Cisco UCS Manager Network Management Guide Using the CLI, Release 6.0

First Published: 2025-09-02

Last Modified: 2026-03-16

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

© 2025–2026 Cisco Systems, Inc. All rights reserved.

CONTENTS

PREFACE

Preface xv

Audience xv

Conventions xv

Related Cisco UCS Documentation xvii

Documentation Feedback xvii

CHAPTER 1

New and Changed Information 1

New and Changed Information 1

CHAPTER 2

Overview 5

Overview 5

Cisco UCS Manager User CLI Documentation 5

CHAPTER 3

LAN Connectivity 7

Fabric Interconnect Overview 7

IOMs and Fabric Interconnects Connectivity 7

Uplink Connectivity 8

Downlink Connectivity 8

Configuring the Fabric Interconnects 9

Fabric Interconnect Information Policy 9

Installing Secure FPGA 9

Enabling the Information Policy on the Fabric Interconnect 10

Disabling the Information Policy on the Fabric Interconnect 10

Viewing the LAN Neighbors of the Fabric Interconnect 11

Viewing the SAN Neighbors of the Fabric Interconnect 12

Viewing the LLDP Neighbors of the Fabric Interconnect 12

Installing Secure FPGA 13
Fabric Evacuation 14
Stopping Traffic on a Fabric Interconnect 14
Displaying the Status of Evacuation for a Fabric Interconnect 15
Displaying the Status of Evacuation for an IOM 16
Verifying Fabric Evacuation 17
Restarting Traffic on a Fabric Interconnect 18
Fabric Interconnect Port Types 19
Fabric Interconnect Switching Modes 19
Ethernet Switching Mode 20
Configuring Ethernet Switching Mode 21
Fibre Channel Switching Mode 22
Configuring Fibre Channel Switching Mode 22

CHAPTER 4 LAN Ports and Port Channels 25
Unified Ports on the Cisco UCS Fabric Interconnects 25
Port Functionality on Cisco UCS 6664 Fabric Interconnect 25
Port Breakout Functionality on Cisco UCS 6652 Fabric Interconnect 26
Unified Ports on Cisco UCS 6600 Series Fabric Interconnects 28
Configuring Ethernet Ports on UCS 6600 Series Fabric Interconnects 29
Configuring Fibre Channel Ports on UCS 6600 Series Fabric Interconnects 30
Unified Ports on the Cisco UCS 6500 Series Fabric Interconnects 32
Configuring Ethernet Breakout Ports on UCS 6536 Fabric Interconnects 32
Configuring Fibre Channel Breakout Ports 33
Converting Ethernet Ports to Fibre Channel Breakout Port 33
Converting Ethernet Breakout Port to Fibre Channel Breakout Port 34
Converting Ethernet Breakout Port to Fibre Channel Breakout Port Using Fibre Channel Uplink 35
Deleting Fibre Channel Breakout Port 36
Deleting Fibre Channel Breakout Ports 36
Deleting Fibre Channel Breakout Ports Using Fibre Channel Uplink 36
Appliance Breakout Port 37
Configuring Breakout Appliance Ports 37
Modifying Speed for Breakout Port of Type 25x4Gbps 38
Modifying FEC Value for Breakout Port of Type 25 x 4Gbps 39

Unified Breakout Storage Ports 39
Converting Fibre Channel Uplink Port to Fibre Channel Storage Port 39
Fibre Channel Uplink Breakout Port Channels 41
Configuring Fibre Channel Uplink Breakout Port Channel and Member Addition 41
Unified Breakout Ports for Cisco UCS X-Series Direct 42
Configuring Ethernet Breakout Ports on Cisco UCS Fabric Interconnects 9108 100G 42
Physical and Backplane Ports 44
Displaying VIF Port Statistics Obtained From the Adaptor 44
Verifying Status of Backplane Ports 44
Verifying VFC to Vethernet Binding and Real-Time Statistics 47
Server Ports 49
Automatic Configuration of Fabric Interconnect Server Ports 49
Automatically Configuring Server Ports 49
Configuring a Server Port 50
Unconfiguring a Server Port 51
Configuring a Server Port for Forward Error Correction 51
Uplink Ethernet Ports 53
Configuring an Uplink Ethernet Port 53
Unconfiguring an Uplink Ethernet Port 54
Configuring an Uplink Ethernet Port for Forward Error Correction 54
Q-in-Q Forwarding 55
Configuring Q-in-Q Forwarding 55
Unconfiguring Q-in-Q Forwarding 56
STP Faults 57
Configuring STP Faults 57
Unconfiguring STP Faults 58
Appliance Ports 58
Configuring an Appliance Port 59
Assigning a Target MAC Address to an Appliance Port or Appliance Port Channel 60
Creating an Appliance Port 62
Mapping an Appliance Port to a Community VLAN 62
Unconfiguring an Appliance Port 63
Configuring an Appliance Port for Forward Error Correction 64
FCoE Uplink Ports 65

Configuring a FCoE Uplink Port 65
Unconfiguring a FCoE Uplink Port 66
Viewing FCoE Uplink Ports 67
Configuring FCoE Uplink for Forward Error Correction 67
Unified Storage Ports 69
Configuring a Unified Storage Port 69
Unified Uplink Ports 70
Configuring a Unified Uplink Port 70
FCoE and Fibre Channel Storage Ports 71
Configuring a Fibre Channel Storage or FCoE Port 71
Unconfiguring a Fibre Channel Storage or FCoE Port 72
Restoring a Fibre Channel Storage Port Back to an Uplink Fibre Channel Port 72
Uplink Ethernet Port Channels 73
Configuring an Uplink Ethernet Port Channel 73
Unconfiguring an Uplink Ethernet Port Channel 74
Adding a Member Port to an Uplink Ethernet Port Channel 75
Deleting a Member Port from an Uplink Ethernet Port Channel 75
Appliance Port Channels 76
Configuring an Appliance Port Channel 76
Unconfiguring an Appliance Port Channel 78
Enabling or Disabling an Appliance Port Channel 78
Adding a Member Port to an Appliance Port Channel 79
Deleting a Member Port from an Appliance Port Channel 80
Fibre Channel Port Channels 80
Configuring a Fibre Channel Port Channel 81
Configuring a FCoE Port Channel 82
Adding Channel Mode Active To The Upstream NPIV Fibre Channel Port Channel 82
Enabling or Disabling a Fibre Channel Port Channel 84
Adding a Member Port to a Fibre Channel Port Channel 84
Deleting a Member Port from a Fibre Channel Port Channel 85
FCoE Port Channels 85
Configuring a FCoE Port Channel 86
Adding a Member Port to a FCoE Uplink Port Channel 86
Unified Uplink Port Channel 87

Configuring a Unified Uplink Port Channel 88
Event Detection and Action 88
Policy-Based Port Error Handling 89
Creating Threshold Definition 89
Configuring Error Disable on a Fabric Interconnect Port 90
Configuring Auto Recovery on a Fabric Interconnect Port 91
Viewing the Network Interface Port Error Counters 92
Adapter Port Channels 93
Viewing Adapter Port Channels 93
Fabric Port Channels 94
Load Balancing Over Ports 94
Cabling Considerations for Fabric Port Channels 95
Configuring a Fabric Port Channel 95
Viewing Fabric Port Channels 96
Enabling or Disabling a Fabric Port Channel Member Port 96

CHAPTER 5 VLANs 99
VLANs 99
About the Native VLAN 100
Named VLANs 100
Private VLANs 101
VLAN Port Limitations 102
Configuring Named VLANs 104
Creating a Named VLAN Accessible to Both Fabric Interconnects (Uplink Ethernet Mode) 104
Creating a Named VLAN Accessible to Both Fabric Interconnects (Ethernet Storage Mode) 105
Creating a Named VLAN Accessible to One Fabric Interconnect (Uplink Ethernet Mode) 106
Creating a Secondary VLAN for a Private VLAN (Accessible to One Fabric Interconnect) 107
Deleting a Named VLAN 108
Configuring Private VLANs 110
Creating a Primary VLAN for a Private VLAN (Accessible to Both Fabric Interconnects) 110
Creating a Primary VLAN for a Private VLAN (Accessible to One Fabric Interconnect) 111
Creating a Secondary VLAN for a Private VLAN (Accessible to Both Fabric Interconnects) 112
Creating a Secondary VLAN for a Private VLAN (Accessible to One Fabric Interconnect) 113
Allowing PVLANs on vNICs 114

Creating a Primary VLAN for a Private VLAN on an Appliance Cloud 115
Creating a Secondary VLAN for a Private VLAN on an Appliance Cloud 116
Community VLANs 117
Creating a Community VLAN 117
Viewing Community VLANS 118
Allowing Community VLANs on vNICs 118
Allowing PVLAN on Promiscuous Access or Trunk Port 119
Deleting a Community VLAN 120
Viewing the VLAN Port Count 121
VLAN Port Count Optimization 121
Enabling Port VLAN Count Optimization 122
Disabling Port VLAN Count Optimization 123
Viewing the Port VLAN Count Optimization Groups 124
VLAN Groups 124
Creating a VLAN Group 125
Creating an Inband VLAN Group 125
Viewing VLAN Groups 126
Deleting a VLAN Group 127
Modifying the Reserved VLAN 127
VLAN Permissions 128
Creating VLAN Permissions 129
Viewing VLAN Permissions 129
Deleting a VLAN Permission 130
Fabric Port-Channel vHBA 130
Enabling Fabric Port Channel vHBA reset 131
Disabling fabric port channel vHBA reset 131
Viewing the Fabric Port Channel vHBA Reset 132
VIC QinQ Tunneling 132
Enabling and Managing QinQ 132
Enabling QinQ on a vNIC of a Service Profile 132
Disabling QinQ on a vNIC of a Service Profile 133
Enabling QinQ on a vNIC of LAN Connectivity Policy 134
Disabling QinQ on a vNIC of LAN Connectivity Policy 135
Enabling QinQ on a vNIC Template 136

Disabling QinQ on a vNIC Template 137
Viewing QinQ 138
VIC QinQ Tunneling - Supported Combinations and Limitations 139
Managing VLANs 140
Adding a VLAN on a vNIC Template 140
Adding a VLAN on a vNIC of LAN Connectivity Policy 141
Adding a VLAN on a vNIC of a Service Profile 142
Deleting a VLAN in a VNIC template 143
Deleting a VLAN on a vNIC of LAN Connectivity Policy 144
Deleting a VLAN on a vNIC of a Service Profile 145

CHAPTER 6 **LAN PIN Groups** 147
LAN Pin Groups 147
Configuring a LAN Pin Group 147

CHAPTER 7 **MAC Pools** 149
MAC Pools 149
Creating a MAC Pool 149
Deleting a MAC Pool 151

CHAPTER 8 **Quality of Service** 153
Quality of Service 153
Configuring System Classes 154
System Classes 154
Configuring a System Class 155
Disabling a System Class 157
Configuring Quality of Service Policies 158
Quality of Service Policy 158
Configuring a QoS Policy 158
Deleting a QoS Policy 160
Configuring Flow Control Policies 160
Flow Control Policy 160
Configuring a Flow Control Policy 161
Deleting a Flow Control Policy 162

Configuring Slow Drain 163
QoS Slow Drain Device Detection and Mitigation 163
Configuring Slow Drain Detection 164
Configuring Slow Drain Timers 164
Displaying Slow Drain Settings 165
Priority Flow Control Watchdog Interval 166
Configuring a Priority Flow Control Watchdog Interval 166
Viewing the Watchdog Settings 167

CHAPTER 9 **Configuring Port Security** 169
Port Security Overview 169
Port Security Violations 170
Guidelines for Port Security on UCS 6454 Fabric Interconnects 170
Configuring Port Security 171

CHAPTER 10 **Upstream Disjoint Layer-2 Networks** 173
Upstream Disjoint Layer-2 Networks 173
Guidelines for Configuring Upstream Disjoint L2 Networks 174
Upstream Disjoint L2 Networks Pinning Considerations 176
Configuring Cisco UCS for Upstream Disjoint L2 Networks 177
Assigning Ports and Port Channels to VLANs 178
Removing Ports and Port Channels from VLANs 179
Viewing Ports and Port Channels Assigned to VLANs 180

CHAPTER 11 **Network-Related Policies** 181
vNIC Template 181
Creating vNIC Template Pairs 182
Undo vNIC Template Pairs 184
Configuring a vNIC Template 185
Deleting a vNIC Template 187
Ethernet Adapter Policies 188
Configuring an Ethernet Adapter Policy 188
Deleting an Ethernet Adapter Policy 190
Receive Side Scaling (RSS) 190

Configuring an Ethernet Adapter Policy to Enable RSS on Windows Operating Systems 190
Configuring an Ethernet Adapter Policy to Support RSS and Multiple Transmit Queues on VMware ESXi 192
Configuring an Ethernet Adapter Policy to Enable Stateless Offloads with NVGRE 193
Configuring an Ethernet Adapter Policy to Enable Stateless Offloads with VXLAN 195
Ethernet and Fibre Channel Adapter Policies 197
Accelerated Receive Flow Steering 199
Guidelines and Limitations for Accelerated Receive Flow Steering 200
Interrupt Coalescing 200
Adaptive Interrupt Coalescing 201
Guidelines and Limitations for Adaptive Interrupt Coalescing 201
RDMA Over Converged Ethernet for SMB Direct 201
Guidelines and Limitations for SMB Direct with RoCE 201
Configuring a Default vNIC Behavior Policy 202
Deleting a vNIC from a LAN Connectivity Policy 203
Creating a LAN Connectivity Policy 203
Deleting a LAN Connectivity Policy 204
About the LAN and SAN Connectivity Policies 205
Privileges Required for LAN and SAN Connectivity Policies 205
Interactions between Service Profiles and Connectivity Policies 205
Creating a LAN Connectivity Policy 206
Creating a vNIC for a LAN Connectivity Policy 207
Deleting a vNIC from a LAN Connectivity Policy 210
Creating an iSCSI vNIC for a LAN Connectivity Policy 210
Deleting an iSCSI vNIC from a LAN Connectivity Policy 212
Network Control Policy 213
Configuring a Network Control Policy 213
Configuring Link Layer Discovery Protocol for Fabric Interconnect vEthernet Interfaces 216
Displaying Network Control Policy Details 216
Deleting a Network Control Policy 217
Creating a Multicast Policy 217
Deleting a Multicast Policy 218
Entering Multicast Policy Mode 219
Enter a Multicast Policy 219

Assigning a Global VLAN Multicast Policy 220
Disassociating a Global VLAN Multicast Policy 220
Disassociating a VLAN Multicast Policy 221
Configuring SRIOV HPN Connection Policy 222
Single Root I/O Virtualization HPN Connection Policy 222
Configuring SRIOV HPN Connection Policy 222
Assigning SRIOV-HPN Connection Policy to a vNIC 223
Deleting SRIOV HPN Connection Policy 224
GENEVE Offload 225
Enhanced Network Stack (ENS) Support 226
Configuring Ethernet Adapter Policies 227
Configuring an Ethernet Adapter Policy 227
Deleting an Ethernet Adapter Policy 229
Configuring the Default vNIC Behavior Policy 230
Default vNIC Behavior Policy 230
Configuring a Default vNIC Behavior Policy 230
Configuring a Network Control Policy 231
Deleting a Network Control Policy 233
Configuring Multicast Policies 234
Multicast Policy 234
Creating a Multicast Policy 235
Configuring IGMP Parameters 235
Modifying Multicast Policy Parameters 237
Assigning a VLAN Multicast Policy 237
Deleting a Multicast Policy 238
LACP Policy 239
Creating a LACP Policy 239
Editing a LACP Policy 240
Assigning LACP Policy to Port-Channels 240
Configuring UDLD Link Policies 241
Understanding UDLD 241
UDLD Configuration Guidelines 243
Configuring a UDLD Link Policy 243
Modifying the UDLD System Settings 244

Configuring a Link Profile 245
Assigning a Link Profile to a Port Channel Ethernet Interface 246
Assigning a Link Profile to a Port Channel FCoE Interface 246
Assigning a Link Profile to an Uplink Ethernet Interface 247
Assigning a Link Profile to an Uplink FCoE Interface 248
VMQ Connection Policy 248
Creating a VMQ Connection Policy 249

CHAPTER 12 Configuring MACsec 251
About MACsec 251
Key Lifetime and Hitless Key Rollover 252
Fallback Key 252
Guidelines and Limitations for MACsec 252
Enabling MACsec Configuration 255
Disabling MACsec Configuration 255
Creating a MACsec Policy 256
Viewing MACsec Policy 258
Deleting a MACsec Policy 258
Creating a MACsec Keychain 259
Viewing a MACsec Keychain 259
Deleting a MACsec Keychain 260
Creating a MACsec Key 260
Viewing MACsec Keys 262
Deleting a MACsec Key 263
Creating a LifeTime 263
Viewing a LifeTime 264
Deleting a LifeTime 265
Creating a MACsec Interface Configuration 265
Viewing MACsec Interface Configuration 267
Deleting a MACsec Interface Configuration 267
Configuring MACsec on an Uplink Interface 268
Viewing MACsec on an Uplink Interface 268
Deleting MACsec on an Uplink Interface 269
Configuring MACsec on an Uplink Port Channel Member Interface 270

Viewing MACsec on an Uplink Port Channel Member Interface 271
Deleting MACsec on an Uplink Port Channel Member Interface 271
Configurable EAPOL Destination and Ethernet Type 272
Enabling EAPOL Configuration 272
Disabling EAPOL Configuration 274
Displaying MACsec Sessions 274
Displaying MACsec Statistics 275

Preface

- Audience, on page xv
- Conventions, on page xv
- Related Cisco UCS Documentation, on page xvii
- Documentation Feedback, on page xvii

Audience

This guide is intended primarily for data center administrators with responsibilities and expertise in one or more of the following:

- Server administration
- Storage administration
- Network administration
- Network security

Conventions

Text Type	Indication
GUI elements	GUI elements such as tab titles, area names, and field labels appear in **this font**. Main titles such as window, dialog box, and wizard titles appear in **this font**.
Document titles	Document titles appear in *this font*.
TUI elements	In a Text-based User Interface, text the system displays appears in `this font`.
System output	Terminal sessions and information that the system displays appear in `this font`.
CLI commands	CLI command keywords appear in **this font**. Variables in a CLI command appear in *this font*.
[ ]	Elements in square brackets are optional.

Text Type	Indication
{x | y | z}	Required alternative keywords 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 nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.
< >	Nonprinting characters such as passwords are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

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

Tip Means *the following information will help you solve a problem*. The tips information might not be troubleshooting or even an action, but could be useful information, similar to a Timesaver.

Timesaver Means *the described action saves time*. You can save time by performing the action described in the paragraph.

Caution Means *reader be careful*. In this situation, you might perform an action that could result in equipment damage or loss of data.

