i>]	Displays the VRFs in the running configuration, displays the interfaces in the VRFs, and displays the protocol configurations for Multi-VRF.

Command	Purpose
Switch# show vrf list [vrf-list-name]	Displays information about VRF lists, such as the VRFs in each list.
Switch# show {vrf vnet}[ipv4 ipv6][interface brief detail lock] [vrf-name]	Displays information about the VRFs.
Switch# show {vrf vnet} counters	Displays information about the number of VRFs or virtual networks supported and configured.

Changing the Inherited IP Address for Subinterfaces

All subinterfaces created on the vnet interface inherit values from the main interface.

Beginning in Cisco IOS XE Release 3.9.1E, you can change the inherited IP address for subinterfaces in interface vnet configuration mode:

	Command	Purpose
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# interface interface name	Specifies the interface name and enters interface configuration mode.
Step 3	Switch(config-if-vnet)# (no) ip address ipv4 address mask	Sets the IP address for the VNET subinterface. The no ip address command on the vnet interface configuration mode will change the IP address of the subinterface back to match the IP address of the main interface.
Step 4	Switch(config-if-vnet)# ip address ipv4 address mask secondary	Sets the secondary IP address for the subinterface. A secondary IP from the main interface is not inherited if you set a secondary IP address for the subinterface.
Step 5	Switch(config-if-vnet)# do show interface ip brief	Displays the IP addresses for the main interface and the subinterfaces.

For example, consider the following configuration:

```
vrf definition vRED
vnet tag 131
!
address-family ipv4
exit-address-family

vrf definition vBLUE
vnet tag 132
!
address-family ipv4
exit-address-family

interface Eth0/0
no shutdown
vnet trunk
ip add 10.1.1.1 255.255.255.0

Switch(config-if)#int eth0/0
Switch(config-if)#vnet name vRED
```

```

Switch(config-if-vnet)# ip address 100.1.1.1 255.255.255.0
Switch(config-if-vnet)#do show ip interface brief
Interface                IP-Address      OK? Method Status      Protocol
Ethernet0/0              10.1.1.1        YES manual up          up
Ethernet0/0.131          100.1.1.1       YES manual up          up
Ethernet0/0.132          10.1.1.1       YES manual up          up

Switch(config)#int eth0/0
Switch(config-if)#vnet name vBLUE
Switch(config-if-vnet)#ip address 101.1.1.1 255.255.255.0
Switch(config-if-vnet)#do show ip interface brief
Interface                IP-Address      OK? Method Status      Protocol
Ethernet0/0              10.1.1.1        YES manual up          up
Ethernet0/0.131          100.1.1.1       YES manual up          up
Ethernet0/0.132          101.1.1.1       YES manual up          up

```

Configuration Examples for Configuring EVN

Example: Virtual Networks Using OSPF with network Commands

In this example, network commands associate a shared VRF interface with a base VRF and two named VRFs, red and blue. There are three OSPF instances because each VRF needs its own OSPF instance. OSPF 1 has no VRF, so it is vnet global.

```

vrf definition red
vnet tag 100
address-family ipv4
exit-address-family
!
vrf definition blue
vnet tag 200
address-family ipv4
exit-address-family
!
interface gigabitethernet 0/0/0
ip address 10.0.0.1 255.255.255.0
vnet trunk
vnet name red
ip ospf cost 100
!
router ospf 1
log-adjacency-changes detail
network 10.0.0.0 255.255.255.0 area 0
router ospf 2 vrf red
log-adjacency-changes
network 10.0.0.0 255.255.255.0 area 0
router ospf 3 vrf blue
log-adjacency-changes
network 10.0.0.0 255.255.255.0 area

```


Example: Virtual Networks Using OSPF with ip ospf vnet area Command

This example differs from the prior example regarding the association between OSPF instances and a particular interface. In this example, OSPF is running on all of the virtual networks of a trunk interface. The **ip ospf vnet area** command associates the GigabitEthernet 0/0/0 interface with the three OSPF instances.

```
vrf definition red
  vnet tag 100
  address-family ipv4
  exit-address-family
!
vrf definition blue
  vnet tag 200
  address-family ipv4
  exit-address-family
!
interface gigabitethernet 0/0/0
  ip address 10.0.0.1 255.255.255.0
  vnet trunk
  ip ospf vnet area 0
  vnet name red
    ip ospf cost 100
  vnet name blue
    ip ospf 3 area 2
!
router ospf 1
  log-adjacency-changes detail
router ospf 2 vrf red
  log-adjacency-changes
router ospf 3 vrf blue
  log-adjacency-changes
```