Warning IMPORTANT SAFETY INSTRUCTIONS

This warning symbol means danger. You are in a situation that could cause bodily injury. Before you work on any equipment, be aware of the hazards involved with electrical circuitry and be familiar with standard practices for preventing accidents. Use the statement number provided at the end of each warning to locate its translation in the translated safety warnings that accompanied this device.

SAVE THESE INSTRUCTIONS

Related Cisco UCS Documentation

Documentation Roadmaps

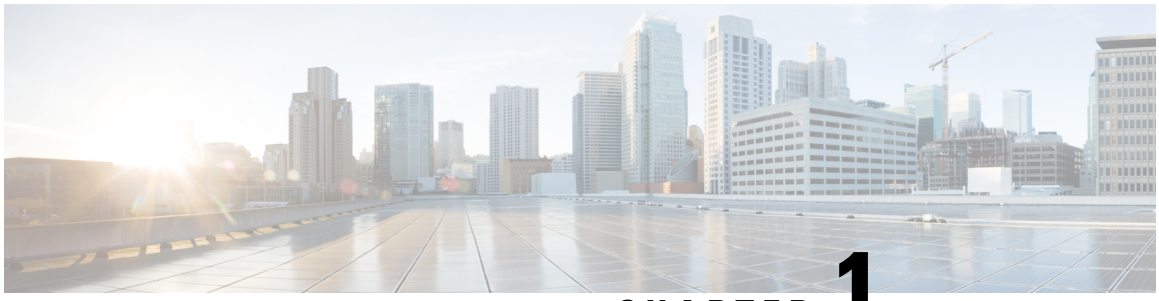
For a complete list of all B-Series documentation, see the *Cisco UCS B-Series Servers Documentation Roadmap* available at the following URL: https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/overview/guide/UCS_roadmap.html

For a complete list of all C-Series documentation, see the *Cisco UCS C-Series Servers Documentation Roadmapdoc roadmap* available at the following URL: https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/overview/guide/ucs_rack_roadmap.html.

For information on supported firmware versions and supported UCS Manager versions for the rack servers that are integrated with the UCS Manager for management, refer to Release Bundle Contents for Cisco UCS Software.

Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to ucs-docfeedback@external.cisco.com. We appreciate your feedback.



CHAPTER 1

New and Changed Information

- New and Changed Information, on page 1

New and Changed Information

This section provides information on new feature and changed behavior in Cisco UCS Manager, Release 6.0

Table 1: New Features and Changed Behavior in Cisco UCS Manager, Release 6.0(2b)

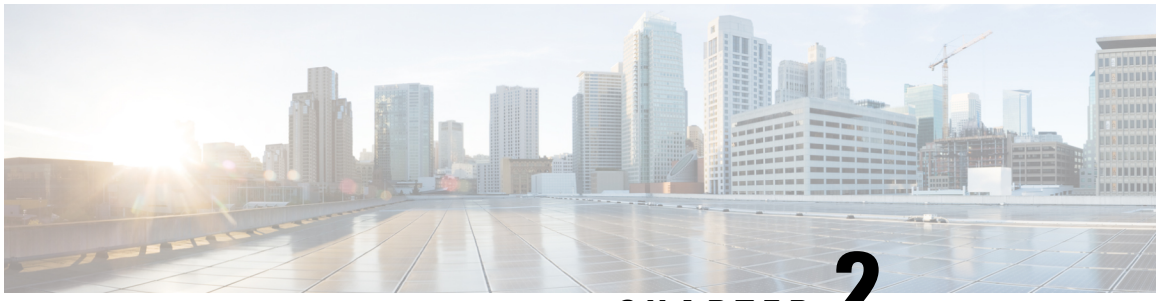
Feature	Description	Where Documented
Support for Cisco UCS 6600 Series Fabric Interconnect	Cisco UCS Manager supports Cisco UCS 6652 Fabric Interconnect	• Unified Ports on Cisco UCS 6600 Series Fabric Interconnects, on page 28 • Configuring Fibre Channel Ports on UCS 6600 Series Fabric Interconnects, on page 30 • Configuring Ethernet Ports on UCS 6600 Series Fabric Interconnects, on page 29 • VIC QinQ Tunneling - Supported Combinations and Limitations, on page 139 • Quality of Service, on page 153 • Creating a Multicast Policy, on page 217 • VLAN Port Count Optimization, on page 121 • Modifying the Reserved VLAN, on page 127

Feature	Description	Where Documented
Support for 50G Ethernet port for Cisco UCS 6652 Fabric Interconnect	Cisco UCS Manager now supports 50G port speed on the Ethernet port for Cisco UCS 6652 Fabric Interconnect.	Port Breakout Functionality on Cisco UCS 6652 Fabric Interconnect, on page 26
Support for 50G Ethernet port for Cisco UCS 6664 Fabric Interconnect	Cisco UCS Manager now supports 50G port speed on the Ethernet port for Cisco UCS 6664 Fabric Interconnect.	Port Functionality on Cisco UCS 6664 Fabric Interconnect, on page 25
Support for Spanning Tree Protocol (STP)-related faults	Cisco UCS Manager adds support for the reporting of Spanning Tree Protocol (STP)-related faults.	• STP Faults, on page 57 • Configuring STP Faults, on page 57 • Unconfiguring STP Faults, on page 58
Support for Receive Side Scaling (RSS) with Enhanced Network Stack (ENS)	Cisco UCS Manager now supports RSS with the Enhanced Network Stack (ENS) on VMware ESXi 8.0 and 9.0. This feature optimizes traffic distribution across multiple CPU cores and requires GENEVE offload to be enabled on Cisco UCS VIC 15000 Series adapters.	• GENEVE Offload, on page 225 • Enhanced Network Stack (ENS) Support, on page 226 • Configuring an Ethernet Adapter Policy, on page 188
MTU Configuration and Forwarding Limits	Cisco UCS VIC 15000 Series adapter supports an increased MTU range of 1500 to 9158.	• Configuring a vNIC Template, on page 185 • Creating a vNIC for a LAN Connectivity Policy, on page 207
Traffic Rate Monitoring for veth and vfc Interfaces	Cisco UCS Manager now supports displaying real-time traffic rates in Gbps and packets per second (pps) for Virtual Ethernet (veth) and Virtual Fibre Channel (vfc) interfaces via the CLI. This enhancement allows for more effective troubleshooting of network performance and congestion.	• Displaying VIF Port Statistics Obtained From the Adaptor, on page 44 • Verifying VFC to Vethernet Binding and Real-Time Statistics, on page 47

Table 2: New Features and Changed Behavior in Cisco UCS Manager, Release 6.0(1b)

Feature	Description	Where Documented
Support for Cisco UCS 6600 Series Fabric Interconnect	Cisco UCS Manager supports Cisco UCS 6664 Fabric Interconnect	• Configuring Fibre Channel Ports on UCS 6600 Series Fabric Interconnects, on page 30 • VIC QinQ Tunneling - Supported Combinations and Limitations, on page 139 • Quality of Service, on page 153 • Creating a Multicast Policy, on page 217 • VLAN Port Count Optimization, on page 121 • Modifying the Reserved VLAN, on page 127
MACsec support for Cisco UCS 6664 Fabric Interconnect and Cisco UCS X-Series Direct	Cisco UCS Manager 6.0(1b) adds MACsec support for Cisco UCS 6664 Fabric Interconnect and Cisco UCS Fabric Interconnect 9108 100G (Cisco UCS X-Series Direct).	Guidelines and Limitations for MACsec, on page 252
Deprecated support for Cisco UCS 6300 series Fabric Interconnect.	Cisco UCS Manager support for Cisco UCS 6300 Series Fabric Interconnect is deprecated.	-

New and Changed Information



CHAPTER 2

Overview

- Overview, on page 5
- Cisco UCS Manager User CLI Documentation, on page 5

Overview

This guide includes the following information:

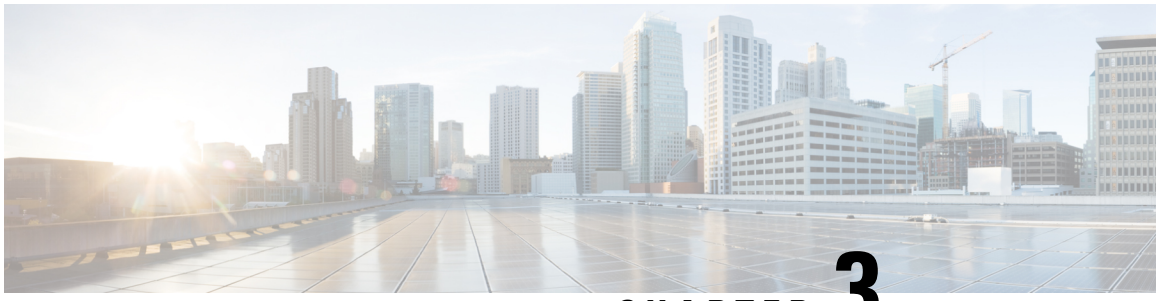
- Configure/Enable Server Ports; Configure/Enable Uplink Ports; Configure/Enable FC Ports.
- Create LAN Pin Groups
- Create VLANs and VLAN groups
- Create Server Links
- Configure QoS System Class
- Configure Global Policies
- Monitor Network Health
- Traffic Monitoring

Cisco UCS Manager User CLI Documentation

Cisco UCS Manager offers you a set of smaller, use-case based documentation described in the following table:

Guide	Description
Cisco UCS Manager Getting Started Guide	Discusses Cisco UCS architecture and Day 0 operations, including Cisco UCS Manager initial configuration, and configuration best practices.

Guide	Description
Cisco UCS Manager Administration Guide	Discusses password management, role-based access configuration, remote authentication, communication services, CIMC session management, organizations, backup and restore, scheduling options, BIOS tokens and deferred deployments.
Cisco UCS Manager Infrastructure Management Guide	Discusses physical and virtual infrastructure components used and managed by Cisco UCS Manager.
Cisco UCS Manager Firmware Management Guide	Discusses downloading and managing firmware, upgrading through Auto Install, upgrading through service profiles, directly upgrading at endpoints using firmware auto sync, managing the capability catalog, deployment scenarios, and troubleshooting.
Cisco UCS Manager Server Management Guide	Discusses the new licenses, registering Cisco UCS domains with Cisco UCS Central, power capping, server boot, server profiles and server-related policies.
Cisco UCS Manager Storage Management Guide	Discusses all aspects of storage management such as SAN and VSAN in Cisco UCS Manager.
Cisco UCS Manager Network Management Guide	Discusses all aspects of network management such as LAN and VLAN connectivity in Cisco UCS Manager.
Cisco UCS Manager System Monitoring Guide	Discusses all aspects of system and health monitoring including system statistics in Cisco UCS Manager.
Cisco UCS S3260 Server Integration with Cisco UCS Manager	Discusses all aspects of management of UCS S-Series servers that are managed through Cisco UCS Manager.



CHAPTER 3

LAN Connectivity

- Fabric Interconnect Overview, on page 7
- IOMs and Fabric Interconnects Connectivity, on page 7
- Configuring the Fabric Interconnects, on page 9
- Fabric Evacuation, on page 14
- Fabric Interconnect Port Types, on page 19
- Fabric Interconnect Switching Modes, on page 19

Fabric Interconnect Overview

The fabric interconnect is the core component of Cisco UCS. The Cisco UCS Fabric Interconnects provide uplink access to LAN, SAN, and out-of-band management segment. Cisco UCS infrastructure management is through the embedded management software, Cisco UCS Manager, for both hardware and software management. The Cisco UCS Fabric Interconnects are Top-of-Rack devices, and provide unified access to the Cisco UCS domain.

The Cisco UCS FIs provide network connectivity and management for the connected servers. The Cisco UCS Fabric Interconnects run the Cisco UCS Manager control software and consist of expansion modules for the Cisco UCS Manager software.

For more information about Cisco UCS Fabric Interconnects, see the *Cisco UCS Manager Getting Started Guide*.

IOMs and Fabric Interconnects Connectivity

Each chassis is equipped with two IOMs: IOM 1 should be connected to Fabric Interconnect A. IOM 2 should be connected to Fabric Interconnect B. This configuration provides redundant paths, ensuring uninterrupted operation of the Cisco UCS system even in the event of a failure in one of the Fabric Interconnects or IOMs. Additionally, this configuration enables traffic load distribution across both Fabric Interconnects, enhancing load balancing and increasing throughput. As a result, the Cisco UCS system achieves high availability, reliability, and optimal performance, making it ideal for data center environments.

Uplink Connectivity

Use fabric interconnect ports configured as uplink ports to connect to uplink upstream network switches. Connect these uplink ports to upstream switch ports as individual links, or as links configured as port channels. Port channel configurations provide bandwidth aggregation as well as link redundancy.

You can achieve northbound connectivity from the fabric interconnect through a standard uplink, a port channel, or a virtual port channel configuration. The port channel name and ID configured on fabric interconnect should match the name and ID configuration on the upstream Ethernet switch.

It is also possible to configure a port channel as a vPC, where port channel uplink ports from a fabric interconnect are connected to different upstream switches. After all uplink ports are configured, create a port channel for these ports.

Downlink Connectivity

Beginning with release 4.3(2a), Cisco UCS Manager supports Cisco UCS X9508 server chassis with Cisco UCS X-Series servers. Cisco UCS X-Series servers support Intelligent Fabric Modules (IFM), which function similarly to the Input/Output Module (IOM) in Cisco UCS B-Series servers. This guide uses the term IOM to refer both IOM and IFM.

Each fabric interconnect is connected to IOMs in the UCS chassis, which provides connectivity to each blade server. Internal connectivity from blade servers to IOMs is transparently provided by Cisco UCS Manager using 10BASE-KR Ethernet standard for backplane implementations, and no additional configuration is required. You must configure the connectivity between the fabric interconnect server ports and IOMs. Each IOM, when connected with the fabric interconnect server port, behaves as a line card to fabric interconnect, hence IOMs should never be cross-connected to the fabric interconnect. Each IOM is connected directly to a single fabric interconnect.

The Fabric Extender (also referred to as the IOM, or FEX) logically extends the fabric interconnects to the blade server. The best analogy is to think of it as a remote line card that's embedded in the blade server chassis, allowing connectivity to the external world. IOM settings are pushed via Cisco UCS Manager and are not managed directly. The primary functions of this module are to facilitate blade server I/O connectivity (internal and external), multiplex all I/O traffic up to the fabric interconnects, and help monitor and manage the Cisco UCS infrastructure.

Configure Fabric interconnect ports that should be connected to downlink IOM cards as server ports. Make sure there is physical connectivity between the fabric interconnect and IOMs. You must also configure the IOM ports and the global chassis discovery policy.

Note For UCS 2200 I/O modules, you can also select the Port Channel option and all I/O module-connected server ports will be automatically added to a port channel.

Configuring the Fabric Interconnects

Fabric Interconnect Information Policy

Fabric Interconnect Information Policy enables you to display the uplink switches that are connected to fabric interconnect.

Important You must enable the information policy on the fabric interconnect to view the details of SAN, LAN, and LLDP neighbours of the fabric interconnect.

Installing Secure FPGA

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fabric-interconnect {a | b}**	Enters fabric interconnect mode for the specified fabric interconnect.
Step 2	UCS-A/fabric-interconnect# **show fault**	Displays if endpoint FPGA firmware is secured or unsecured.
Step 3	UCS-A/fabric-interconnect # **activate secure-fpga**	Initiates installation of secure FPGA on fabric interconnect. **Warning** This command will upgrade the FPGA and automatically reboot the system after completion of the FPGA upgrade. Kindly refrain from reloading or power-cycling the system during the upgrade, as the manual reboot will result in failure of Fabric Interconnect.
Step 4	UCS-A/fabric-interconnect * # **commit-buffer**	Commits the transaction to the system configuration.

Cisco UCS Manager restarts the fabric interconnect, logs you out, and disconnects Cisco UCS Manager CLI.

Example

The following example shows how to install secure FPGA on the fabric interconnect::

```
UCS-A# scope fabric-interconnect {a | b}
UCS-A/fabric-interconnect# activate secure-fpga
Warning: This command will reset Fabric Interconnect and the system will be down till the
```

```
Fabric Interconnect is reset.
UCS-A/fabric-interconnect# commit-buffer
```

Enabling the Information Policy on the Fabric Interconnect

Note By default, the information policy is disabled on the fabric interconnect.

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope system**	Enters system mode.
Step 2	UCS-A/system # **scope info-policy**	Enters the information policy state.
Step 3	(Optional) UCS-A/system/info-policy # **show**	Shows if the information policy is enabled or disabled.
Step 4	UCS-A/system/info-policy # **enable**	Enables the information policy on the fabric interconnect.
Step 5	UCS-A/system/info-policy* # **commit-buffer**	Enables the information policy on the fabric interconnect.

Example

The following example shows how to enable the information policy on the fabric interconnect:

```
UCS-A#  scope system
UCS-A/system # scope info-policy
UCS-A/system/info-policy # show
Info Policy:
State: Disabled
UCS-A/system/info-policy # enable
UCS-A/system/info-policy* # commit-buffer
UCS-A/system/info-policy #
```

Disabling the Information Policy on the Fabric Interconnect

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope system**	Enters system mode.
Step 2	UCS-A/system # **scope info-policy**	Enters the information policy state.

	Command or Action	Purpose
Step 3	(Optional) UCS-A/system/info-policy # **show**	Shows if the information policy is enabled or disabled.
Step 4	UCS-A/system/info-policy # **disable**	Disables the information policy on the fabric interconnect.
Step 5	UCS-A/system/info-policy* # **commit-buffer**	Disables information policy on the fabric interconnect.

Example

The following example shows how to disable the information policy on the fabric interconnect:

```
UCS-A#  scope system
UCS-A/system # scope info-policy
UCS-A/system/info-policy # show
Info Policy:
State: Enabled
UCS-A/system/info-policy # disable
UCS-A/system/info-policy* # commit-buffer
UCS-A/system/info-policy #
```

Viewing the LAN Neighbors of the Fabric Interconnect

You must enable the information policy on the fabric interconnect to view the LAN neighbors.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fabric-interconnect** {**a** | **b**}	Enters fabric interconnect mode for the specified fabric interconnect.
Step 2	UCS-A/fabric-interconnect # **show lan-neighbors**	Displays the fabric interconnect LAN neighbors.

Example

The following example shows how to display the LAN neighbors of the fabric interconnect:

```
UCS-A #  scope fabric-interconnect a
UCS-Afabric-interconnect # show lan-neighbors
Info Policy:Enabled
Lan Neighbors:
Local Interface: Ethernet1/2
Device Id: bgl-samc02-B(SSI140305YK)
IPv4 Address: 10.105.214.105
FI Port DN: sys/switch-A/slot-1/switch-ether/port-2
```

Viewing the SAN Neighbors of the Fabric Interconnect

You must enable the information policy on the fabric interconnect to view the SAN neighbors.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fabric-interconnect** {**a** | **b**}	Enters fabric interconnect mode for the specified fabric interconnect.
Step 2	UCS-A/fabric-interconnect # **show san-neighbors**	Displays the fabric interconnect SAN neighbors.

Example

The following example shows how to display the SAN neighbors of the fabric interconnect :

```
UCS-A #  scope fabric-interconnect a
UCS-A/fabric-interconnect # show san-neighbors
Info Policy: Enabled
San neighbors:
Local Interface: fc2/1
Port VSAN: 100
Fabric Mgmt Addr: 10.65.124.252
Fabric pwwn: 20:02:00:05:9b:22:ad:C0
Fabric nwwn: 20:64:00:05:9b:22:ad:C1
My pwwn: 20:41:00:0d:ec:ee:dd:00
My nwwn: 20:64:00:0d:ec:ee:dd:01
FI Port DN: sys/switch-A/slot-2/switch-fc/port-1
```

Viewing the LLDP Neighbors of the Fabric Interconnect

You must enable the information policy on the fabric interconnect to view the LLDP neighbors.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fabric-interconnect** {**a** | **b**}	Enters fabric interconnect mode for the specified fabric interconnect.
Step 2	UCS-A/fabric-interconnect # **show lldp-neighbors**	Displays the fabric interconnect LLDP neighbors.

Example

The following example shows how to display the LLDP neighbors of the fabric interconnect :

```
UCS-A #  scope fabric-interconnect a
UCS-A/fabric-interconnect # show lldp-neighbors
Info Policy: Enabled
```

```
Lldp Neighbors:

Local Interface: Eth1/5
Chassis Id: 000d.ecff.5e90
Remote Interface: Eth1/9
Remote Port Description: Ethernet1/9
System Name: bgl-samc02-B
System Description: Cisco Nexus Operating System (NX-OS) Software TAC support:
http://www.cisco.com/tac Copyright (c) 2002-2011, Cisco Systems, Inc
System Capabilities: B
Enabled Capabilities: B
Native VLAN: 1
IPv4 Mgmt Address: 10.105.214.105
FI Port DN: sys/switch-A/slot-1/switch-ether/port-5
```

Installing Secure FPGA

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fabric-interconnect {a | b}**	Enters fabric interconnect mode for the specified fabric interconnect.
Step 2	UCS-A/fabric-interconnect# **show fault**	Displays if endpoint FPGA firmware is secured or unsecured.
Step 3	UCS-A/fabric-interconnect # **activate secure-fpga**	Initiates installation of secure FPGA on fabric interconnect. **Warning** This command will upgrade the FPGA and automatically reboot the system after completion of the FPGA upgrade. Kindly refrain from reloading or power-cycling the system during the upgrade, as the manual reboot will result in failure of Fabric Interconnect.
Step 4	UCS-A/fabric-interconnect * # **commit-buffer**	Commits the transaction to the system configuration.

Cisco UCS Manager restarts the fabric interconnect, logs you out, and disconnects Cisco UCS Manager CLI.

Example

The following example shows how to install secure FPGA on the fabric interconnect::

```
UCS-A# scope fabric-interconnect {a | b}
UCS-A/fabric-interconnect# activate secure-fpga
Warning: This command will reset Fabric Interconnect and the system will be down till the
```

```
Fabric Interconnect is reset.
UCS-A/fabric-interconnect# commit-buffer
```

Fabric Evacuation

Cisco UCS Manager introduces fabric evacuation, which is the ability to evacuate all traffic that flows through a fabric interconnect from all servers attached to it through an IOM or FEX while upgrading a system. Fabric evacuation is not supported on direct-attached rack servers.

Upgrading the secondary fabric interconnect in a system disrupts active traffic on the fabric interconnect. This traffic fails over to the primary fabric interconnect. You can use fabric evacuation during the upgrade process as follows:

1. Stop all the traffic that is active through a fabric interconnect.
2. For vNICs configured with failover, verify that the traffic has failed over by using Cisco UCS Manager, or tools such as vCenter.
3. Upgrade the secondary fabric interconnect.
4. Restart all the stopped traffic flows.
5. Change the cluster lead to the secondary fabric interconnect.
6. Repeat steps 1 to 4 and upgrade the primary fabric interconnect.

Note

- Fabric interconnect traffic evacuation is supported only in a cluster configuration.
- You can evacuate traffic only from the subordinate fabric interconnect.
- The IOM or FEX backplane ports of the fabric interconnect on which evacuation is configured will go down, and their state will appear as **Admin down**. During the manual upgrade process, to move these backplane ports back to the Up state and resume traffic flow, you must explicitly configure **Admin Evac Mode** as **Off**.
- Starting with Cisco UCS Manager Release 3.1(3), you can use fabric evacuation during Auto Install.
- If you use fabric evacuation outside of the upgrade process, you must re-acknowledge the FEX to get the VIFs back to the online state.

Stopping Traffic on a Fabric Interconnect

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope fabric-interconnect** {**a** | **b**}	Enters the fabric interconnect mode.
Step 2	UCS-A /fabric-interconnect # **stop server traffic** [**force**]	Stops all the traffic that is active through the specified Fabric Interconnect.

	Command or Action	Purpose
		Use the **force** option to evacuate a fabric interconnect regardless of its current evacuation state.
Step 3	UCS-A /fabric-interconnect # **commit-buffer**	Commits the transaction to the system configuration.

Example

This example shows how to stop all traffic that is active through Fabric Interconnect B:

```
UCS-A# scope fabric-interconnect b
UCS-A /fabric-interconnect # stop server traffic
Warning: Enabling fabric evacuation will stop all traffic through this Fabric Interconnect
 from servers attached through IOM/FEX. The traffic will fail over to the Primary Fabric
Interconnect for fail over vnics.
UCS-A /fabric-interconnect # commit-buffer
```

Displaying the Status of Evacuation for a Fabric Interconnect

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope fabric-interconnect** {**a** | **b**}	Enters fabric interconnect mode for the specified fabric interconnect.
Step 2	UCS-A /fabric-interconnect # **show detail**	Displays details about the specified fabric interconnect.

Example

This example shows how to display the status of a fabric interconnect.

Note **Admin Evacuation**and **Oper Evacuation** and show the status of evacuation at the fabric interconnect.

```
 UCS-A /fabric-interconnect # show detail

Fabric Interconnect:
    ID: B
    Product Name: Cisco UCS 6248UP
    PID: UCS-FI-6248UP
    VID: V01
    Vendor: Cisco Systems, Inc.
    Serial (SN): SSI171400HG
    HW Revision: 0
```

```
    Total Memory (MB): 16165
    OOB IP Addr: 10.193.32.172
    OOB Gateway: 10.193.32.1
    OOB Netmask: 255.255.255.0
    OOB IPv6 Address: ::
    OOB IPv6 Gateway: ::
    Prefix: 64
    Operability: Operable
    Thermal Status: Ok
    Admin Evacuation: On
    Oper Evacuation: On
    Current Task 1:
    Current Task 2:
    Current Task 3:
```

Displaying the Status of Evacuation for an IOM

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope chassis** *chassis-num*	Enters chassis mode for the specified chassis.
Step 2	UCS-A /chassis # **scope iom** *iom-id*	Enters chassis IOM mode for the specified IOM.
Step 3	UCS-A /chassis/iom # **show detail**	Displays evacuation status details for the specified IOM.

Example

This example shows how to display the evacuation status details for an IOM.

Note **Oper Evacuation** shows the operational status of evacuation for the IOM.

```
UCS-A# scope chassis 1
UCS-A /chassis # scope iom 1
UCS-A /chassis/iom # show detail

IOM:
    ID: 1
    Side: Left
    Fabric ID: A
    User Label:
    Overall Status: Fabric Conn Problem
    Oper qualifier: Server Port Problem
    Operability: Operable
    Presence: Equipped
    Thermal Status: OK
    Discovery: Online
    Config State: Ok
    Peer Comm Status: Connected
    Product Name: Cisco UCS 2204XP
```

```
    PID: UCS-IOM-2204XP
    VID: V02
    Part Number: 73-14488-02
    Vendor: Cisco Systems Inc
    Serial (SN): FCH1718J9FT
    HW Revision: 0
    Mfg Date: 2013-05-12T00:00:00.000
    Controller Subject: Iocard
    Fabric Port Aggregation Capability: Port Channel
    Oper Evacuation: On
    Current Task 1:
    Current Task 2:
```

Verifying Fabric Evacuation

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **show service-profile circuit server** *server-id*	Shows the network circuit information for the service profile associated with the specified server.

Example

The following example shows the VIF (Virtual NIC) paths before fabric evacuation.

Note

- VIF at Fabric Interconnect A shows that traffic is initially active through the fabric interconnect.
- VIF at Fabric Interconnect B is passive before evacuation.

```
UCS-A# show service-profile circuit server 1/6
Service Profile: test1
Server: 1/6
    Fabric ID: A
        Path ID: 1
        VIF        vNIC            Link State  Oper State Prot State    Prot Role   Admin
Pin  Oper Pin   Transport
          ---------- --------------- ----------- ---------- ------------ ----------- ----------
 ---------- ---------
               692 eth0            Up          Active     Active        Primary     0/0
     1/15       Ether
    Fabric ID: B
        Path ID: 1
        VIF        vNIC            Link State  Oper State Prot State    Prot Role   Admin
Pin  Oper Pin   Transport
          ---------- --------------- ----------- ---------- ------------ ----------- ----------
 ---------- ---------
               693 eth0            Up          Active     Passive       Backup      0/0
     1/15       Ether
```

```
UCS-A#
```

The following example shows the VIF paths after Fabric Interconnect A is evacuated.

Note

- After failover, the VIF state at Fabric Interconnect A goes into error.
- VIF at Fabric Interconnect B takes over as active.

```
UCS-A# show service-profile circuit server 1/6
Service Profile: test1
Server: 1/6
    Fabric ID: A
        Path ID: 1
        VIF        vNIC            Link State  Oper State Prot State    Prot Role   Admin
Pin  Oper Pin   Transport
       ---------- --------------- ----------- ---------- ------------- ----------- ----------
 ---------- ---------
              692 eth0            Error       Error      Active        Primary     0/0
     0/0        Ether
    Fabric ID: B
        Path ID: 1
        VIF        vNIC            Link State  Oper State Prot State    Prot Role   Admin
Pin  Oper Pin   Transport
       ---------- --------------- ----------- ---------- ------------- ----------- ----------
 ---------- ---------
              693 eth0            Up          Active     Passive       Backup      0/0
     1/15       Ether
UCS-A#
```

Restarting Traffic on a Fabric Interconnect

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope fabric-interconnect** {**a** | **b**}	Enters the fabric interconnect mode.
Step 2	UCS-A /fabric-interconnect # **start server traffic**	Restarts traffic through the specified fabric interconnect.
Step 3	UCS-A /fabric-interconnect # **commit-buffer**	Commits the transaction to the system configuration.

Example

This example shows how to restart traffic through Fabric Interconnect B:

```
UCS-A# scope fabric-interconnect b
UCS-A /fabric-interconnect # start server traffic
Warning: Resetting fabric evacuation will cause server traffic that failed over to the
```

```
Primary Fabric Interconnect to fail back to this Fabric Interconnect.
UCS-A /fabric-interconnect # commit-buffer
```

Fabric Interconnect Port Types

By default, all fabric interconnect ports are unconfigured. For Ethernet LAN connectivity, fabric interconnect ports can be in the following states:

- **Unconfigured**—Port is not configured and cannot be used.
- **Server Port**—Port is configured for downlink connection to an IOM Fabric Extender (FEX) module in a blade chassis.
- **Uplink Port**—Port is configured for uplink connection to the upstream Ethernet switch. Uplink ports are always configured as trunk ports.
- **Disabled**—Port is configured either as an uplink or server port and is currently disabled by the administrator.

On Cisco UCS 6400 Series Fabric Interconnects, ports 1 to 16 are unified ports and can be configured as either Ethernet or FC ports.

Note The Cisco UCS 6454 Fabric Interconnect supported 8 unified ports (ports 1 - 8) with Cisco UCS Manager 4.0(1) and 4.0(2), but with release 4.0(4) and later it supports 16 unified ports (ports 1 - 16).

On Cisco UCS 6536 Fabric Interconnects, ports 33 to 36 are unified ports. The unified ports can be configured as either Ethernet or FC ports.

On Cisco UCS Fabric Interconnects 9108 100G, also referred as Cisco UCS X-Series Direct (UCSX-S9108-100G), supports port breakout for Ethernet Ports (1-8) and Unified Ports (1-2). The unified ports can be configured as either Ethernet or FC ports.

On Cisco UCS 6664 Fabric Interconnect, ports 25-40 are unified ports. These ports can be configured as either Ethernet or Fibre Channel ports.

Note For detailed information on each type of Fabric Interconnect port, see Cisco UCS Manager Getting Started Guide.

Fabric Interconnect Switching Modes

The Cisco UCS Fabric Interconnects operate in two main switching modes: Ethernet or Fibre Channel. These modes are independent of each other. They determine how the fabric interconnect behaves as a device between the server and network/server and storage device.

Ethernet Switching Mode

The Ethernet switching mode determines how the fabric interconnect behaves as a switching device between the servers and the network. The fabric interconnect operates in either of the following Ethernet switching modes:

End-Host Mode

End-host mode allows the fabric interconnect to act as an end host to the network, representing all servers (hosts) connected to it through vNICs. This behavior is achieved by pinning (either dynamically pinning or hard pinning) vNICs to uplink ports, which provides redundancy to the network, and makes the uplink ports appear as server ports to the rest of the fabric.

In end-host mode, the fabric interconnect does not run the Spanning Tree Protocol (STP), but it avoids loops by denying uplink ports from forwarding traffic to each other and by denying egress server traffic on more than one uplink port at a time. End-host mode is the default Ethernet switching mode and should be used if either of the following is used upstream:

- Layer 2 switching for Layer 2 aggregation
- Virtual Switching System (VSS) aggregation layer

Note When you enable end-host mode, if a vNIC is hard pinned to an uplink port and this uplink port goes down, the system cannot repin the vNIC, and the vNIC remains down.

Switch Mode

Switch mode is the traditional Ethernet switching mode. The fabric interconnect runs STP to avoid loops, and broadcast and multicast packets are handled in the traditional way. Use the switch mode only if the fabric interconnect is directly connected to a router, or if either of the following is used upstream:

- Layer 3 aggregation
- VLAN in a box

Note For both Ethernet switching modes, even when vNICs are hard-pinned to uplink ports, all server-to-server unicast traffic in the server array is sent only through the fabric interconnect and is never sent through uplink ports. Server-to-server multicast and broadcast traffic is sent through all uplink ports in the same VLAN.

Cisco UCS Fabric Interconnect in Switch Mode with Cisco MDS 9000 Family Fibre Channel Switching Modules

While creating a port channel between a Cisco MDS 9000 family FC switching module and a Cisco UCS Fabric Interconnect in switch mode, use the following order:

1. Create the port channel on the MDS side.
2. Add the port channel member ports.
3. Create the port channel on the Fabric Interconnect side.

4. Add the port channel member ports.

If you create the port channel on the Fabric Interconnect side first, the ports will go into a suspended state.

When the Cisco UCS Fabric Interconnect is in switch mode, the port channel mode can only be in **ON** mode and not **Active**. However, to get the peer wwn information for the Fabric Interconnect, the port channel must be in **Active** mode.

Configuring Ethernet Switching Mode

Important When you change the Ethernet switching mode, Cisco UCS Manager logs you out and restarts the fabric interconnect. For a cluster configuration, Cisco UCS Manager restarts both fabric interconnects. The subordinate fabric interconnect reboots first as a result of the change in switching mode. The primary fabric interconnect reboots only after you acknowledge it in **Pending Activities**. The primary fabric interconnect can take several minutes to complete the change in Ethernet switching mode and become system ready. The existing configuration is retained.

While the fabric interconnects are rebooting, all blade servers lose LAN and SAN connectivity, causing a complete outage of all services on the blades. This might cause the operating system to fail.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **set mode** {**end-host** | **switch**}	Sets the fabric interconnect to the specified switching mode.
Step 3	UCS-A /eth-uplink # **commit-buffer**	Commits the transaction to the system configuration. Cisco UCS Manager restarts the fabric interconnect, logs you out, and disconnects Cisco UCS Manager CLI.