Example: Overriding Command Inheritance

In the following example, the OSPF cost of 30 for VRF blue overrides the OSPF cost of 20 for the other VRFs on the interface:

```
interface gigabitethernet 2/0/0
  vnet trunk
  ip address 10.1.1.1 255.255.255.0
  ! Set OSPF cost for all VRFs on this interface to 20.
  ip ospf cost 20
  vnet name blue
    description Subinterface for VRF NG blue
    ! Set OSPF cost for blue to 30.
    ip ospf cost 30
```

The show derived command indicates the subinterface changed to a cost of 30:

```
Device(config-if-vnet)# do show derived | s interface GigabitEthernet2/0/0

interface GigabitEthernet2/0/0
vnet trunk
ip address 10.1.1.1 255.255.255.0
ip ospf cost 20
interface GigabitEthernet2/0/0.200
description Subinterface for VRF NG blue
```

```
vrf forwarding blue
ip address 10.1.1.1 255.255.255.0
ip ospf cost 30
Device(config-if-vnet)#
```

Example: Enabling an Attribute to vnet Global Only

Similarly, you might want to enable an attribute to vnet global only. To do so, use the vnet global interface submode, as follows:

```
interface gigabitethernet1/1/1
vnet trunk
ip address 10.1.2.1 255.255.255.0
vnet global
! Set OSPF cost for global to 40.
ip ospf cost 40
```

Example: Command Inheritance and Virtual Network Interface Mode Override in a Multicast Environment

The following example illustrates command inheritance and virtual network interface mode override in a multicast network. A trunk interface leverages the fact that configuration requirements from different VRFs will be similar over the same trunk interface. Eligible commands configured on the trunk interface are inherited by all VRFs running over the same interface.

In this example, IP multicast (PIM sparse mode) is configured on the trunk interface, which has several VRFs:

```
vrf definition red
vnet tag 13
!
address-family ipv4
exit-address-family
!
ip multicast-routing
ip multicast-routing vrf red
interface GigabitEthernet0/1/0
vnet trunk
ip address 125.1.15.18 255.255.255.0
ip pim sparse-mode
```

The user decides that he does not want IP multicast configured for VRF red on GigabitEthernet 0/1/0, so he uses the virtual network interface mode override. IP Multicast is disabled for VRF red only. The `no ip pim` command disables all modes of Protocol Independent Multicast (PIM), including sparse mode, dense mode, and sparse-dense mode, for VRF red.

```
interface GigabitEthernet0/1/0
vnet trunk
ip address 125.1.15.18 255.255.255.0
ip pim sparse-mode
vnet name red
no ip pim
```

Example: EVN Using IP Multicast

The following example configures PIM sparse mode and leverages Anycast RP for RP redundancy. In this example, only one VRF is configured.

The example shows how to enable multicast routing globally and on each L3 interface. The black text indicates the group of commands configuring the global table; the red text indicates the group of commands configuring VRF red.

```

ip multicast-routing
interface GigabitEthernet 1/1/1
  description GigabitEthernet to core (Global)
  ip pim sparse-mode
vrf definition red
  vnet tag 100
!
  address-family ipv4
  exit-address-family
!
ip multicast-routing vrf red
!
interface gigabitethernet1/1/1.100
  description GigabitEthernet to core (VRF red)
  vrf forwarding red
  ip pim sparse-mode
Configure the RP in the VRF using Anycast RP.

interface loopback0
  description Anycast RP Global
  ip address 10.122.5.200 255.255.255.255
  ip pim sparse-mode
!
interface loopback1
  description MSDP Peering interface
  ip address 10.122.5.250 255.255.255.255
  ip pim sparse-mode
!
ip msdp peer 10.122.5.251 connect-source loopback 1
ip msdp originator-id loopback 1
ip pim rp-address 10.122.5.200
access-list 10 permit 239.0.0.0 0.255.255.255
!
!
interface loopback 10
  description Anycast RP VRF Red
  vrf forwarding red
  ip address 10.122.15.200 255.255.255.255
  ip pim sparse-mode
interface loopback 11
  description MSDP Peering interface VRF red
  vrf forwarding red
  ip address 10.122.15.250 255.255.255.255
  ip pim sparse-mode
!
ip msdp vrf red peer 10.122.15.251 connect-source loopback 11
ip msdp vrf red originator-id loopback 11
!
ip pim vrf red rp-address 10.122.15.200
access-list 11 permit 239.192.0.0 0.0.255.255