Example

The following example sets the fabric interconnect to end-host mode and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # set mode end-host
Warning: When committed, this change will cause the switch to reboot
UCS-A /eth-uplink* # commit-buffer
UCS-A /eth-uplink #
```

Fibre Channel Switching Mode

The Fibre Channel switching mode determines how the fabric interconnect behaves as a switching device between the servers and storage devices. The fabric interconnect operates in either of the following Fibre Channel switching modes:

End-Host Mode

End-host mode is synonymous with N Port Virtualization (NPV) mode. This mode is the default Fibre Channel Switching mode. End-host mode allows the fabric interconnect to act as an end host to the connected fibre channel networks, representing all servers (hosts) connected to it through virtual host bus adapters (vHBAs). This behavior is achieved by pinning (either dynamically pinning or hard-pinning) vHBAs to Fibre Channel uplink ports, which makes the Fibre Channel ports appear as server ports (N-ports) to the rest of the fabric. When in end-host mode, the fabric interconnect avoids loops by preventing uplink ports from receiving traffic from one another.

Note When you enable end-host mode, if a vHBA is hard-pinned to an uplink Fibre Channel port and this uplink port goes down, the system cannot repin the vHBA, and the vHBA remains down.

Switch Mode

Switch mode is not the default Fibre Channel switching mode. Switch mode allows the fabric interconnect to connect directly to a storage device. Enabling Fibre Channel switch mode is useful in Pod models where there is no SAN (for example, a single Cisco UCS domain that is connected directly to storage), or where a SAN exists (with an upstream MDS). In Fibre Channel switch mode, SAN pin groups are irrelevant. Any existing SAN pin groups are ignored.

Configuring Fibre Channel Switching Mode

Note When the Fibre Channel switching mode is changed, both Cisco UCS fabric interconnects reload simultaneously. Reloading the fabric interconnects will cause a system-wide downtime for approximately 10 to 15 minutes.

Procedure

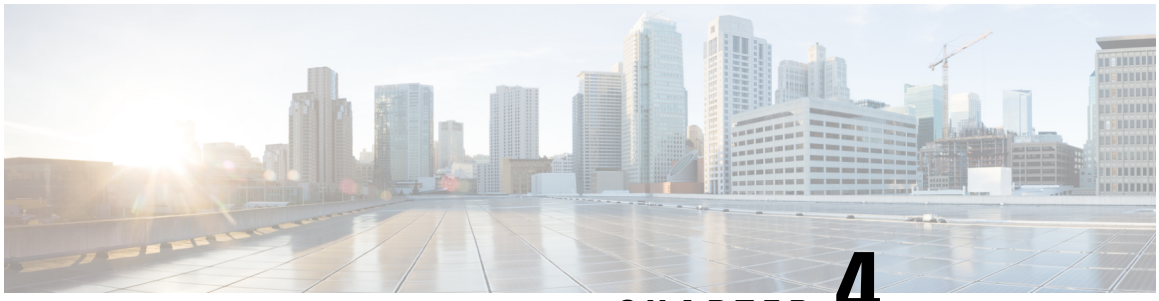
	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters Fibre Channel uplink mode.
Step 2	UCS-A /fc-uplink # **set mode** {**end-host** | **switch**}	Sets the fabric interconnect to the specified switching mode.
Step 3	UCS-A /fc-uplink # **commit-buffer**	Commits the transaction to the system configuration.

	Command or Action	Purpose
		Cisco UCS Manager restarts the fabric interconnect, logs you out, and disconnects Cisco UCS Manager CLI.

Example

The following example shows how to set the fabric interconnect to end-host mode and commit the transaction:

```
UCS-A # scope fc-uplink
UCS-A /fc-uplink # set mode end-host
UCS-A /fc-uplink* # commit-buffer
UCS-A /fc-uplink #
```



CHAPTER 4

LAN Ports and Port Channels

- Unified Ports on the Cisco UCS Fabric Interconnects, on page 25
- Unified Ports on the Cisco UCS 6500 Series Fabric Interconnects, on page 32
- Unified Breakout Ports for Cisco UCS X-Series Direct, on page 42
- Physical and Backplane Ports, on page 44
- Server Ports, on page 49
- Uplink Ethernet Ports, on page 53
- Appliance Ports, on page 58
- FCoE Uplink Ports, on page 65
- Unified Storage Ports, on page 69
- Unified Uplink Ports, on page 70
- FCoE and Fibre Channel Storage Ports, on page 71
- Uplink Ethernet Port Channels, on page 73
- Appliance Port Channels, on page 76
- Fibre Channel Port Channels, on page 80
- FCoE Port Channels, on page 85
- Unified Uplink Port Channel, on page 87
- Event Detection and Action, on page 88
- Adapter Port Channels, on page 93
- Fabric Port Channels, on page 94

Unified Ports on the Cisco UCS Fabric Interconnects

Port Functionality on Cisco UCS 6664 Fabric Interconnect

The Cisco UCS 6664 Fabric Interconnect is a 2-rack unit (RU) fixed-port system designed for flexible and high-performance networking. It features 64 front panel ports that support a variety of connectivity options.

Front Panel Port Configuration and Types

The UCS 6664 Fabric Interconnect supports the following possible configurations or port types for each front panel port:

Port Number	Port Hardware	Admin Port Speed	Port Type	Port Role
1-24	QSFP 28	40 Gbps/100 Gbps	Gigabit Ethernet	• Server Port • Ethernet/FCoE Uplink Port • FCoE Storage Port • Appliance Port (EHM only) • Monitor Port
25-40 (Unified Ports)	SFP28	16 Gbps/32 Gbps/64 Gbps	Fibre Channel (FC)	• FC Uplink Port • FC Storage Port
		10 Gbps/25 Gbps/50Gbps	Gigabit Ethernet	• Server Port • Ethernet/FcoE Uplink Port • Appliance Port (EHM only) • Monitor Port
41-64 **Note:** Ports 49–64 are MAC Security (MACsec)-capable	QSFP 28	40 Gbps/100 Gbps	Gigabit Ethernet	• Server Port • Ethernet/FCoE Uplink Port • FCoE Storage Port • Appliance Port (EHM only) • Monitor Port

Note Breakout port functionality is not supported on Cisco UCS 6664 Fabric Interconnect.

Port Breakout Functionality on Cisco UCS 6652 Fabric Interconnect

The Cisco UCS 6652 Fabric Interconnect is a 1-**Rack Unit (RU)** fixed-port system equipped with port breakout functionality on its high-bandwidth uplink ports. This allows to subdivide a single high-bandwidth **Quad Small Form-factor Pluggable Double Density (QSFP-DD)** port into multiple lower-bandwidth ports, optimizing port utilization and adapting to various infrastructure requirements.

Table 3: Breakout Options and Logical Ports

Physical Port	Breakout Options	Logical Ports After Breakout	Supported Speeds through breakout cables
Ethernet 1/49 - 1/52	4x100G	Ethernet 1/49/1 to 1/52/4	Up to 4x100 Gbps
Ethernet 1/49 - 1/52	2x100G	Ethernet 1/49/1 to 1/52/2	Up to 2x100 Gbps
Ethernet 1/49 - 1/52	4x25G	Ethernet 1/49/1 to 1/52/4	Up to 4x25 Gbps
Ethernet 1/49 - 1/52	4x10G	Ethernet 1/49/1 to 1/52/4	Up to 4x10 Gbps

Port Configuration and Types

The Cisco UCS 6652 Fabric Interconnect supports the following possible configurations or port types for each front panel port:

Port Number	Port Hardware	Admin Port Speed	Port Type	Port Role
1–16	SFP28	10/25/50 Gbps	Gigabit Ethernet	• Server Port • Ethernet/FCoE Uplink Port • FCoE Storage Port • Appliance Port (End-Host Mode (EHM) only) • Monitor Port
17–32 (Unified Ports)	SFP28	16/32/64 Gbps	Fibre Channel (FC)	• FC Uplink Port • FC Storage Port
		10/25/50 Gbps	Gigabit Ethernet	• Server Port • Ethernet/FCoE Uplink Port • Appliance Port (EHM only) • Monitor Port

Port Number	Port Hardware	Admin Port Speed	Port Type	Port Role
33–48	SFP28	10/25/50 Gbps	Gigabit Ethernet	• Server Port • Ethernet/FCoE Uplink Port • FCoE Storage Port • Appliance Port (EHM only) • Monitor Port
49–52	QSFP-DD	40/100/400 Gbps	Gigabit Ethernet	• Ethernet/FCoE Uplink Port • FCoE Storage Port • Appliance Port (EHM only) • Monitor Port

Breakout Guidelines and Limitations

Note:

Breakout ports are supported as destinations for traffic monitoring. To ensure proper configuration of the Cisco UCS 6652, the following guidelines must be observed:

- **Breakout Availability:** Breakout functionality is available **only** for physical ports **49–52** (QSFP-DD). The native 50G SFP28 ports (1–48) do not support subdivision into multiple logical ports.
- **Ethernet vs. Fibre Channel:**
  - **Ethernet:** Breakout is fully supported for Ethernet traffic, resulting in up to 16 logical ports across the four QSFP-DD interfaces.
  - **Fibre Channel (FC):** Breakout is not supported for Fibre Channel. Native FC is only supported on the Unified Ports (17–32), which use the SFP28 interface and cannot be subdivided.
- **Port Roles and Configuration**:

  Physical ports 49–52 and their logical breakout counterparts can be configured as Uplink Ports, FCoE Uplink Ports, FCoE Storage Ports, and Appliance Ports (in End-Host Mode).
- **MACsec Capability**: One of the distinct advantages of the 6652 is that ports 49–52 are driven by MACsec-capable retimers. Consequently, all logical ports derived from a breakout on these ports remain MACsec capable.
- **Traffic Monitoring**: Breakout ports can be utilized as destinations for traffic monitoring through a Switched Port Analyzer (SPAN).

Unified Ports on Cisco UCS 6600 Series Fabric Interconnects

Unified ports are ports on the Cisco UCS 6600 Series Fabric Interconnects that can be configured to carry either Ethernet/FCoE or Fibre Channel traffic.

- **Cisco UCS 6652**: Ports **17 to 32** (16 ports) are unified ports. They support 10/25/50 Gbps Ethernet/FCoE or 16/32/64 Gbps Fibre Channel.
- **Cisco UCS 6664**: Ports **25 to 40** (16 ports) are unified ports. They support 10/25/50 Gbps Ethernet/FCoE or 16/32/64 Gbps Fibre Channel.

Note When you configure a port on a Fabric Interconnect, the administrative state is automatically set to enabled. Configurable beacon LEDs indicate which unified ports are configured for the selected port mode.

Configuring Ethernet Ports on UCS 6600 Series Fabric Interconnects

The Cisco UCS 6600 Series Fabric Interconnect provides the communication backbone for Cisco UCS X/B-Series and C-Series servers, supporting a unified fabric that handles both LAN and SAN traffic.

Before you begin

The Cisco UCS 6600 Series Fabric Interconnect supports a wide range of speeds.

- Verify the transceiver compatibility for the desired speed (10/25/40/50/100/200/400 Gbps).
- Identify the specific port range:
  - **Cisco UCS 6652**: Ports 1-16 and 33-48 (10/25/50G), Ports 17-32 (Unified: 10/25/50G Ethernet or 16/32/64G FC), and Ports 49-52 (40/100/400G).
  - **Cisco UCS 6664**: Ports 1-24 and 41-64 (40/100G, or 10/25G via QSA), Ports 25-40 (Unified: 10/25/50G Ethernet or 16/32/64G FC)

Procedure

	Command or Action	Purpose
Step 1	`scope eth-server or scope eth-uplink`	Use `eth-server` for ports connecting to a chassis or rack server, and `eth-uplink` for ports connecting to the upstream network.
Step 2	`scope fabric {a | b}`	Enter the fabric mode for the specified fabric.
Step 3	`create interface 1 [port_num]`	Create the interface for the specified port. **Note** The slot number for the fixed module on the 6600 series is always 1.
Step 4	`UCS-A /eth-server/fabric* # commit-buffer`	Commit the transaction to the system configuration.

The port is configured and the administrative state is set to enabled. The Ethernet port is successfully configured and initialized. The system will automatically detect the speed based on the inserted transceiver unless manual overrides (like FEC settings) are required.

Example

The following example creates an interface for Ethernet server port 10 on slot 1 of fabric A and commits the transaction:

```
UCS-A# scope eth-server
UCS-A /eth-server # scope fabric a
UCS-A /eth-server/fabric # create interface 1 10
UCS-A /eth-server/fabric* # commit-buffer
```

What to do next

- Verify the operational status of the port.
- If connecting a 25G or 100G link to specific upstream switches (like N9K-C93180YC-FX3), configure the appropriate Forward Error Correction (FEC) settings.

Configuring Fibre Channel Ports on UCS 6600 Series Fabric Interconnects

Unified ports on the 6600 series are configured as Ethernet/FCoE by default. To use these ports for native Fibre Channel (uplinks or storage), you must change the transport type.

- Changing the port mode from Ethernet to Fibre Channel (or vice-versa) is a disruptive operation.
- When you commit these changes, the Fabric Interconnect will automatically reboot to apply the new port configuration.

Before you begin

The 6600 series supports a wide range of Fibre Channel speeds (16/32/64 Gbps).

- Ensure you are using Cisco-qualified Fibre Channel SFPs.
- Identify the unified port range:
  - Cisco UCS 6652: Unified ports are 17 through 32.
  - Cisco UCS 6664: Unified ports are 25 through 40.
- Verify that the Fabric Interconnect is in Fibre Channel switching mode if configuring for FC Storage.

Procedure

	Command or Action	Purpose
Step 1	`UCS-A# scope cabling`	Enter the cabling configuration mode. **Note**

	Command or Action	Purpose
		Cabling mode is used to define the underlying hardware transport (Ethernet vs. Fibre Channel) for unified ports.
Step 2	`scope fabric {a | b}`	Enter the cabling fabric mode for the specified fabric.
Step 3	`UCS-A /cabling/fabric # create breakout 1 [port_num]`	Creates the breakout configuration for the desired unified port. **Note** For the Cisco UCS 6600 series, the slot ID is 1. Port 25 is used in this example for a Cisco UCS 6664 Fabric Interconnect.
Step 4	`UCS-A /cabling/fabric/breakout* # set transport fc`	Set the transport type to Fibre Channel. **Note** This command explicitly converts the port from its default Ethernet state to Fibre Channel.
Step 5	`UCS-A /cabling/fabric/breakout* # commit-buffer`	Commit the transaction to the system configuration. The system may display a warning that the Fabric Interconnect must reboot. Confirm the action to proceed.

The specified unified ports are now configured for Fibre Channel transport. After the reboot, the ports will be available in the FC Uplink or FC Storage scopes.

Example

The following example converts unified port 17 on a UCS 6652 to Fibre Channel mode:

```
UCS-A# scope cabling
UCS-A /cabling # scope fabric a
UCS-A /cabling/fabric # create breakout 1 17
UCS-A /cabling/fabric/breakout* # set transport fc
UCS-A /cabling/fabric/breakout* # commit-buffer
```

What to do next

- After the Fabric Interconnect reboots, scope to `fc-uplink` or `fc-storage` to further configure the port as an Uplink or Storage interface.
- Assign the appropriate VSAN to the newly created Fibre Channel interface.

Unified Ports on the Cisco UCS 6500 Series Fabric Interconnects

Unified ports are ports on the Cisco UCS 6500 Series Fabric Interconnects that you can configure to carry either Ethernet or Fibre Channel traffic. A Cisco UCS domain cannot use these un-reserved ports until you configure them.

Note When you configure a port on a Fabric Interconnect, the administrative state is automatically set to enabled. If the port is connected to another device, this may cause traffic disruption. You can disable the port after configuring it. Configurable beacon LEDs indicate which unified ports are configured for the selected port mode.

Configuring Ethernet Breakout Ports on UCS 6536 Fabric Interconnects

Procedure

Step 1 On the **Equipment** tab, expand **Equipment** > **Fabric Interconnects** > ***Fabric_Interconnect_Name***.

The Fabric Interconnect **General** tab appears, providing at-a-glance status, actions, physical display, properties, and firmware information for the selected fabric interconnect.

Step 2 View the available port(s) to break out.

Ensure that the port overall status is up and admin status is available. Do one of the following:

- In the **Work** pane, click the **Physical Ports** tab. The **Ethernet Ports** and **FC Ports** subtabs appear.
- In the **Work** pane, click the **Physical Display** tab. The Physical Display shows a graphical representation of the base fabric interconnect with a legend to help you identify port admin status.
- In the **Navigation** pane, expand ***Fabric_Interconnect_Name*** > **Fixed Module** > **Ethernet Ports.** this action displays ports in a tree view.

Step 3 Select one or more ports that you can break out. On the UCS 6536 fabric interconnect, ports 1 to 36 support breakout. Do one of the following:

- On the **Physical Display**, click a port or Ctrl-click to select multiple ports.
- On the **Ethernet Ports** tab, click a port or Ctrl-click to select multiple ports.
- On the **Ethernet Ports** tree view, click a port or Ctrl-click to select multiple ports.

Step 4 Configure the selected port(s) as breakout ports.

- On the **Ethernet Ports** tab, right-click the selected port(s) and choose **Configure 4x10G Breakout Port** or **Configure 4x25G Breakout Port** from the pop-up menu.
- On the **Ethernet Ports** tree view, right-click the selected port(s) and choose **Configure 4x10G Breakout Port** or **Configure 4x25G Breakout Port** from the pop-up menu. You can also select ports in the **Ethernet Ports** tree view and select **Configure Breakout Port** from the **Work** pane **Actions** Area. From the drop-down list, choose whether you want to configure the breakout port as a **4x10G port** or a **4x25G** port.

Step 5 Click **OK**.

Step 6 Configure the breakout ports according to your requirements.

Right-click one or more ports and select one of the following options. This table describes the actions that occur when you select the option. If a option is disabled, the port is already configured as such.

Configure Option	Action
Configure as Server Port	You confirm your action. Configuration takes place. The system displays a successful message. Click **Yes**.
Configure as Uplink Port	You confirm your action. Configuration takes place. The system displays a successful message. Click **Yes**.
Configure as FCoE Uplink Port	You confirm your action. Configuration takes place. The system displays a successful message. Click **Yes**.
Configure as FCoE Storage Port	You confirm your action. Configuration takes place. The system displays a successful message. Click **Yes**.
Configure as Appliance Port	You confirm your action. Configuration takes place. The system displays a successful message. Click **Yes**.

Step 7 The confirmation dialog box displays. Click **Yes**.

Note
Ethernet breakout port configuration will not lead to Fabric Interconnect reboot.

Configuring Fibre Channel Breakout Ports

Converting Ethernet Ports to Fibre Channel Breakout Port

You can follow the below steps to configure ethernet ports to Fibre Channel ports using scope cabling:

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope cabling**	Enters the cabling mode.
Step 2	UCS-A /cabling # **scope fabric a**	Enters cabling fabric mode for the specified fabric.
Step 3	UCS-A /cabling/fabric # **create breakout** *1 36*	Creates the breakout port on the selected slot and port.
Step 4	UCS-A /cabling/fabric/breakout* # **set transport fc**	Creates Fibre Channel uplink breakout ports.
Step 5	UCS-A /cabling/fabric/breakout* # **commit-buffer**	Commits the transaction to the server.

Example

The following example creates breakout ports on Cisco UCS 6536 Fabric Interconnect sets the breakout type, and commits the transaction:

```
UCS-A# scope cabling
UCS-A /cabling # scope fabric a
UCS-A /cabling/fabric/breakout* # create breakout 1 36
UCS-A /cabling/fabric/breakout* # set transport fc
```

Note This operation will change port mode from Ethernet to Fibre Channel or vice-versa. When committed, this change will require the switch to reboot.

```
UCS-A /cabling/fabric/breakout* # commit-buffer
```

Converting Ethernet Breakout Port to Fibre Channel Breakout Port

Follow the below commands to convert any existing ethernet breakout ports to Fibre Channel breakout ports using scope cabling:

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope cabling**	Enters the cabling mode.
Step 2	UCS-A /cabling # **scope fabric a**	Enters cabling fabric mode for the specified fabric.
Step 3	UCS-A /cabling/fabric # **show breakout**	Shows the existing breakout ports.
Step 4	UCS-A /fc-uplink/fabric # **scope breakout** *1 36*	Enters the breakout port.
Step 5	UCS-A /fc-uplink/fabric/breakout # **set transport fc**	Creates the FC uplink breakout ports.
Step 6	UCS-A /cabling/fabric/breakout* # **commit-buffer**	Commits the transaction to the server.

Example

The following example creates breakout ports on a UCS 6536 Fabric Interconnect, sets the breakout type, and commits the transaction:

```
UCS-A# scope cabling
UCS-A /cabling # scope fabric a
UCS-A /fc-uplink/fabric # show breakout
port breakout:
    Slot ID    Port ID    breakout type FC breakout type transport type
    ---------- ---------- ------------- ---------------- --------------
             1         36        10g 4x            Unknown           Ether
```

```
UCS-A /cabling/fabric # scope breakout 1 36
UCS-A /fc-uplink/fabric/breakout # set transport fc
```

Note This operation will change port mode from Ethernet to Fibre Channel or vice-versa. When committed, this change will require the switch to reboot.

```
UCS-A /cabling/fabric/breakout* # commit-buffer
```

Converting Ethernet Breakout Port to Fibre Channel Breakout Port Using Fibre Channel Uplink

To create ethernet breakout port through Fibre Channel Uplink, the port should already be in ethernet breakout mode.

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope fc-uplink**	Enters the Fibre Channel uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric a**	Enters cabling fabric mode for the specified fabric.
Step 3	UCS-A /fc-uplink/fabric # **create aggr-interface***slot-idport-id*	Creates the breakout port on the selected slot and port. The Slot ID ranges from 1 through 4 and the Port ID ranges from 36 through 33.
Step 4	UCS-A /fc-uplink/fabric/aggr-interface* # **create br-interface** *slot-id*	Creates the breakout port on the selected port.
Step 5	UCS-A /fc-uplink/fabric/aggr-interface/br-interface* # **up**	Sets the breakout port on the selected slot and port as FC uplink port.
Step 6	Repeat steps 4 and 5 for the remaining *slot-id* from 1 through 4.	
Step 7	UCS-A /fc-uplink/fabric/aggr-interface/br-interface*# **commit-buffer**	Commits the transaction to the server.

Example

The following example creates breakout ports on a UCS 6536 Fabric Interconnect, sets the breakout type, and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # create aggr-interface 1 36
UCS-A /fc-uplink/fabric/aggr-interface* # create br-interface 1
UCS-A /fc-uplink/fabric/aggr-interface/br-interface* # up
UCS-A /fc-uplink/fabric/aggr-interface* # create br-interface 2
UCS-A /fc-uplink/fabric/aggr-interface/br-interface* # up
```

```
UCS-A /fc-uplink/fabric/aggr-interface* # create br-interface 3
UCS-A /fc-uplink/fabric/aggr-interface/br-interface* # up
UCS-A /fc-uplink/fabric/aggr-interface* # create br-interface 4
UCS-A /fc-uplink/fabric/aggr-interface/br-interface* # up
UCS-A /fc-uplink/fabric/aggr-interface/br-interface* # commit-buffer
```

Note You must create all four breakout interfaces to proceed with **commit-buffer**.

Deleting Fibre Channel Breakout Port

Deleting Fibre Channel Breakout Ports

The example described in this topic describes how to delete an entire breakout interface and to convert the port to normal ethernet interface using scope cabling.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope cabling**	Enters the cabling mode.
Step 2	UCS-A# /cabling # **scope fabric**{**a** | **b**}	Enters cabling fabric mode for the specified fabric.
Step 3	UCS-A /cabling/fabric # **delete breakout** *slot-id port-id*	Deletes the breakout for the specified ports. Slot ID ranges from 1 through 4 and Port ID ranges from 33 through 36. **Note** This operation will change the port mode (from Ethernet to FC or vice-versa). When committed, it leads to reboot.
Step 4	UCS-A /cabling/fabric* # **commit-buffer**	Commits the transaction to the server.

What to do next

Verify that you deleted the specified breakout port using the **show** command.

Deleting Fibre Channel Breakout Ports Using Fibre Channel Uplink

You can follow the below steps to delete an entire breakout interface and to convert the port to normal ethernet interface.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters the cabling mode.
Step 2	UCS-A# /fc-uplink # **scope fabric**{**a** | **b**}	Enters cabling fabric mode for the specified fabric.
Step 3	UCS-A /fc-uplink/fabric # **delete aggr-interface** *slot-id port-id*	Deletes the breakout for the specified ports. Slot ID ranges from 1 through 4 and Port ID ranges from 33 through 36. **Note** This operation will change the port mode (from Ethernet to FC or vice-versa). When committed, it leads to reboot.
Step 4	UCS-A /fc-uplink/fabric* # **commit-buffer**	Commits the transaction to the server.

What to do next

Verify that you deleted the specified breakout port using the **show** command.

Appliance Breakout Port

Configuring Breakout Appliance Ports

You can follow the below steps to configure appliance breakout ports for both Cisco UCS Fabric Interconnects 9108 100G, Cisco UCS 6500 Series Fabric Interconnect, , and Cisco UCS 6400 Series Fabric Interconnect:

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-storage**	Enters Ethernet storage mode.
Step 2	UCS-A# /eth-storage # **scope fabric**{**a** | **b**}	Enters Ethernet storage mode for the specified fabric.
Step 3	UCS-A# /eth-storage/fabric # **enter aggr-interface** *slot-numaggregate-port-num*	Enters the interface for the specified aggregate(main) appliance port.
Step 4	UCS-A# /eth-storage/fabric/port-channel/member-aggr-port # **create br -interface***breakout-port-num*	Creates an interface for the specified breakout appliance port.
Step 5	UCS-A# /eth-storage/fabric/port-channel/member-aggr-port/br-member-port # **commit-buffer** **Example:**	Commits the transaction to the server.

	Command or Action	Purpose

	The following example creates an interface for an appliance port 1 of the aggregate port 20 on slot 1 of fabric B, and commits the transaction:
	
	`UCS-A# scope eth-storage`
	`UCS-A /eth-storage # scope fabric a`
	`UCS-A /eth-storage/fabric # enter aggr-interface 1 20`
	`UCS-A /eth-storage/fabric/aggr-interface # create br-interface 1`
	`UCS-A /eth-storage/fabric/aggr-interface/br-interface* # commit-buffer`
	
	Example:
	
	Note
	If the port is only connected to 100G SFP which is broken out in 25x4 breakout port then when creating an appliance port, the default speed for a breakout port would be **Auto**.	

Modifying Speed for Breakout Port of Type 25x4Gbps

Beginning from Cisco UCS Manager release 4.2(3b), you can modify the speed for breakout port of type 25x4Gbps for Cisco UCS Fabric Interconnects 9108 100G, Cisco UCS 6500, and Cisco UCS 6400 Series Fabric Interconnects. The commands are:

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-storage**	Enters Ethernet storage mode.
Step 2	UCS-A# /eth-storage # **scope fabric** {**a** | **b**}	Enters Ethernet storage mode for the specified fabric.
Step 3	UCS-A# /eth-storage/fabric /aggr-interface # **create aggr-interface** *1 28*	Enters the interface for the specified aggregate(main) appliance port.
Step 4	UCS-A# /eth-storage/fabric/aggr-interface # **scope br -interface***1*	Creates an interface for the specified breakout appliance port.
Step 5	UCS-A# /eth-storage/fabric/aggr-interface/br-interface* # **set adminspeed 25gbps**	Modifies the admin speed to 25Gbps.

	Command or Action	Purpose
Step 6	UCS-A# /eth-storage/fabric/aggr-interface/br-interface* # **commit-buffer**	Commits the transaction.

Modifying FEC Value for Breakout Port of Type 25 x 4Gbps

From Release 4.2(3q) onwards, you can modify the FEC value for breakout port of type 25 x 4Gbps. The commands are:

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-storage**	Enters Ethernet storage mode.
Step 2	UCS-A# /eth-storage # **scope fabric** {**a** | **b**}	Enters Ethernet storage mode for the specified fabric.
Step 3	UCS-A# /eth-storage/fabric /aggr-interface # **create aggr-interface** *1 28*	Enters the interface for the specified aggregate(main) appliance port.
Step 4	UCS-A# /eth-storage/fabric/aggr-interface # **scope br -interface***1*	Creates an interface for the specified breakout appliance port.
Step 5	UCS-A# /eth-storage/fabric/aggr-interface/br-interface* # **set fec cl91**	Modifies the FEC value to cl91.
Step 6	UCS-A# /eth-storage/fabric/aggr-interface/br-interface* # **commit-buffer**	Commits the transaction to the server. **Note** You cannot modify FEC value for any other port which is not a 25x4 breakout port. If you modify the speed of a breakout port other than 25x4, it will revert to **Auto**.

Unified Breakout Storage Ports

Converting Fibre Channel Uplink Port to Fibre Channel Storage Port

Follow the below commands to convert Fibre Channel Uplink port to Fibre Channel Storage port using scope cabling:

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope fc-storage**	Enters the Fibre Channel storage mode.

	Command or Action	Purpose
Step 2	UCS-A /fc-storage # **scope fabric a**	Enters cabling fabric mode for the specified fabric.
Step 3	UCS-A /fc-storage/fabric # **create aggr-interface***1 36*	Creates the breakout port on the specified aggregate (main) FC storage port.
Step 4	UCS-A /fc-storage/fabric/aggr-interface* # **create br-interface** br-fc *1*	Creates the breakout port on the selected port.
Step 5	UCS-A /fc-storage/fabric/aggr-interface/br-fc* # **commit-buffer**	Commits the transaction to the server.
Step 6	UCS-A /fc-storage/fabric/aggr-interface/br-fc # **up**	
Step 7	UCS-A /fc-storage/fabric/aggr-interface # **up**	
Step 8	UCS-A /fc-storage/fabric # **show interface fc**	Displays the output.

Example

```
Breakout FC Interface:

Slot ID    Aggr-Port ID Port ID    Admin State Speed         Config State Operational State
 State Reason    Lic State            Grace Prd
---------- ------------ ---------- ----------- ------------- ------------ -----------------
 --------------- -------------------- ---------
         1           36 1          Enabled     16gbps        Inconsistent Sfp Not Present
  FC storage interface unsupported in FC end host mode

                  Not Applicable       0
UCS-A /fc-storage/fabric # scope aggr-interface 1 36
UCS-A /fc-storage/fabric/aggr-interface # show br-interface br-fc

Breakout FC Interface:
    Slot ID    Aggr-Port ID Port ID    Admin State Speed         Config State Operational
State State Reason Lic State            Grace Prd
    ---------- ------------ ---------- ----------- ------------- ------------
----------------- ------------ -------------------- ---------
            1           36 1         Enabled     16gbps        Inconsistent Sfp Not Present
   FC storage interface unsupported in FC end host mode

                   Not Applicable       0
UCS-A /fc-storage/fabric/aggr-interface #
```

Fibre Channel Uplink Breakout Port Channels

Configuring Fibre Channel Uplink Breakout Port Channel and Member Addition

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters Fibre Channel uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric a**	Enters Fibre Channel uplink fabric mode for the specified fabric.
Step 3	UCS-A /fc-uplink/fabric # **create port-channel** *100*	Creates a port channel on the specified Fibre Channel uplink port, and enters Fibre Channel uplink fabric port channel mode.
Step 4	UCS-A /fc-uplink/fabric/port channel* # **create aggr-interface** *1 36*	Creates the interface for the specified aggregate (main) Fibre Channel uplink port.
Step 5	UCS-A /fc-uplink/fabric/port channel/aggr-interface* # **create br-member-port** *2*	Creates the member port for the specified breakout Fibre Channel uplink port.
Step 6	UCS-A /fc-uplink/fabric/port channel/aggr-interface/br-member-port* # **commit-buffer**	Commits the transaction to the system configuration.
Step 7	UCS-A /fc-uplink/fabric/port channel/aggr-interface/br-member-port # **up**	
Step 8	UCS-A /fc-uplink/fabric/port channel/aggr-interface/br-member-port # show br-member-port	Displays the output.

Example

The following is the example for creating and adding a member to a breakout port channel:

```
Breakout Fc Member Port:

Slot Id    Aggr-Port ID  Port Id    Membership         Admin State User Label Oper State
     Speed         Oper Speed    State Reason Lic State            Grace Perio
d            Fill Pattern
---------- ------------ ---------- ------------------ ----------- ---------- ----------------
 ------------ ------------ ------------ -------------------- -----------
------------ ------------
        1 36           2          Down               Enabled                Sfp Not Present
  Auto          Indeterminate SFP not present Not Applicable
             0 Idle
UCS-A /fc-uplink/fabric/port-channel/aggr-interface # up
UCS-A /fc-uplink/fabric/port-channel # show aggr-interface

Aggregate-Interface:
    Slot    Port          Config State Lic State Grace Prd
```

```
    ------- ------------- ------------ --------- ---------
          1 36            Disabled     Unknown   0
UCS-A /fc-uplink/fabric/port-channel # up
UCS-A /fc-uplink/fabric # show interface

Breakout Interface:

Slot Id    Aggr-Port ID  Port Id    Admin State Oper State       Fill Pattern  State Reason
    Speed         Oper Speed    Lic State            Grace Prd
---------- ------------ ---------- ----------- ---------------- ------------- ---------------
 ------------ ------------- -------------------- ---------
         1 34            1          Disabled    Sfp Not Present  Idle          SFP not
present 16gbps        Indeterminate Not Applicable       0
         1 34            2          Disabled    Sfp Not Present  Idle          SFP not
present 16gbps        Indeterminate Not Applicable       0
         1 34            3          Disabled    Sfp Not Present  Idle          SFP not
present 16gbps        Indeterminate Not Applicable       0
         1 34            4          Enabled     Sfp Not Present  Idle          SFP not
present 16gbps        Indeterminate Not Applicable       0
         1 35            1          Enabled     Sfp Not Present  Idle          SFP not
present 32gbps        Indeterminate Not Applicable       0
         1 35            2          Enabled     Sfp Not Present  Idle          SFP not
present 32gbps        Indeterminate Not Applicable       0
         1 35            3          Enabled     Sfp Not Present  Idle          SFP not
present 32gbps        Indeterminate Not Applicable       0
         1 35            4          Enabled     Sfp Not Present  Idle          SFP not
present 32gbps        Indeterminate Not Applicable       0
         1 36            1          Disabled    Sfp Not Present  Idle          SFP not
present 16gbps        Indeterminate Not Applicable       0
         1 36            3          Enabled     Sfp Not Present  Idle          SFP not
present 16gbps        Indeterminate Not Applicable       0
         1 36            4          Enabled     Sfp Not Present  Idle          SFP not
present 16gbps        Indeterminate Not Applicable       0

Breakout Fc Member Port:

Port-channel Slot  AggregatePort Port  Oper State      State Reason
------------ ----- ------------- ----- --------------- ------------
100              1 36            2     Sfp Not Present SFP not present
UCS-A /fc-uplink/fabric #
```

Unified Breakout Ports for Cisco UCS X-Series Direct

Configuring Ethernet Breakout Ports on Cisco UCS Fabric Interconnects 9108 100G

Procedure

Step 1 On the **Equipment** tab, expand **Equipment** > **Fabric Interconnects** > ***Fabric_Interconnect_Name***.

The Fabric Interconnect **General** tab appears, providing at-a-glance status, actions, physical display, properties, and firmware information for the selected fabric interconnect.

Step 2 View the available port(s) to break out.

Ensure that the port overall status is up and admin status is available. Do one of the following:

- In the **Work** pane, click the **Physical Ports** tab. The **Ethernet Ports** and **FC Ports** subtabs appear.
- In the **Work** pane, click the **Physical Display** tab. The Physical Display shows a graphical representation of the base fabric interconnect with a legend to help you identify port admin status.
- In the **Navigation** pane, expand ***Fabric_Interconnect_Name*** > **Fixed Module** > **Ethernet Ports.** this action displays ports in a tree view.

Step 3 Select one or more ports that you can break out. On the Cisco UCS Fabric Interconnects 9108 100G, ports 1 to 8 support breakout. Do one of the following:

- On the **Physical Display**, click a port or Ctrl-click to select multiple ports.
- On the **Ethernet Ports** tab, click a port or Ctrl-click to select multiple ports.
- On the **Ethernet Ports** tree view, click a port or Ctrl-click to select multiple ports.

Step 4 Configure the selected port(s) as breakout ports.

- On the **Ethernet Ports** tab, right-click the selected port(s) and choose **Configure 4x10G Breakout Port** or **Configure 4x25G Breakout Port** from the pop-up menu.
- On the **Ethernet Ports** tree view, right-click the selected port(s) and choose **Configure 4x10G Breakout Port** or **Configure 4x25G Breakout Port** from the pop-up menu. You can also select ports in the **Ethernet Ports** tree view and select **Configure Breakout Port** from the **Work** pane **Actions** Area. From the drop-down list, choose whether you want to configure the breakout port as a **4x10G port** or a **4x25G** port.

Step 5 Click **OK**.

Step 6 Configure the breakout ports according to your requirements.

Right-click one or more ports and select one of the following options. This table describes the actions that occur when you select the option. If a option is disabled, the port is already configured as such.

Configure Option	Action
Configure as Uplink Port	You confirm your action. Configuration takes place. The system displays a successful message. Click **Yes**.
Configure as FCoE Uplink Port	You confirm your action. Configuration takes place. The system displays a successful message. Click **Yes**.
Configure as FCoE Storage Port	You confirm your action. Configuration takes place. The system displays a successful message. Click **Yes**.
Configure as Appliance Port	You confirm your action. Configuration takes place. The system displays a successful message. Click **Yes**.

Note
The **Configure as Server Port** option is supported on Cisco UCS Fabric Interconnects 9108 100G (Cisco UCS X-Series Direct). However, configuring a server port as a breakout port is not supported.

Step 7 The confirmation dialog box displays. Click **Yes**.

Note

Ethernet breakout port configuration will not lead to Fabric Interconnect reboot.

Physical and Backplane Ports

Displaying VIF Port Statistics Obtained From the Adaptor

You can verify the health and throughput of Ethernet vNICs by inspecting VIF statistics. Starting with Release 6.0(2), the CLI provides real-time traffic rates, which are essential for troubleshooting network congestion and micro-bursts.

Procedure

	Command or Action	Purpose
Step 1	UCS-A /fabric-interconnect # **connect nxos** {**a** | **b**}	Enters NX-OS mode for the specified fabric interconnect.
Step 2	UCS-A(nxos)# **show interface vethernet** *veth-id*	Displays the operational status, bound physical interface, and real-time traffic rates. The output includes throughput measured in Gbps/Mbps and packets per second (pps).

Example

The following example shows how to display real-time traffic rates for a 25 Gbps vNIC (veth 1130):

```
UCS-A /fabric-interconnect # connect nxos a
UCS-A(nxos)# show interface vethernet 1130
Vethernet1130 is up
Bound Interface is port-channel1327
Port description is server 1/3, VNIC vnic2
Hardware is Virtual, address is 0000.abcd.dcba
Speed is 25 Gb/s
input rate 22.42 Gbps, 1.83 Mpps
output rate 38.15 Mbps, 55.10 Kpps
```

Verifying Status of Backplane Ports

Procedure

	Command or Action	Purpose
Step 1	UCS-A /fabric-interconnect # **connect nxos** {**a** | **b**}	Enters NX-OS mode for the fabric interconnect.