```

Troubleshooting EVN Configuration

Routing Context for EXEC Mode Reduces Repetitive VRF Specification

There may be occasions when you want to issue several EXEC commands to apply to a single virtual network. In order to reduce the repetitive entering of virtual routing and forwarding (VRF) names for multiple EXEC commands, the routing-context vrf command allows you to set the VRF context of such EXEC commands once, and then proceed using EXEC commands.

The table below shows four EXEC commands in Cisco IOS XE software without routing context and in routing context. Note that in the left column, each EXEC command must specify the VRF. In the right column, the VRF context is specified once and the prompt changes to reflect that VRF; there is no need to specify the VRF in each command.

EXEC Commands CLI without Routing Context	EXEC Commands CLI with Routing Context
	Device# routing-context vrf red Device%red#
Device# show ip route vrf red [Routing table output for VRF red]	Device%red# show ip route [Routing table output for VRF red]
Device# ping vrf red 10.1.1.1 [Ping result using VRF red]	Device%red# ping 10.1.1.1 [Ping result using VRF red]
Device# telnet vrf red 10.1.1.1 [Telnet to 10.1.1.1 in VRF red]	Device%red# telnet 10.1.1.1 [Telnet to 10.1.1.1 in VRF red]
Device# traceroute vrf red 10.1.1.1 [Traceroute output in VRF red]	Device%red# traceroute 10.1.1.1 [Traceroute output in VRF red]

traceroute Output Indicates VRF Name and VRF Tag

The output of the traceroute command is enhanced to make troubleshooting easier by displaying the incoming VRF name/tag and the outgoing VRF name/tag, as shown in the following example:

```
Device# traceroute vrf red 10.0.10.12
Type escape sequence to abort.
Tracing the route to 10.0.10.12
VRF info: (vrf in name/id, vrf out name/id)
 1 10.1.13.15 (red/13,red/13) 0 msec
   10.1.16.16 (red/13,red/13) 0 msec
   10.1.13.15 (red/13,red/13) 1 msec
 2 10.1.8.13 (red/13,red/13) 0 msec
   10.1.7.13 (red/13,red/13) 0 msec
   10.1.8.13 (red/13,red/13) 0 msec
 3 10.1.2.11 (red/13,blue/10) 1 msec 0 msec 0 msec
 4 * * *
```

Debug Output Filtering Per VRF

Using EVN, you can filter debug output per VRF by using the debug condition vrf command. The following is sample output from the debug condition vrf command:

```
Device# debug condition vrf red

Condition 1 set
CEF filter table debugging is on
CEF filter table debugging is on
D1#
*Aug 19 23:06:38.178: vrfmgr(0) Debug: Condition 1, vrf red triggered, count 1
```

CISCO-VRF-MIB

EVN provides a CISCO-VRF-MIB for VRF discovery and management.



ROM Monitor

he ROM monitor firmware runs when the router is powered up or reset. The firmware helps to initialize the processor hardware and boot the operating system software. You can use the ROM monitor to perform certain configuration tasks, such as recovering a lost password or downloading software over the console port. If there is no Cisco IOS software image loaded on the router, the ROM monitor runs the router.

This appendix contains the following sections:

- [Entering the ROM Monitor](#)
- [ROM Monitor Commands](#)
- [ROM Monitor Command Descriptions](#)
- [Configuration Register](#)
- [Console Download](#)
- [Debug Commands](#)
- [Exiting the ROM Monitor](#)



Note

For complete syntax and usage information for the switch commands used in this chapter, see the [Cisco IOS Command Reference Guides for the Catalyst 4500 Series Switch](#).

If a command is not in the *Cisco Catalyst 4500 Series Switch Command Reference* , you can locate it in the [Cisco IOS Master Command List, All Releases](#).

Entering the ROM Monitor

To use the ROM monitor, you must be using a terminal or PC that is connected to the router over the console port. Refer to the installation chapter in the *Cisco 806 Router Hardware Installation Guide* that came with the router to connect the router to a PC or terminal.

To configure the router to boot up in ROM monitor mode the next time it is rebooted, perform this task:

	Command	Task
Step 1	Switch# configure terminal	Enters global configuration mode.
Step 2	Switch(config)# config-reg 0x0	Resets the configuration register.

	Command	Task
Step 3	Switch(config)# exit	Exits global configuration mode.
Step 4	Switch# reload	<p>Reboots the router with the new configuration register value. The router remains in ROM monitor and does not boot the Cisco IOS software.</p> <p>As long as the configuration value is 0x0, you must manually boot the operating system from the console. See the boot command in the “ROM Monitor Command Descriptions” section in this appendix.</p> <p>After the router reboots, it is in ROM monitor mode. The number in the prompt increments with each new line.</p>

ROM Monitor Commands

Enter **?** or **help** at the ROM monitor prompt to display a list of available commands and options, as follows:

```
rommon 1 > ?
alias          set and display aliases command
boot          boot up an external process
confreg       configuration register utility
dev           list the device table
dir           list files in file system
help          monitor builtin command help
history       monitor command history
meminfo       main memory information
repeat        repeat a monitor command
reset         system reset
set           display the monitor variables
sysret        print out info from last system return
unalias       unset an alias
unset         unset a monitor variable
```

Commands are case sensitive. You can halt any command by pressing the Break key on a terminal. If you are using a PC, most terminal emulation programs halt a command when you press the Ctrl and the Break keys at the same time. If you are using another type of terminal emulator or terminal emulation software, refer to the documentation for that product for information on how to send a Break command.

ROM Monitor Command Descriptions

Table 86-1 describes the most commonly used ROM monitor commands.

Table 86-1 Most Commonly Used ROM Monitor Commands

Command	Description
reset or i	Resets and initializes the router, similar to a power up.
dev	Lists boot device identifications on the router; for example: <pre>rommon 10> dev Devices in device table: id name flash: flash</pre>
dir device:	Lists the files on the named device; flash, for example: <pre>rommon 4 > dir flash: File size Checksum File name 2835276 bytes (0x2b434c) 0x2073 c806-oy6-mz</pre>
boot commands	For more information about the ROM monitor boot commands, refer to the <i>Cisco IOS Configuration Guide</i> and the <i>Cisco IOS Command Reference</i> .
b	Boots the first image in flash memory.
b flash: [filename]	Attempts to boot the image directly from the first partition of flash memory. If you do not enter a filename, this command will boot this first image in flash.

Configuration Register

The virtual configuration register is in nonvolatile RAM (NVRAM) and has the same functionality as other Cisco routers. You can view or modify the virtual configuration register from either the ROM monitor or the operating system software. Within ROM monitor, you can change the configuration register by entering the register value in hexadecimal format, or by allowing the ROM monitor to prompt you for the setting of each bit.

Changing the Configuration Register Manually

To change the virtual configuration register from the ROM monitor manually, enter the **confreg** command followed by the new value of the register in hexadecimal, as shown in the following example:

```
rommon 1 > confreg 0x2101
```

You must reset or power cycle for new config to take effect
 rommon 2 >

The value is always interpreted as hexadecimal. The new virtual configuration register value is written into NVRAM but does not take effect until you reset or reboot the router.

Changing the Configuration Register Using Prompts

Entering **confreg** without an argument displays the contents of the virtual configuration register and a prompt to alter the contents by describing the meaning of each bit.

In either case, the new virtual configuration register value is written into NVRAM but does not take effect until you reset or reboot the router.

The following display shows an example of entering the **confreg** command:

```
rommon 7> confreg

Configuration Summary
enabled are:
console baud: 9600
boot: the ROM Monitor

do you wish to change the configuration? y/n [n]: y
enable "diagnostic mode"? y/n [n]: y
enable "use net in IP bcast address"? y/n [n]:
enable "load rom after netboot fails"? y/n [n]:
enable "use all zero broadcast"? y/n [n]:
enable "break/abort has effect"? y/n [n]:
enable "ignore system config info"? y/n [n]:
change console baud rate? y/n [n]: y
enter rate: 0 = 9600, 1 = 4800, 2 = 1200, 3 = 2400 [0]: 0
change the boot characteristics? y/n [n]: y
enter to boot:
0 = ROM Monitor
1 = the boot helper image
2-15 = boot system
[0]: 0

Configuration Summary
enabled are:
diagnostic mode
console baud: 9600
boot: the ROM Monitor

do you wish to change the configuration? y/n [n]:

You must reset or power cycle for new config to take effect
```

Console Download

You can use console download, a ROM monitor function, to download over the router console port either a software image or a configuration file. After download, the file is either saved to the mini-Flash memory module or to main memory for execution (image files only).