	Command or Action	Purpose
Step 2	UCS-A(nxos)# **show interface br**	Displays the configuration of the interface, including the speed and status of the backplane ports.

Example

The following example shows how to verify the status of backplane ports for fabric interconnect A:

```
UCS-A /fabric-interconnect # connect nxos a
UCS-A(nxos)# show interface br
```

```
--------------------------------------------------------------------------------
Ethernet      VLAN   Type Mode   Status  Reason                   Speed     Port
Interface                                                                  Ch #
--------------------------------------------------------------------------------
Eth1/1        1       eth  access down    SFP not inserted           40G(D) --
Eth1/2        1       eth  access down    SFP not inserted           40G(D) --
Br-Eth1/3/1   1       eth  access down    Administratively down      10G(D) --
Br-Eth1/3/2   1       eth  access down    Administratively down      10G(D) --
Br-Eth1/3/3   1       eth  access down    Administratively down      10G(D) --
Br-Eth1/3/4   1       eth  access down    Administratively down      10G(D) --
Eth1/4        1       eth  access down    SFP not inserted           40G(D) --
Br-Eth1/5/1   4044    eth  trunk  down    Link not connected         10G(D) --
Br-Eth1/5/2   4044    eth  trunk  down    Link not connected         10G(D) --
Br-Eth1/5/3   4044    eth  trunk  down    Link not connected         10G(D) --
Br-Eth1/5/4   4044    eth  trunk  down    Link not connected         10G(D) --
Eth1/6        1       eth  access down    SFP not inserted           40G(D) --
Eth1/7        1       eth  access down    SFP not inserted           40G(D) --
Eth1/8        1       eth  access down    SFP not inserted           40G(D) --
Eth1/9        1       eth  access down    SFP not inserted           40G(D) --
Eth1/10       1       eth  access down    SFP not inserted           40G(D) --
Eth1/11       1       eth  fabric up      none                       40G(D) --
Eth1/12       1       eth  access down    SFP not inserted           40G(D) --
Eth1/13       1       eth  access down    SFP not inserted           40G(D) --
Eth1/14       1       eth  access down    SFP not inserted           40G(D) --
Eth1/15       1       eth  access down    SFP not inserted           40G(D) --
Eth1/16       1       eth  access down    SFP not inserted           40G(D) --
Eth1/17       1       eth  access down    SFP not inserted           40G(D) --
Eth1/18       1       eth  access down    SFP not inserted           40G(D) --
Eth1/19       1       eth  access down    SFP not inserted           40G(D) --
Eth1/20       1       eth  access down    SFP not inserted           40G(D) --
Br-Eth1/21/1  1       eth  trunk  up      none                       10G(D) --
Br-Eth1/21/2  1       eth  trunk  up      none                       10G(D) --
Br-Eth1/21/3  1       eth  trunk  down    Link not connected         10G(D) --
Br-Eth1/21/4  1       eth  trunk  up      none                       10G(D) --
Eth1/22       1       eth  access down    SFP not inserted           40G(D) --
Eth1/23       1       eth  access down    SFP not inserted           40G(D) --
Eth1/24       1       eth  access down    SFP not inserted           40G(D) --
Eth1/25       1       eth  access down    SFP not inserted           40G(D) --
Eth1/26       1       eth  access down    SFP not inserted           40G(D) --
Eth1/27       1       eth  access down    SFP not inserted           40G(D) --
Eth1/28       1       eth  access down    SFP not inserted           40G(D) --
Eth1/29       1       eth  access down    SFP not inserted           40G(D) --
Eth1/30       1       eth  access down    SFP not inserted           40G(D) --
Eth1/31       1       eth  access down    SFP not inserted           40G(D) --
Eth1/32       1       eth  access down    SFP not inserted           40G(D) --
```

```
--------------------------------------------------------------------------------
Port-channel VLAN    Type Mode   Status  Reason                    Speed   Protocol
Interface
--------------------------------------------------------------------------------
Po1285       1       eth  vntag  up      none                       a-10G(D)  none
Po1286       1       eth  vntag  up      none                       a-10G(D)  none
Po1287       1       eth  vntag  up      none                       a-10G(D)  none
Po1288       1       eth  vntag  up      none                       a-10G(D)  none
Po1289       1       eth  vntag  up      none                       a-10G(D)  none

--------------------------------------------------------------------------------
Port   VRF          Status IP Address                              Speed    MTU
--------------------------------------------------------------------------------
mgmt0  --           down   10.197.157.252                         --       1500

--------------------------------------------------------------------------------
Vethernet     VLAN   Type Mode   Status  Reason                    Speed
--------------------------------------------------------------------------------
Veth691       4047   virt trunk  down    nonParticipating         auto
Veth692       4047   virt trunk  up      none                     auto
Veth693       1      virt trunk  down    nonParticipating         auto
Veth695       1      virt trunk  up      none                     auto
Veth699       1      virt trunk  up      none                     auto

--------------------------------------------------------------------------------
Interface Secondary VLAN(Type)                    Status Reason
--------------------------------------------------------------------------------
Vlan1     --                                      down   Administratively down

--------------------------------------------------------------------------------
Ethernet       VLAN   Type Mode   Status  Reason                  Speed     Port
Interface                                                                   Ch #
--------------------------------------------------------------------------------
Eth1/1/1       1       eth  vntag  up      none                      10G(D) 1286
Eth1/1/2       1       eth  access down    Administratively down     10G(D) --
Eth1/1/3       1       eth  vntag  up      none                      10G(D) 1286
Eth1/1/4       1       eth  access down    Administratively down     10G(D) --
Eth1/1/5       1       eth  vntag  up      none                      10G(D) 1287
Eth1/1/6       1       eth  access down    Administratively down     10G(D) --
Eth1/1/7       1       eth  vntag  up      none                      10G(D) 1287
Eth1/1/8       1       eth  access down    Administratively down     10G(D) --
Eth1/1/9       1       eth  vntag  up      none                      10G(D) 1289
Eth1/1/10      1       eth  access down    Administratively down     10G(D) --
Eth1/1/11      1       eth  vntag  up      none                      10G(D) 1289
Eth1/1/12      1       eth  access down    Administratively down     10G(D) --
Eth1/1/13      1       eth  vntag  up      none                      10G(D) 1285
Eth1/1/14      1       eth  access down    Administratively down     10G(D) --
Eth1/1/15      1       eth  vntag  up      none                      10G(D) 1285
Eth1/1/16      1       eth  access down    Administratively down     10G(D) --
Eth1/1/17      1       eth  access down    Administratively down     10G(D) --
Eth1/1/18      1       eth  vntag  up      none                      10G(D) 1288
Eth1/1/19      1       eth  access down    Administratively down     10G(D) --
Eth1/1/20      1       eth  vntag  up      none                      10G(D) 1288
Eth1/1/21      1       eth  access down    Administratively down     10G(D) --
Eth1/1/22      1       eth  access down    Administratively down     10G(D) --
Eth1/1/23      1       eth  access down    Administratively down     10G(D) --
Eth1/1/24      1       eth  access down    Administratively down     10G(D) --
Eth1/1/25      1       eth  access down    Administratively down     10G(D) --
Eth1/1/26      1       eth  access down    Administratively down     10G(D) --
Eth1/1/27      1       eth  access down    Administratively down     10G(D) --
Eth1/1/28      1       eth  access down    Administratively down     10G(D) --
Eth1/1/29      1       eth  access down    Administratively down     10G(D) --
Eth1/1/30      1       eth  access down    Administratively down     10G(D) --
Eth1/1/31      1       eth  access down    Administratively down     10G(D) --
```

```
Eth1/1/32      1       eth  access down    Administratively down      10G(D) --
Eth1/1/33      4044    eth  trunk  up      none                      1000(D) --
```

Verifying VFC to Vethernet Binding and Real-Time Statistics

Starting with Cisco UCS Manager 6.0(2b) release, you can display *real-time traffic rates* for virtual interfaces (vNICs and vHBAs) through the NX-OS shell. This provides visibility into throughput measured in *Gbps* and *packets per second (pps)*, which is essential for identifying micro-bursts and network congestion.

Because *VFC interfaces* are encapsulated within *Vethernet interfaces*, any congestion or drops at the Ethernet layer will directly impact *FCoE storage performance*.

Before you begin

Before you begin, ensure the following:

- You must have administrative privileges to access the ***NX-OS shell***.
- The Fabric Interconnect is running Release 6.0(2b) or later to support real-time traffic rate reporting.

Procedure

	Command or Action	Purpose
Step 1	Enter NX-OS mode for the specified fabric interconnect. **Example:** `UCS-A# connect nxos {a | b}`	
Step 2	Identify the Vethernet interface bound to the VFC. Look for the *Bound interface* field in the output. **Example:** `UCS-A(nxos)# show interface vfc [vfc_id]`	Look for the `Bound interface` field in the output to determine which Vethernet ID to inspect.
Step 3	Display the real-time traffic rates for the bound Vethernet interface. **Example:** `UCS-A(nxos)# show interface vethernet [veth_id]`	The output includes throughput in *Gbps* and *packets per second (pps)*. The system displays the traffic rate in Gbps and packets per second (pps).

You have successfully verified the traffic rates for the VFC and its underlying Ethernet transport layer, enabling effective troubleshooting of network congestion.

Example

Example 1: Displaying Real-Time Ethernet vNIC Statistics

The following example shows the *input rate* and *output rate* for a 25 Gbps Ethernet vNIC (`veth1130`):

```
UCS-A(nx-os)# sh int veth 1130
Vethernet1130 is up
Bound Interface is port-channel1327
Port description is server 1/3, VNIC vnic2
Hardware is Virtual, address is 0000.abcd.dcba
Port mode is trunk
Speed is 25 Gb/s
Duplex mode is full
Polling interval: 5 seconds
input rate 22.42 Gbps, 1.83 Mpps
output rate 38.15 Mbps, 55.10 Kpps
Rx
253175496 unicast packets  137 multicast packets  162 broadcast packets
253175795 input packets  383305412813 bytes
0 input packet drops
Tx
7362977 unicast packets  286 multicast packets  6585 broadcast packets
7369848 output packets  492853045 bytes
0 flood packets
0 output packet drops
```

Example 2: Displaying Real-Time FC vHBA Statistics

The following example shows the *FCoE input/output rates* for a vHBA (`vfc1134`):

```
UCS-A(nx-os)# show interface vfc 1134
vfc1134 is trunking
Bound interface is Vethernet9326
Port description is server 1/3, VHBA C1B3-vhba-A
Hardware is Ethernet
Port WWN is 24:6d:00:08:31:54:67:89
Admin port mode is F, trunk mode is on
snmp link state traps are enabled
Port mode is TF
Port vsan is 600
Operating Speed is 25 Gbps
Admin Speed is auto
Trunk vsans (admin allowed and active)  (600)
Trunk vsans (up)                        (600)
Trunk vsans (isolated)                  ()
Trunk vsans (initializing)              ()
Polling interval: 5 seconds
91112433 fcoe in packets
189929511828 fcoe in octets
92386242 fcoe out packets
191899291796 fcoe out octets
13.08 Gbps, 780.59 Kpps fcoe input rate
12.91 Gbps, 773.17 Kpps fcoe output rate
Interface last changed at Thu Jan 22 16:32:50 2026
```

Example 3: Displaying Statistics for the Bound Vethernet Interface

In FCoE configurations, storage traffic is encapsulated in an underlying Ethernet layer. To troubleshoot storage performance, you can inspect the *bound Vethernet interface* (`veth9326`) of the VFC:

```
UCS-A(nx-os)# show interface vethernet 9326
Vethernet9326 is up
    Bound Interface is port-channel1327
    Port description is server 1/3, VHBA C1B3-vhba-A
    Speed is 25 Gb/s
  Polling interval: 5 seconds
```

```
  input rate 13.11 Gbps, 781.65 Kpps
  output rate 13.17 Gbps, 788.43 Kpps
Rx
  98945041 unicast packets  0 multicast packets  0 broadcast packets
  98945041 input packets  206257149672 bytes
  0 input packet drops
```

Note The *VFC interfaces* are encapsulated within *Vethernet interfaces*, any congestion or drops at the Ethernet layer (Vethernet) will directly impact *FCoE storage performance*.

Server Ports

Automatic Configuration of Fabric Interconnect Server Ports

Starting with Cisco UCS Manager release 3.1(3), you can automatically configure the fabric interconnect server ports. The server **Port Auto-Discovery Policy** determines how the system reacts when a new rack server, chassis, or FEX is added. By enabling this policy, Cisco UCS Manager automatically determines the type of device connected to the switch port and configures the switch port accordingly.

Note

- If you do not want a Cisco UCS C-Series appliance to be UCS Managed, pre-configure the appliance ports before connecting VIC ports to the Cisco UCS Fabric Interconnects.
- The **Port Auto-Discovery Policy** is not applicable for servers connected through direct 25G port or 4x25g breakout on Cisco UCS 6454, UCS 64108, and 6536 Fabric Interconnects.

Automatically Configuring Server Ports

Procedure

Step 1 UCS-A# **scope org/**

Enters the root organization mode.

Step 2 UCS-A / org# **scope por**

Enters organization port discovery policy mode.

Step 3 UCS-A / org / port-disc-policy# **set descr**

Provides a description for the port discovery policy.

Step 4 UCS-A / org / port-disc-policy# **set server-auto-disc**

Enables port auto-discovery.

Note

By default `server-auto-disc` is disabled. Port auto-discovery is triggered by enabling `server-auto-disc`.

Example

The following example shows how to enable automatic configuration of fabric interconnect server ports:

```
UCS-A# scope org/
UCS-A /org# scope por
UCS-A / org / port-disc-policy #  set descr
UCS-A / org / port-disc-policy #  set server-auto-disc
```

Configuring a Server Port

All of the port types listed are configurable on both the fixed and expansion module.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-server**	Enters Ethernet server mode.
Step 2	UCS-A /eth-server # **scope fabric** {**a** | **b**}	Enters Ethernet server fabric mode for the specified fabric.
Step 3	UCS-A /eth-server/fabric # **create interface** *slot-num port-num*	Creates an interface for the specified Ethernet server port.
Step 4	UCS-A /eth-server/fabric # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to create an interface for Ethernet server port 4 on slot 1 of fabric B and commit the transaction:

```
UCS-A# scope eth-server
UCS-A /eth-server # scope fabric b
UCS-A /eth-server/fabric # create interface 1 4
UCS-A /eth-server/fabric* # commit-buffer
UCS-A /eth-server/fabric #
```

Unconfiguring a Server Port

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-server**	Enters Ethernet server mode.
Step 2	UCS-A /eth-server # **scope fabric** {**a** | **b**}	Enters Ethernet server fabric mode for the specified fabric.
Step 3	UCS-A /eth-server/fabric # **delete interface** *slot-num port-num*	Deletes the interface for the specified Ethernet server port.
Step 4	UCS-A /eth-server/fabric # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example unconfigures Ethernet server port 12 on slot 1 of fabric B and commits the transaction:

```
UCS-A# scope eth-server
UCS-A /eth-server # scope fabric b
UCS-A /eth-server/fabric # delete interface 1 12
UCS-A /eth-server/fabric* # commit-buffer
UCS-A /eth-server/fabric #
```

Configuring a Server Port for Forward Error Correction

- **Cisco UCS 6400 Series Fabric Interconnects:** To establish a 25Gbps link, the server port must be configured with **FEC Cl74**. This configuration is specific to connections between the N9K-C93180YC-FX3 and UCS 6400 Series Fabric Interconnects.
- **Cisco UCS 6500 Series Fabric Interconnects:** For 100Gbps links, the server port requires **FEC Cl91**.
- **Cisco UCS 6600 Series Fabric Interconnects:** For 25Gbps server ports connecting to the N9K-C93180YC-FX3, **only FEC Cl74 is supported**. Using *Auto* or any other FEC option is unsupported. For 100Gbps server ports, the FEC must be set to **Auto**.

Note The CL-74 configuration is not applicable for other server port connectivity such as I/O module or direct-attached rack server.

Table 4: FEC CL-74 Support Matrix

Port Speed	FEC CL-74
1 Gbps	Not supported

Port Speed	FEC CL-74
10 Gbps	Not supported
25 Gbps	Supported
40 Gbps	Not supported
100 Gbps	Supported
Auto	Based on inserted transceiver's maximum supported speed

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-server**	Enters Server mode.
Step 2	UCS-A /eth-server # **scope fabric** {**a** | **b**}	Enters Server mode for the specified fabric.
Step 3	UCS-A /eth-server/fabric/interface # **scope interface** *slot-id port-id*	Enters Server interface mode for the specified interface.
Step 4	UCS-A /eth-server/fabric/interface # **set fec** {**auto** | **cl74** | **cl91**}	Sets the forward error correction setting as **auto** or **cl74** or **cl91** for the server port.
Step 5	UCS-A /eth-server/fabric/interface # **set auto-neg** {**enabled** | **disabled**}	Sets the auto negotiate as enabled or disabled for the server port.
Step 6	UCS-A /eth-server/fabric/interface # **commit-buffer**	Commits the transaction to the system configuration. **Note** Following are the mandatory configuration parameters on the server port for connecting to N9K-C93180YC-FX3: • The FEC must be **auto** for 100Gps server port. • The FEC must be **cl74** for 25Gps server port. • The auto-negotiation must be **disabled** for 100Gps server port.

Example

Example 1: The following example shows how to enable forward error correction cl74 with auto-negotiation enabled, on an interface for the 25Gps server port 15 on slot 2 of fabric A, and commit the transaction:

```
UCS-A# scope eth-server
UCS-A /eth-server # scope fabric a
UCS-A /eth-server/fabric # scope interface 2 15
UCS-A /eth-server/fabric # set fec cl74
UCS-A /eth-server/fabric/interface # set auto-neg enabled
UCS-A /eth-server/fabric* # commit-buffer
UCS-A /eth-server/fabric #
```

Example 2: The following example shows how to enable forward error correction auto with auto-negotiation disabled, on an interface for the 100 Gps server port 17 on slot 1 of fabric A, and commit the transaction:

```
UCS-A# scope eth-server
UCS-A /eth-server # scope fabric a
UCS-A /eth-server/fabric # scope interface 1 17
UCS-A /eth-server/fabric # set fec auto
UCS-A /eth-server/fabric/interface # set auto-neg disabled
UCS-A /eth-server/fabric* # commit-buffer
UCS-A /eth-server/fabric #
```

Uplink Ethernet Ports

Configuring an Uplink Ethernet Port

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric a** | **b**}	Enters Ethernet uplink fabric mode for the specified fabric.
Step 3	UCS-A /eth-uplink/fabric # **create interface** *slot-num port-num*	Creates an interface for the specified Ethernet uplink port.
Step 4	(Optional) UCS-A /eth-uplink/fabric # **set speed** {**10gbps** | **1gbps**}	Sets the speed for the specified Ethernet uplink port.
Step 5	UCS-A /eth-uplink/fabric # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to create an interface for Ethernet uplink port 3 on slot 2 of fabric B, set the speed to 10 gbps, and commit the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric b
UCS-A /eth-uplink/fabric # create interface 2 3
UCS-A /eth-uplink/fabric # set speed 10gbps
```

```
UCS-A /eth-uplink/fabric* # commit-buffer
UCS-A /eth-uplink/fabric #
```

Unconfiguring an Uplink Ethernet Port

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric** {**a** | **b**}	Enters Ethernet uplink fabric mode for the specified fabric.
Step 3	UCS-A /eth-uplink/fabric # **delete interface** *slot-num port-num*	Deletes the interface for the specified Ethernet uplink port.
Step 4	UCS-A /eth-uplink/fabric # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example unconfigures Ethernet uplink port 3 on slot 2 of fabric B and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric b
UCS-A /eth-uplink/fabric # delete interface 2 3
UCS-A /eth-uplink/fabric* # commit-buffer
UCS-A /eth-uplink/fabric #
```

Configuring an Uplink Ethernet Port for Forward Error Correction

You can configure forward error correction (FEC) for uplink Ethernet ports, Ethernet appliances, and FCoE uplinks for transceiver modules that operate at 25 Gbps and 100 Gbps speeds that support this feature.