Use console download when you do not have access to a Trivial File Transfer Protocol (TFTP) server.



Note

If you want to download a software image or a configuration file to the router over the console port, you must use the **ROM monitor** command.

**Note**

If you are using a PC to download a Cisco IOS image over the router console port at 115,200 bps, ensure that the PC serial port is using a 16550 universal asynchronous transmitter/receiver (UART). If the PC serial port is not using a 16550 UART, we recommend using a speed of 38,400 or less when downloading a Cisco IOS image over the console port.

Error Reporting

Because the ROM monitor console download uses the console to perform the data transfer, error messages are only displayed on the console when the data transfer is terminated.

If an error does occur during a data transfer, the transfer is terminated, and an error message is displayed. If you have changed the baud rate from the default rate, the error message is followed by a message telling you to restore the terminal to the baud rate specified in the configuration register.

With ROMMON version 15.1(1r)SG4 and 15.1(1r)SG5 on Supervisor Engine 8-E, the follow error message is displayed if the supervisor is idle for more than an hour. You can ignore this message; it does not affect device performance.

```
rommon 0 >ICMP: Unsupported type/opcode! d00
ICMP: Unsupported type/opcode! d00
ICMP: Unsupported type/opcode! d00
ICMP: Unsupported type/opcode! d00
ICMP: Unsupported type/opcode! d00
ICMP: Unsupported type/opcode! d00
ICMP: Unsupported type/opcode! d00
ICMP: Unsupported type/opcode! d00

Pinging 10.64.71.1
!!!!
10.64.71.1 is alive!

rommon 2 >version
```

Debug Commands

Most ROM monitor debugging commands are functional only when Cisco IOS software has crashed or is halted.

The following are ROM monitor debugging commands:

- **frame**—Displays an individual stack frame.
- **sysret**—Displays return information from the last booted system image.

This information includes the reason for terminating the image, a stack dump of up to eight frames, and, if an exception is involved, the address where the exception occurred.

For example:

```
rommon 8> sysret
System Return Info:
count: 19, reason: user break
pc:0x801111b0, error address: 0x801111b0
Stack Trace:
FP: 0x80005ea8, PC: 0x801111b0
FP: 0x80005eb4, PC: 0x80113694
```

```

FP: 0x80005f74, PC: 0x8010eb44
FP: 0x80005f9c, PC: 0x80008118
FP: 0x80005fac, PC: 0x80008064
FP: 0x80005fc4, PC: 0xffff03d70
FP: 0x80005ffc, PC: 0x00000000
FP: 0x00000000, PC: 0x00000000

```

- **meminfo**—Displays size in bytes, starting address, available range of main memory, the starting point and size of packet memory, and size of NVRAM.

For example:

```
rommon 9> meminfo
```

```

Main memory size: 40 MB.
Available main memory starts at 0x10000, size 40896KB
IO (packet) memory size: 5 percent of main memory.
NVRAM size: 32KB

```

Exiting the ROM Monitor

You must set the configuration register to a value from 0x2 to 0xF for the router to boot a Cisco IOS image from flash memory upon startup or reloading.

The following example shows how to reset the configuration register and cause the router to boot a Cisco IOS image stored in flash memory:

```
rommon 1 > confreg 0x2101
```

You must reset or power cycle for new config to take effect

```
rommon 2 >boot
```

The router will boot the Cisco IOS image in flash memory. The configuration register will change to 0x2101 the next time the router is reset or power cycled.



Zero-Touch Provisioning

To address network provisioning challenges, Cisco introduces a zero-touch provisioning model. Starting with Cisco IOS XE Release 3.10.1E, the feature is available on Cisco Catalyst 4500-X Series Switches and Cisco Catalyst 4500E Series Switches with Supervisor Engine 9-E and 8-E.



Note

The Zero-Touch Provisioning feature is enabled automatically; no configuration is required.