Table 5: Supported Port Speed and FEC Matrix

Port Speed	FEC CL-74	FEC CL-91	RS Cons 16	RS 1eee
1 Gbps	Not supported	Not supported	-	-
10 Gbps	Not supported	Not supported	Not supported	Not supported
25 Gbps	Supported	Supported	Supported	Supported
40 Gbps	Not supported	Not supported	Not supported	Not supported
100 Gbps	Not supported	Supported	Not supported	Not supported

Port Speed	FEC CL-74	FEC CL-91	RS Cons 16	RS 1eee
Auto	Automatically selects the optimal FEC mode based on the transceiver's maximum supported speed.	Automatically selects the optimal FEC mode based on the transceiver's maximum supported speed.	Automatically selects the optimal FEC mode based on the transceiver's maximum supported speed.	Automatically selects the optimal FEC mode based on the transceiver's maximum supported speed.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric a** | **b**}	Enters Ethernet uplink fabric mode for the specified fabric.
Step 3	UCS-A /eth-uplink/fabric # **scope interface** *slot-id port-id*	Enters Ethernet interface mode for the specified interface.
Step 4	Required: UCS-A /eth-uplink/fabric # **set fec** {**auto** |**cl74** | **cl91rs-cons16** | **rs-1eee**}	Sets the forward error correction setting. For the Cisco UCS 6400 and 6500 series Fabric Interconnects, forward error correction is only configurable for 25 Gbps or 100 Gbps port speed.
Step 5	UCS-A /eth-uplink/fabric # **commit-buffer**	Commits the transaction to the system configuration.

Q-in-Q Forwarding

QinQ is defined by IEEE 802.1ad. QinQ is also known as 802.1Q-in-802.1Q that helps to expand the VLAN space through the addition of 802.1Q tag to 802.1Q-tagged packets. This expansion is also termed as VLAN stacking or double VLAN.

In general, the QinQ packets have a standard format. In a VLAN stacking, one 802.11Q tagged packet is encapsulated in another 802.1Q tag. During transmission, packets are forwarded on the outer VLAN tag on the public network and on the inner VLAN tag for private network.

Note The 802.1Q supports 4096 VLANs.

Configuring Q-in-Q Forwarding

You can configure Q-in-Q forwarding for Cisco UCS Fabric Interconnects.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **set q-in-q-forwarding enabled**	Enables Q-in-Q forwarding for the specified fabric.
Step 3	UCS-A /eth-uplink/fabric # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to enable Q-in-Q forwarding and commit the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # set q-in-q-forwarding enabled
UCS-A /eth-uplink/fabric* # commit-buffer
UCS-A /eth-uplink/fabric #
```

Unconfiguring Q-in-Q Forwarding

You can unconfigure Q-in-Q forwarding for Cisco UCS Fabric Interconnects 9108 100G, Cisco UCS 6536 Fabric Interconnect and Cisco UCS 6400 Series Fabric Interconnect.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **set q-in-q-forwarding disabled**	Disables Q-in-Q forwarding for the specified fabric.
Step 3	UCS-A /eth-uplink/fabric # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to disable Q-in-Q forwarding and commit the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # set q-in-q-forwarding disabled
UCS-A /eth-uplink/fabric* # commit-buffer
UCS-A /eth-uplink/fabric #
```

STP Faults

Spanning Tree Protocol (STP) is key for maintaining network stability by actively preventing harmful network loops. The **STP Faults** setting provides direct control over how your system reports issues related to STP. This configuration allows you to either enable or disable the generation of alerts for STP-related faults.

This granular control over STP fault reporting is vital for fine-tuning network monitoring, enabling you to streamline alert management and focus attention specifically on critical network events, thereby contributing to a more stable and manageable network environment.

Note The following occurs when you change the STP fault reporting status:

- **Enabled** to **Disabled**—The system clears all existing STP-related faults.
- **Disabled** to **Enabled**—The system begins raising new STP-related faults.

Configuring STP Faults

You can configure Cisco UCS Manager to report faults related to the Spanning Tree Protocol (STP). Enabling STP fault reporting allows the system to raise alarms when STP-related issues, such as topology changes or port inconsistencies, are detected on Ethernet uplinks.

To enable STP Fault reporting, use the following:

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **set stp-faults enabled**	Enables the raising of STP-related faults.
Step 3	UCS-A /eth-uplink/fabric # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example enters Ethernet uplink mode, enables STP fault reporting, commits the transaction, and verifies the configuration:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # set stp-faults enabled
UCS-A /eth-uplink/fabric* # commit-buffer
UCS-A /eth-uplink/fabric # show detail
-----------
Ethernet Uplink:
    Mode: End Host
    MAC Table Aging Time (dd:hh:mm:ss): Mode Default
    VLAN Port Count Optimization: Enabled
    Fabric Port Channel vHBA reset: Disabled
    service for unsupported transceivers: Disabled
    Q-in-Q Forwarding: Disabled
```

```
    Enable STP faults: Enabled
    Current Task:
```

Unconfiguring STP Faults

You can unconfigure the Spanning Tree Protocol (STP)-related fault reporting in Cisco UCS Manager. When you disable the STP, the system stops raising alarms for STP-related issues on Ethernet uplinks.

Note Disabling this setting clears all existing STP-related faults from the system.

To disable STP Fault reporting:

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **set stp-faults disabled**	Disables STP-related fault reporting.
Step 3	UCS-A /eth-uplink/fabric # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example enters Ethernet uplink mode, disables STP fault reporting, commits the transaction, and verifies the configuration:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # set stp-faults disabled
UCS-A /eth-uplink* # commit-buffer
UCS-A /eth-uplink # show detail
------------------------
Ethernet Uplink:
    Mode: End Host
    MAC Table Aging Time (dd:hh:mm:ss): Mode Default
    VLAN Port Count Optimization: Enabled
    Fabric Port Channel vHBA reset: Disabled
    service for unsupported transceivers: Disabled
    Q-in-Q Forwarding: Disabled
    Enable STP faults: Disabled
    Current Task:
```

Appliance Ports

Appliance ports are only used to connect fabric interconnects to directly attached NFS storage.

Note When you create a new appliance VLAN, its IEEE VLAN ID is not added to the LAN Cloud. Therefore, appliance ports that are configured with the new VLAN remain down, by default, due to a pinning failure. To bring up these appliance ports, you have to configure a VLAN in the LAN Cloud with the same IEEE VLAN ID.

Cisco UCS Manager supports up to four appliance ports per fabric interconnect.

Configuring an Appliance Port

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-storage**	Enters Ethernet storage mode.
Step 2	UCS-A /eth-storage # **scope fabric**{**a** | **b**}	Enters Ethernet storage mode for the specified fabric.
Step 3	UCS-A /eth-storage/fabric # **create interface** *slot-num port-num*	Creates an interface for the specified appliance port.
Step 4	(Optional) UCS-A /eth-storage/fabric/interface # **set portmode** {**access** | **trunk**}	Specifies whether the port mode is access or trunk. By default, the mode is set to trunk. **Note** If traffic for the appliance port needs to traverse the uplink ports, you must also define each VLAN used by this port in the LAN cloud. For example, you need the traffic to traverse the uplink ports if the storage is also used by other servers, or if you want to ensure that traffic fails over to the secondary fabric interconnect if the storage controller for the primary fabric interconnect fails.
Step 5	(Optional) UCS-A /eth-storage/fabric/interface # **set pingroupname** *pin-group name*	Specifies the appliance pin target to the specified fabric and port, or fabric and port channel.
Step 6	(Optional) UCS-A /eth-storage/fabric/interface # **set prio** *sys-class-name*	Specifies the QoS class for the appliance port. By default, the priority is set to best-effort. The sys-class-name argument can be one of the following class keywords: • **FC**—Use this priority for QoS policies that control vHBA traffic only. • **Platinum**—Use this priority for QoS policies that control vNIC traffic only.

	Command or Action	Purpose
		• **Gold**—Use this priority for QoS policies that control vNIC traffic only. • **Silver**—Use this priority for QoS policies that control vNIC traffic only. • **Bronze**—Use this priority for QoS policies that control vNIC traffic only. • **Best Effort**—Do not use this priority. It is reserved for the Basic Ethernet traffic lane. If you assign this priority to a QoS policy and configure another system class as CoS 0, Cisco UCS Manager does not default to this system class. It defaults to the priority with CoS 0 for that traffic.
Step 7	(Optional) UCS-A /eth-storage/fabric/interface # **set adminspeed** {**10gbps** | **1 gbps**}	Specifies the admin speed for the interface. By default, the admin speed is set to 10gbps.
Step 8	UCS-A /eth-storage/fabric/interface # **commit buffer**	Commits the transaction to the system configuration.

Example

The following example creates an interface for an appliance port 2 on slot 3 of fabric B, sets the port mode to access, pins the appliance port to a pin group called pingroup1, sets the QoS class to fc, sets the admin speed to 10 gbps, and commits the transaction:

```
UCS-A# scope eth-storage
UCS-A /eth-storage # scope fabric b
UCS-A /eth-storage/fabric # create interface 3 2
UCS-A /eth-storage/fabric* # set portmode access
UCS-A /eth-storage/fabric* # set pingroupname pingroup1
UCS-A /eth-storage/fabric* # set prio fc
UCS-A /eth-storage/fabric* # set adminspeed 10gbps
UCS-A /eth-storage/fabric* # commit-buffer
UCS-A /eth-storage/fabric #
```

What to do next

Assign a VLAN or target MAC address for the appliance port.

Assigning a Target MAC Address to an Appliance Port or Appliance Port Channel

The following procedure assigns a target MAC address to an appliance port. To assign a target MAC address to an appliance port channel, scope to the port channel instead of the interface.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-storage**	Enters Ethernet storage mode.
Step 2	UCS-A /eth-storage # **scope fabric**{**a** | **b**}	Enters Ethernet storage mode for the specified fabric.
Step 3	UCS-A /eth-storage/fabric # **scope interface** *slot-id port-id*	Enters Ethernet interface mode for the specified interface. **Note** To assign a target MAC address to an appliance port channel, use the **scope port-channel** command instead of **scope interface** .
Step 4	UCS-A /eth-storage/fabric/interface # **create eth-target** *eth-target name*	Specifies the name for the specified MAC address target.
Step 5	UCS-A /eth-storage/fabric/interface/eth-target # **set mac-address** *mac-address*	Specifies the MAC address in nn:nn:nn:nn:nn:nn format.

Example

The following example assigns a target MAC address for an appliance device on port 3, slot 2 of fabric B and commits the transaction:

```
UCS-A# scope eth-storage
UCS-A /eth-storage* # scope fabric b
UCS-A /eth-storage/fabric* # scope interface 2 3
UCS-A /eth-storage/fabric/interface* # create eth-target macname
UCS-A /eth-storage/fabric/interface* # set mac-address 01:23:45:67:89:ab
UCS-A /eth-storage/fabric/interface* # commit-buffer
UCS-A /eth-storage/fabric #
```

The following example assigns a target MAC address for appliance devices on port channel 13 of fabric B and commits the transaction:

```
UCS-A# scope eth-storage
UCS-A /eth-storage* # scope fabric b
UCS-A /eth-storage/fabric* # scope port-channel 13
UCS-A /eth-storage/fabric/port-channel* # create eth-target macname
UCS-A /eth-storage/fabric/port-channel* # set mac-address 01:23:45:67:89:ab
UCS-A /eth-storage/fabric/port-channel* # commit-buffer
UCS-A /eth-storage/fabric #
```

Creating an Appliance Port

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-storage**	Enters Ethernet storage mode.
Step 2	UCS-A/eth-storage# **create vlan** *vlan-name vlan-id*	Creates a named VLAN, specifies the VLAN name and VLAN ID, and enters Ethernet storage VLAN mode.
Step 3	UCS-A/eth-storage/vlan# **set sharing primary**	Saves the changes.
Step 4	UCS-A/eth-storage/vlan# **commit buffer**	Commits the transaction to the system configuration.
Step 5	UCS-A/eth-storage# **create vlan** *vlan-name vlan-id*	Creates a named VLAN, specifies the VLAN name and VLAN ID, and enters Ethernet storage VLAN mode .
Step 6	UCS-A/eth-storage/vlan# **set sharing community**	Associates the primary VLAN to the secondary VLAN that you are creating.
Step 7	UCS-A/eth-storage/vlan# **set pubnwname** *primary vlan-name*	Specifies the primary VLAN to be associated with this secondary VLAN.
Step 8	UCS-A/eth-storage/vlan# **commit buffer**	Commits the transaction to the system configuration.

Example

The following example creates an appliance port:

```
UCS-A# scope eth-storage
UCS-A/eth-storage# create vlan PRI600 600
UCS-A/eth-storage/vlan* # set sharing primary
UCS-A/eth-storage/vlan* # commit-buffer
UCS-A/eth-storage # create vlan COM602 602
UCS-A/eth-storage/vlan* # set sharing isolated
UCS-A/eth-storage/vlan* # set pubnwname PRI600
UCS-A/eth-storage/vlan* # commit-buffer
```

Mapping an Appliance Port to a Community VLAN

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-storage**	Enters Ethernet storage mode.

	Command or Action	Purpose
Step 2	UCS-A/eth-storage# **scope fabric** {*a* | *b*}	Enters Ethernet storage fabric interconnect mode for the specified fabric interconnect.
Step 3	UCS-A/eth-storage/fabric# **create interface** *slot-num port-num*	Creates an interface for the specified Ethernet server port.
Step 4	UCS-A/eth-storage/fabric/interface# **exit**	Exits from the interface. **Note** Ensure you commit the transaction after associating with the VLAN.
Step 5	UCS-A/eth-storage/fabric# **exit**	Exits from the fabric.
Step 6	UCS-A/eth-storage# **scope vlan** *vlan-name*	Enters the specified VLAN. **Note** Ensure community VLAN is created in the appliance cloud.
Step 7	UCS-A/eth-storage/vlan# **create member-port** *fabric slot-num port-num*	Creates the member port for the specified fabric, assigns the slot number, and port number and enters member port configuration.
Step 8	UCS-A/eth-storage/vlan/member-port# **commit**	Commits the transaction to the system configuration.

Example

The following example maps an appliance port to an community VLAN:

```
UCS-A# scope eth-storage
UCS-A/eth-storage# scope fabric a
UCS-A/eth-storage/fabric# create interface 1 22
UCS-A/eth-storage/fabric/interface*# exit
UCS-A/eth-storage/fabric*# exit
UCS-A/eth-storage*# scope vlan COM602
UCS-A/eth-storage/vlan*# create member-port a 1 22
UCS-A/eth-storage/vlan/member-port* commit
```

Unconfiguring an Appliance Port

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope eth-storage**	Enters Ethernet storage mode.
Step 2	UCS-A /eth-storage # **scope fabric** {**a** | **b**}	Enters Ethernet storage mode for the specified fabric.

	Command or Action	Purpose
Step 3	UCS-A /eth-storage/fabric # **delete eth-interface** *slot-num port-num*	Deletes the interface for the specified appliance port.
Step 4	UCS-A /eth-storage/fabric # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example unconfigures appliance port 3 on slot 2 of fabric B and commits the transaction:

```
UCS-A# scope eth-storage
UCS-A /eth-storage # scope fabric b
UCS-A /eth-storage/fabric # delete eth-interface 2 3
UCS-A /eth-storage/fabric* # commit-buffer
UCS-A /eth-storage/fabric #
```

Configuring an Appliance Port for Forward Error Correction

You can configure forward error correction (FEC) for appliance ports that operate at 25 Gbps and 100 Gpbs speeds that support this feature.

Table 6: Supported Port Speed and FEC Matrix

Port Speed	FEC CL-74	FEC CL-91	RS Cons 16	RS 1eee
1 Gbps	Not supported	Not supported	-	-
10 Gbps	Not supported	Not supported	Not supported	Not supported
25 Gbps	Supported	Supported	Supported	Supported
40 Gbps	Not supported	Not supported	Not supported	Not supported
100 Gbps	Not supported	Supported	Not supported	Not supported
Auto	Automatically selects the optimal FEC mode based on the transceiver's maximum supported speed.	Automatically selects the optimal FEC mode based on the transceiver's maximum supported speed.	Automatically selects the optimal FEC mode based on the transceiver's maximum supported speed.	Automatically selects the optimal FEC mode based on the transceiver's maximum supported speed.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-storage**	Enters Appliance port mode.
Step 2	UCS-A /eth-storage # **scope fabric a** | **b**}	Enters Appliance port fabric mode for the specified fabric.

	Command or Action	Purpose
Step 3	UCS-A /eth-storage/fabric # **delete eth-interface** *slot-id port-id*	Enters Appliance interface mode for the specified interface.
Step 4	Required: UCS-A /eth-storage/fabric # **set fec** {**auto** |**cl74** | **cl91rs-cons16** | **rs-1eee**}	Sets the forward error correction setting. For the Cisco UCS 6400 and 6500 series Fabric Interconnects, forward error correction is only configurable for 25 Gbps or 100 Gbps port speed.
Step 5	UCS-A /eth-storage/fabric # **commit-buffer**	Commits the transaction to the system configuration.

FCoE Uplink Ports

FCoE uplink ports are physical Ethernet interfaces between the fabric interconnects and the upstream Ethernet switch, used for carrying FCoE traffic. With this support the same physical Ethernet port can carry both Ethernet traffic and Fibre Channel traffic.

FCoE uplink ports connect to upstream Ethernet switches using the FCoE protocol for Fibre Channel traffic. This allows both the Fibre Channel traffic and Ethernet traffic to flow on the same physical Ethernet link.