This chapter includes the following major sections:

- [Information About Zero-Touch Provisioning, page 87-1](#)
- [DHCP Server Configuration for Zero-Touch Provisioning, page 87-2](#)
- [Sample Zero-Touch Provisioning Configurations, page 87-2](#)
- [Zero-Touch Provisioning Boot Log, page 87-2](#)

Information About Zero-Touch Provisioning

Zero-Touch Provisioning Overview

To address network provisioning challenges, Cisco introduces a Zero-Touch Provisioning model. Zero-Touch Provisioning automates the process of installing configuration files on Cisco devices that are deployed in a network for the first time. It reduces manual tasks required to scale the network capacity.

When a device that supports Zero-Touch Provisioning boots up, and does not find the startup configuration (during fresh install on Day Zero), the device enters the Zero-Touch Provisioning mode. The device locates a Dynamic Host Control Protocol (DHCP) server, bootstraps itself with its interface IP address, gateway, and Domain Name System (DNS) server IP address, and enables Guest Shell. The device then obtains the IP address or URL of a TFTP server, and downloads the configuration file for the device.



Note

In case Zero-Touch Provisioning fails, the device falls back to AutoInstall to load configuration files. For more information, see [Using AutoInstall and Setup](#).

DHCP Server Configuration for Zero-Touch Provisioning

In Zero-Touch Provisioning, a DHCP server must be running on the same network as the new device that is being provisioned. Zero-Touch Provisioning is supported on both management ports and in-band ports.

When the new device is switched on, it retrieves the IP address information of the TFTP server where the configuration resides.

The DHCP server responds to DHCP discovery events with the following options:

- Option 150—(Optional) Contains a list of IP addresses that point to the TFTP server on the management network that hosts the configuration file for the new device.

After receiving these DHCP options, the device connects to the TFTP server, and downloads the configuration. The device, at this point does not have any route to reach the TFTP server, so it uses the default route provided by the DHCP server.

Sample Zero-Touch Provisioning Configurations

Sample DHCP Server Configuration on a Management Port

The following is a sample DHCP server configuration when connected via the management port on a device:

```
Device> enable
Device# configure terminal
Device(config)# ip dhcp excluded-address 10.1.1.1
Device(config)# ip dhcp excluded-address vrf Mgmt-vrf 10.1.1.1 10.1.1.10
Device(config)# ip dhcp pool pnp_device_pool
Device(config-dhcp)# vrf Mgmt-vrf
Device(config-dhcp)# network 10.1.1.0 255.255.255.0
Device(config-dhcp)# default-router 10.1.1.1
Device(config-dhcp)# option 150 ip 203.0.113.254
Device(config-dhcp)# option 67 ascii switch.cfg
Device(config-dhcp)# end
```

- Option 67 ascii switch.cfg points to the configuration file that needs to be used by the new device.

Once the DHCP server is running, boot a management-network connected device, and the rest of the configuration is automatic.

Zero-Touch Provisioning Boot Log

The following sample Zero-Touch Provisioning boot log displays that Guest Shell is successfully enabled, the Python script is downloaded to the Guest Shell, and the Guest Shell executes the downloaded Python script and configures the device for Day Zero.

```
% failed to initialize nvram

! <This message indicates that the startup configuration
is absent on the device. This is the first indication that the Day Zero work flow is going
to start.>
```

This product contains cryptographic features and is subject to United States and local country laws governing import, export, transfer and use. Delivery of Cisco cryptographic products does not imply third-party authority to import, export, distribute or use encryption. Importers, exporters, distributors and users are responsible for compliance with U.S. and local country laws. By using this product you agree to comply with applicable laws and regulations. If you are unable to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:

<http://www.cisco.com/wwl/export/crypto/tool/stqrg.html>

If you require further assistance please contact us by sending email to export@cisco.com.

cisco ISR4451-X/K9 (2RU) processor with 7941237K/6147K bytes of memory.

Processor board ID FJC1950D091
4 Gigabit Ethernet interfaces
32768K bytes of non-volatile configuration memory.
16777216K bytes of physical memory.
7341807K bytes of flash memory at bootflash:.
0K bytes of WebUI ODM Files at webui:.

%INIT: waited 0 seconds for NVRAM to be available

--- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: % !!<DO NOT TOUCH.
This is Zero-Touch Provisioning>> Generating 2048 bit RSA keys, keys will be
non-exportable... [OK] (elapsed time was 1 seconds)
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
The process for the command is not responding or is otherwise unavailable
Guestshell enabled successfully

The Day Zero provisioning is complete, and the IOS prompt is accessible.



Acronyms and Abbreviations

Table 88-1 defines the acronyms and abbreviations used in this publication.

Table 88-1 **Acronyms**

Acronym	Expansion
ACE	access control entry
ACL	access control list
AFI	authority and format identifier
Agport	aggregation port
ALPS	Airline Protocol Support
AMP	Active Monitor Present
APaRT	Automated Packet Recognition and Translation
ARP	Address Resolution Protocol
AV	attribute value
AVVID	Architecture for Voice, Video and Integrated Data
BDD	binary decision diagrams
BECN	backward explicit congestion notification
BGP	Border Gateway Protocol
BPDU	bridge protocol data unit
BRF	bridge relay function
BSC	Bisync
BSTUN	Block Serial Tunnel
BUS	broadcast and unknown server
BVI	bridge-group virtual interface
CAM	content-addressable memory
CAR	committed access rate
CCA	circuit card assembly
CDP	Cisco Discovery Protocol
CEF	Cisco Express Forwarding
CGMP	Cisco Group Management Protocol

Table 88-1 **Acronyms (continued)**

Acronym	Expansion
CHAP	Challenge Handshake Authentication Protocol
CIR	committed information rate
CIST	Common and Internal Spanning Tree
CLI	command-line interface
CLNS	Connection-Less Network Service
CMNS	Connection-Mode Network Service
COPS	Common Open Policy Server
COPS-DS	Common Open Policy Server Differentiated Services
CoS	class of service
CPLD	Complex Programmable Logic Device
CRC	cyclic redundancy check
CRF	concentrator relay function
CST	Common Spanning Tree
CUDD	University of Colorado Decision Diagram
DBL	Dynamic Buffer Limiting
DCC	Data Country Code
dCEF	distributed Cisco Express Forwarding
DDR	dial-on-demand routing
DE	discard eligibility
DEC	Digital Equipment Corporation
DFI	Domain-Specific Part Format Identifier
DFP	Dynamic Feedback Protocol
DISL	Dynamic Inter-Switch Link
DLC	Data Link Control
DLSw	Data Link Switching
DMP	data movement processor
DNS	Domain Name System
DoD	Department of Defense
DOS	denial of service
DRAM	dynamic RAM
DSAP	destination service access point
DSCP	differentiated services code point
DSPU	downstream SNA Physical Units
DTP	Dynamic Trunking Protocol
DTR	data terminal ready
DXI	data exchange interface

Table 88-1 **Acronyms (continued)**

Acronym	Expansion
EAP	Extensible Authentication Protocol
EARL	Enhanced Address Recognition Logic
EEPROM	electrically erasable programmable read-only memory
EHSA	enhanced high system availability
EHT	Explicit Host Tracking
EIA	Electronic Industries Association
ELAN	Emulated Local Area Network
EOBC	Ethernet out-of-band channel
ESI	end-system identifier
FECN	forward explicit congestion notification
FM	feature manager
FRU	field replaceable unit
FSM	feasible successor metrics
GARP	General Attribute Registration Protocol
GMRP	GARP Multicast Registration Protocol
GVRP	GARP VLAN Registration Protocol
HSRP	Hot Standby Routing Protocol
ICC	Inter-card Communication
ICD	International Code Designator
ICMP	Internet Control Message Protocol
IDB	interface descriptor block
IDP	initial domain part or Internet Datagram Protocol
IFS	IOS File System
IGMP	Internet Group Management Protocol
IGRP	Interior Gateway Routing Protocol
ILMI	Integrated Local Management Interface
IP	Internet Protocol
IPC	interprocessor communication
IPX	Internetwork Packet Exchange
IS-IS	Intermediate System-to-Intermediate System Intradomain Routing Protocol
ISL	Inter-Switch Link
ISO	International Organization of Standardization
LAN	local area network
LANE	LAN Emulation
LAPB	Link Access Procedure, Balanced

Table 88-1 **Acronyms (continued)**

Acronym	Expansion
LDA	Local Director Acceleration
LCP	Link Control Protocol
LEC	LAN Emulation Client
LECS	LAN Emulation Configuration Server
LEM	link error monitor
LER	link error rate
LES	LAN Emulation Server
LLC	Logical Link Control
LTL	Local Target Logic
MAC	Media Access Control
MACL	MAC Access Control
MD5	Message Digest 5
MFD	multicast fast drop
MIB	Management Information Base
MII	media-independent interface
MLS	Multilayer Switching
MLSE	maintenance loop signaling entity
MOP	Maintenance Operation Protocol
MOTD	message-of-the-day
MLSE	maintenance loops signaling entity
MRM	multicast routing monitor
MSDP	Multicast Source Discovery Protocol
MST	Multiple Spanning Tree
MSTI	MST instance
MTU	maximum transmission unit
MVAP	multiple VLAN access port
NBP	Name Binding Protocol
NCIA	Native Client Interface Architecture
NDE	NetFlow Data Export
NET	network entity title
NetBIOS	Network Basic Input/Output System
NFFC	NetFlow Feature Card
NMP	Network Management Processor
NSAP	network service access point
NTP	Network Time Protocol
NVRAM	nonvolatile RAM

Table 88-1 **Acronyms (continued)**

Acronym	Expansion
OAM	Operation, Administration, and Maintenance
ODM	order dependent merge
OSI	Open System Interconnection
OSPF	open shortest path first
PACL	Port Access Control List
PAE	port access entity
PAgP	Port Aggregation Protocol
PBD	packet buffer daughterboard
PBR	Policy Based Routing
PC	Personal Computer
PCM	pulse code modulation
PCR	peak cell rate
PDP	policy decision point
PDU	protocol data unit
PEP	policy enforcement point
PGM	Pragmatic General Multicast
PHY	physical sublayer
PIB	policy information base
PIM	Protocol Independent Multicast
PoE	Power over Internet
PPP	Point-to-Point Protocol
PRID	Policy Rule Identifiers
PVST+	per-VLAN Spanning Tree+
QM	QoS manager
QoS	quality of service
RADIUS	Remote Access Dial-In User Service
RAM	random-access memory
RCP	Remote Copy Protocol
RGMP	Router-Ports Group Management Protocol
RIB	routing information base
RIF	Routing Information Field
RMON	Remote Network Monitor
ROM	read-only memory
ROMMON	ROM monitor
RP	route processor or rendezvous point
RPC	remote procedure call

Table 88-1 **Acronyms (continued)**

Acronym	Expansion
RPF	reverse path forwarding
RPR	Route Processor Redundancy
RSPAN	remote SPAN
RST	reset
RSVP	ReSerVation Protocol
SAID	Security Association Identifier
SAP	service access point
SCM	service connection manager
SCP	Switch-Module Configuration Protocol
SDLC	Synchronous Data Link Control
SGBP	Stack Group Bidding Protocol
SIMM	single in-line memory module
SLB	server load balancing
SLCP	Supervisor Line-Card Processor
SLIP	Serial Line Internet Protocol
SMDS	Software Management and Delivery Systems
SMF	software MAC filter
SMP	Standby Monitor Present
SMRP	Simple Multicast Routing Protocol
SMT	Station Management
SNAP	Subnetwork Access Protocol
SNMP	Simple Network Management Protocol
SPAN	Switched Port Analyzer
SSTP	Cisco Shared Spanning Tree
STP	Spanning Tree Protocol
SVC	switched virtual circuit
SVI	switched virtual interface
TACACS+	Terminal Access Controller Access Control System Plus
TARP	Target Identifier Address Resolution Protocol
TCAM	Ternary Content Addressable Memory
TCL	table contention level
TCP/IP	Transmission Control Protocol/Internet Protocol
TFTP	Trivial File Transfer Protocol
TIA	Telecommunications Industry Association
TopN	Utility that allows you to analyze port traffic by reports
TOS	type of service

Table 88-1 **Acronyms (continued)**

Acronym	Expansion
TLV	type-length-value
TTL	Time To Live
TVX	valid transmission
UDLD	UniDirectional Link Detection Protocol
UDP	User Datagram Protocol
UNI	User-Network Interface
UTC	Coordinated Universal Time
VACL	VLAN access control list
VCC	virtual channel circuit
VCI	virtual circuit identifier
VCR	Virtual Configuration Register
VINES	Virtual Network System
VLAN	virtual LAN
VMPS	VLAN Membership Policy Server
VPN	virtual private network
VRF	VPN routing and forwarding
VTP	VLAN Trunking Protocol
VVID	voice VLAN ID
WFQ	weighted fair queueing
WRED	weighted random early detection
WRR	weighted round-robin
XNS	Xerox Network System