Note FCoE uplinks and unified uplinks enable the multi-hop FCoE feature, by extending the unified fabric up to the distribution layer switch.

You can configure the same Ethernet port as any of the following:

- **FCoE uplink port**—As an FCoE uplink port for only Fibre Channel traffic.
- **Uplink port**—As an Ethernet port for only Ethernet traffic.
- **Unified uplink port**—As a unified uplink port to carry both Ethernet and Fibre Channel traffic.

Configuring a FCoE Uplink Port

All of the port types listed are configurable on both the fixed and expansion module including server ports.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters FC Uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric**{**a** | **b**}	Enters FC - Uplink mode for the specific fabric.
Step 3	UCS-A /fc-uplink/fabric # **create fcoeinterface** *slot-numberport-number*	Creates interface for the specified FCoE uplink port.

	Command or Action	Purpose
Step 4	UCS-A /fc-uplink/fabric/fabricinterface # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example creates an interface for FCoE uplink port 8 on slot 1 of fabric A and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # create fcoeinterface 1 8
UCS-A /fc-uplink/fabric/fcoeinterface* # commit-buffer
UCS-A /fc-uplink/fabric/fcoeinterface #
```

Unconfiguring a FCoE Uplink Port

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters FC Uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric**{**a** | **b**}	Enters FC - Uplink mode for the specific fabric.
Step 3	UCS-A /fc-uplink/fabric # **delete fcoeinterface** *slot-numberport-number*	Deletes the specified interface.
Step 4	UCS-A /fc-uplink/fabric/fabricinterface # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example deletes the FCoE uplink interface on port 8 on slot 1 of fabric A and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # delete fcoeinterface 1 8
UCS-A /fc-uplink/fabric/fcoeinterface* # commit-buffer
UCS-A /fc-uplink/fabric/fcoeinterface #
```

Viewing FCoE Uplink Ports

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters FC Uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric** {**a** | **b**}	Enters FC - Uplink mode for the specific fabric.
Step 3	UCS-A /fc-uplink/fabric # **show fcoeinterface**	Lists the available interfaces.

Example

The following example displays the available FCoE uplink interfaces on fabric A:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # show fcoeinterface
FCoE Interface:

Slot Id    Port Id    Admin State Operational State Operational State Reason  Li
c State            Grace Prd
---------- ---------- ----------- ----------------- ------------------------- --
------------------ ---------
         1         26 Enabled     Indeterminate                               Li
cense Ok                    0

Fcoe Member Port:

Port-channel Slot  Port  Oper State      State Reason
------------ ----- ----- --------------- ------------
1                1    10 Sfp Not Present Unknown
1                1     3 Sfp Not Present Unknown
1                1     4 Sfp Not Present Unknown
1                1     6 Sfp Not Present Unknown
1                1     8 Sfp Not Present Unknown
2                1     7 Sfp Not Present Unknown
UCS-A /fc-uplink/fabric #
```

Configuring FCoE Uplink for Forward Error Correction

Cisco UCS Manager Release 4.3(4b) introduces support for FCoE uplink ports in Fibre Channel switch mode on the Cisco UCS Fabric Interconnects 9108 100G.

Cisco UCS Manager Release 4.2(3b) introduces support for FCoE uplink ports in Fibre Channel switch mode on the Cisco UCS 6536 Fabric Interconnect.

You can configure forward error correction (FEC) for FCoE uplinks that operate at 25 Gbps and 100 Gbps speeds that support this feature.

Table 7: FEC CL-74 and FEC CL-91 Support Matrix

Port Speed	FEC CL-74	FEC CL-91
1 Gbps	Not supported	Not supported
10 Gbps	Not supported	Not supported
25 Gbps	Supported	Supported
40 Gbps	Not supported	Not supported
100 Gbps	Not supported	Supported
Auto	Based on inserted tranceiver's maximum supported speed	Based on inserted tranceiver's maximum supported speed

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters FCoE uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric a** | **b**}	Enters fabric mode for the specified fabric.
Step 3	UCS-A /fc-uplink/fabric # **scope fcoeinterface** *slot-id port-id*	Enters FCoE interface mode for the specified interface.
Step 4	Required: UCS-A /fc-uplink/fabric/fcoeinterface # **set fec** {**auto** |**cl74** | **cl91**}	Sets the forward error correction setting as auto, cl74, or cl91 for the FCoE uplink. For the UCS 6400 Series Fabric Interconnect, Cisco UCS 6536 Fabric Interconnect, and Cisco UCS Fabric Interconnects 9108 100G fabric interconnects, the forward error correction is only configurable for 25 Gbps or 100 Gbps port speeds.
Step 5	UCS-A /fc-uplink/fabric/fcoeinterface # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to enable forward error correction cl74 on an interface for FCoE uplink 35 on slot 1 of fabric A, and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # scope fcoeinterface 1 35
UCS-A /fc-uplink/fabric/fcoeinterface # set fec cl74
UCS-A /fc-uplink/fabric/fcoeinterface # commit-buffer
```

Unified Storage Ports

Unified storage involves configuring the same physical port as both an Ethernet storage interface and an FCoE storage interface. You can configure any appliance port or FCoE storage port as a unified storage port. To configure a unified storage port, you must have the fabric interconnect in Fibre Channel switching mode.

In a unified storage port, you can enable or disable individual FCoE storage or appliance interfaces.

- In an unified storage port, if you do not specify a non-default VLAN for the appliance port, the FCoE-storage-native-vlan will be assigned as the native VLAN on the unified storage port. If the appliance port has a non-default native VLAN specified as native VLAN, this will be assigned as the native VLAN for the unified storage port.
- When you enable or disable the appliance interface, the corresponding physical port is enabled or disabled. So when you disable the appliance interface in unified storage, even if the FCoE storage is enabled, it goes down with the physical port.
- When you enable or disable the FCoE storage interface, the corresponding VFC is enabled or disabled. So when the FCoE storage interface is disabled in a unified storage port, the appliance interface will continue to function normally.

Configuring a Unified Storage Port

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-storage**	Enters Ethernet storage mode.
Step 2	UCS-A /eth-storage # **scope fabric**{**a** | **b**}	Enters Ethernet storage mode for the specified fabric.
Step 3	UCS-A /eth-storage/fabric # **create interface** *slot-num port-num*	Creates an interface for the specified appliance port.
Step 4	UCS-A /eth-storage/fabric/interface* # **commit buffer**	Commits the transaction to the system configuration.
Step 5	UCS-A /eth-storage/fabric/interface* # **scope fc-storage**	Enters FC storage mode.
Step 6	UCS-A /fc-storage* # **scope fabric**{**a** | **b**}	Enters Ethernet storage mode for the specific appliance port.
Step 7	UCS-A /fc-storage/fabric # **create interface fcoe** *slot-num port-num*	Adds FCoE storage port mode on the appliance port mode and creates a unified storage port.

Example

The following example creates an interface for an appliance port 2 on slot 3 of fabric A, adds fc storage to the same port to convert it as an unified port , and commits the transaction:

```
UCS-A# scope eth-storage
UCS-A /eth-storage # scope fabric a
UCS-A /eth-storage/fabric # create interface 3 2
UCS-A /eth-storage/fabric* # commit-buffer
UCS-A /eth-storage/fabric* # scope fc-storage
UCS-A /fc-storage*# scope fabric a
UCS-A /fc-storage/fabric* # create interface fcoe 3 2
UCS-A /fc-storage/fabric* # commit-buffer
UCS-A /fc-storage/fabric*
```

Unified Uplink Ports

When you configure an Ethernet uplink and an FCoE uplink on the same physical Ethernet port, it is called a unified uplink port. You can individually enable or disable either the FCoE or Ethernet interfaces independently.

- Enabling or disabling the FCoE uplink results in the corresponding VFC being enabled or disabled.
- Enabling or disabling an Ethernet uplink results in the corresponding physical port being enabled or disabled.

If you disable an Ethernet uplink, it disables the underlying physical port in a unified uplink. Therefore, even when the FCoE uplink is enabled, the FCoE uplink also goes down. But if you disable an FCoE uplink, only the VFC goes down. If the Ethernet uplink is enabled, it can still function properly in the unified uplink port.

Configuring a Unified Uplink Port

To configure a unified uplink port, you will convert an existing FCoE uplink port as a unified port.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric** {**a** | **b**}	Enters Ethernet uplink fabric mode for the specified fabric.
Step 3	UCS-A /eth-uplink/fabric # **create interface** *15*	Converts the FCoE uplink port as a unified port.
Step 4	UCS-A /eth-uplink/fabric/port-channel # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example creates a unified uplink port on an existing FCoE port:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric b
UCS-A /eth-uplink/fabric # create interface 1 5
UCS-A /eth-uplink/fabric/interface* # commit-buffer
UCS-A /eth-uplink/interface #
```

FCoE and Fibre Channel Storage Ports

Configuring a Fibre Channel Storage or FCoE Port

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-storage**	Enters Fibre Channel storage mode.
Step 2	UCS-A /fc-storage # **scope fabric** {**a** | **b**}	Enters Fibre Channel storage mode for the specified fabric.
Step 3	UCS-A /fc-storage/fabric # **create interface** {**fc** | **fcoe**} *slot-num port-num*	Creates an interface for the specified Fibre Channel storage port.
Step 4	UCS-A /fc-storage/fabric # **commit-buffer**	Commits the transaction.

Example

The following example creates an interface for Fibre Channel storage port 10 on slot 2 of fabric A and commits the transaction:

```
UCS-A# scope fc-storage
UCS-A /fc-storage # scope fabric a
UCS-A /fc-storage/fabric* # create interface fc 2 10
UCS-A /fc-storage/fabric # commit-buffer
```

What to do next

Assign a VSAN.

Unconfiguring a Fibre Channel Storage or FCoE Port

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-storage**	Enters Fibre Channel storage mode.
Step 2	UCS-A /fc-storage # **scope fabric** {**a** | **b**}	Enters Fibre Channel storage mode for the specified fabric.
Step 3	UCS-A /fc-storage/fabric # **delete interface** {**fc** | **fcoe**} *slot-num port-num*	Deletes the interface for the specified Fibre Channel or FCoE storage port.
Step 4	UCS-A /fc-storage/fabric # **commit-buffer**	Commits the transaction.

Example

The following example unconfigures Fibre Channel storage port 10 on slot 2 of fabric A and commits the transaction:

```
UCS-A# scope fc-storage
UCS-A /fc-storage # scope fabric a
UCS-A /fc-storage/fabric* # delete interface fc 2 10
UCS-A /fc-storage/fabric # commit-buffer
```

Restoring a Fibre Channel Storage Port Back to an Uplink Fibre Channel Port

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters Fibre Channel uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric** {**a** | **b**}	Enters Fibre Channel uplink mode for the specified fabric.
Step 3	UCS-A /fc-uplink/fabric # **create interface** *slot-num port-num*	Creates an interface for the specified Fibre Channel uplink port.
Step 4	UCS-A /fc-uplink/fabric # **commit-buffer**	Commits the transaction.

Example

The following example creates an interface for Fibre Channel uplink port 10 on slot 2 of fabric A and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
```

```
UCS-A /fc-uplink/fabric* # create interface 2 10
UCS-A /fc-uplink/fabric # commit-buffer
```

Uplink Ethernet Port Channels

An uplink Ethernet port channel allows you to group several physical uplink Ethernet ports (link aggregation) to create one logical Ethernet link to provide fault-tolerance and high-speed connectivity. In Cisco UCS Manager, you create a port channel first and then add uplink Ethernet ports to the port channel. You can add up to 16 uplink Ethernet ports to a port channel.

Important The state of a configured port changes to unconfigured in the following scenarios:

- The port is deleted or removed from a port channel. The port channel can be of any type, such as, uplink or storage.
- A port channel is deleted.

Note Cisco UCS uses Link Aggregation Control Protocol (LACP), not Port Aggregation Protocol (PAgP), to group the uplink Ethernet ports into a port channel. If the ports on the upstream switch are not configured for LACP, the fabric interconnects treat all ports in an uplink Ethernet port channel as individual ports, and therefore forward packets.

Configuring an Uplink Ethernet Port Channel

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric** {**a** | **b** }	Enters Ethernet uplink fabric mode for the specified fabric.
Step 3	UCS-A /eth-uplink/fabric # **create port-channel** *port-num*	Creates a port channel on the specified Ethernet uplink port, and enters Ethernet uplink fabric port channel mode.
Step 4	(Optional) UCS-A /eth-uplink/fabric/port-channel # {**enable** | **disable**}	Enables or disables the administrative state of the port channel. The port channel is disabled by default.
Step 5	(Optional) UCS-A /eth-uplink/fabric/port-channel # **set name** *port-chan-name*	Specifies the name for the port channel.

	Command or Action	Purpose
Step 6	(Optional) UCS-A /eth-uplink/fabric/port-channel # **set flow-control-policy** *policy-name*	Assigns the specified flow control policy to the port channel.
Step 7	UCS-A /eth-uplink/fabric/port-channel # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example creates a port channel on port 13 of fabric A, sets the name to portchan13a, enables the administrative state, assigns the flow control policy named flow-con-pol432 to the port channel, and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric a
UCS-A /eth-uplink/fabric # create port-channel 13
UCS-A /eth-uplink/fabric/port-channel* # enable
UCS-A /eth-uplink/fabric/port-channel* # set name portchan13a
UCS-A /eth-uplink/fabric/port-channel* # set flow-control-policy flow-con-pol432
UCS-A /eth-uplink/fabric/port-channel* # commit-buffer
UCS-A /eth-uplink/fabric/port-channel #
```

Unconfiguring an Uplink Ethernet Port Channel

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric** {**a** | **b** }	Enters Ethernet uplink fabric mode for the specified fabric.
Step 3	UCS-A /eth-uplink/fabric # **delete port-channel** *port-num*	Deletes the port channel on the specified Ethernet uplink port.
Step 4	UCS-A /eth-uplink/fabric # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example unconfigures the port channel on port 13 of fabric A and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric a
UCS-A /eth-uplink/fabric # delete port-channel 13
UCS-A /eth-uplink/fabric* # commit-buffer
UCS-A /eth-uplink/fabric #
```

Adding a Member Port to an Uplink Ethernet Port Channel

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric** {**a** | **b** }	Enters Ethernet uplink fabric mode for the specified fabric.
Step 3	UCS-A /eth-uplink/fabric # **scope port-channel** *port-num*	Enters Ethernet uplink fabric port channel mode for the specified port channel.
Step 4	UCS-A /eth-uplink/fabric/port-channel # **create member-port** *slot-num port-num*	Creates the specified member port from the port channel and enters Ethernet uplink fabric port channel member port mode.
Step 5	UCS-A /eth-uplink/fabric/port-channel # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example adds the member port on slot 1, port 7 to the port channel on port 13 of fabric A and commits the transaction.

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric a
UCS-A /eth-uplink/fabric # scope port-channel 13
UCS-A /eth-uplink/fabric/port-channel # create member-port 1 7
UCS-A /eth-uplink/fabric/port-channel* # commit-buffer
UCS-A /eth-uplink/fabric/port-channel #
```

Deleting a Member Port from an Uplink Ethernet Port Channel

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric** {**a** | **b** }	Enters Ethernet uplink fabric mode for the specified fabric.
Step 3	UCS-A /eth-uplink/fabric # **scope port-channel** *port-num*	Enters Ethernet uplink fabric port channel mode for the specified port channel.
Step 4	UCS-A /eth-uplink/fabric/port-channel # **delete member-port** *slot-num port-num*	Deletes the specified member port from the port channel.

	Command or Action	Purpose
Step 5	UCS-A /eth-uplink/fabric/port-channel # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example deletes a member port from the port channel on port 13 of fabric A and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric a
UCS-A /eth-uplink/fabric # scope port-channel 13
UCS-A /eth-uplink/fabric/port-channel # delete member-port 1 7
UCS-A /eth-uplink/fabric/port-channel* # commit-buffer
UCS-A /eth-uplink/fabric/port-channel #
```

Appliance Port Channels

An appliance port channel allows you to group several physical appliance ports to create one logical Ethernet storage link for the purpose of providing fault-tolerance and high-speed connectivity. In Cisco UCS Manager, you create a port channel first and then add appliance ports to the port channel. You can add up to eight appliance ports to a port channel.

Configuring an Appliance Port Channel

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-storage**	Enters Ethernet storage mode.
Step 2	UCS-A /eth-storage # **scope fabric** {**a** | **b** }	Enters Ethernet storage fabric mode for the specified fabric.
Step 3	UCS-A /eth-storage/fabric # **create port-channel** *port-num*	Creates a port channel on the specified Ethernet storage port, and enters Ethernet storage fabric port channel mode.
Step 4	(Optional) UCS-A /eth-storage/fabric/port-channel # {**enable** | **disable**}	Enables or disables the administrative state of the port channel. The port channel is disabled by default.
Step 5	(Optional) UCS-A /eth-storage/fabric/port-channel # **set name** *port-chan-name*	Specifies the name for the port channel.

	Command or Action	Purpose
Step 6	(Optional) UCS-A /eth-storage/fabric/port-channel # **set pingroupname** *pin-group name*	Specifies the appliance pin target to the specified fabric and port, or fabric and port channel.
Step 7	(Optional) UCS-A /eth-storage/fabric/port-channel # **set portmode** {**access** | **trunk**}	Specifies whether the port mode is access or trunk. By default, the mode is set to trunk.
Step 8	(Optional) UCS-A /eth-storage/fabric/port-channel # **set prio** *sys-class-name*	Specifies the QoS class for the appliance port. By default, the priority is set to best-effort. The sys-class-name argument can be one of the following class keywords: • **FC**—Use this priority for QoS policies that control vHBA traffic only. • **Platinum**—Use this priority for QoS policies that control vNIC traffic only. • **Gold**—Use this priority for QoS policies that control vNIC traffic only. • **Silver**—Use this priority for QoS policies that control vNIC traffic only. • **Bronze**—Use this priority for QoS policies that control vNIC traffic only. • **Best Effort**—Do not use this priority. It is reserved for the Basic Ethernet traffic lane. If you assign this priority to a QoS policy and configure another system class as CoS 0, Cisco UCS Manager does not default to this system class. It defaults to the priority with CoS 0 for that traffic.
Step 9	(Optional) UCS-A /eth-storage/fabric/port-channel # **set speed** {**1gbps** | **2gbps** | **4gbps** | **8gbps** | **auto**}	Specifies the speed for the port channel.
Step 10	UCS-A /eth-storage/fabric/port-channel # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example creates a port channel on port 13 of fabric A and commits the transaction:

```
UCS-A# scope eth-storage
UCS-A /eth-storage # scope fabric a
UCS-A /eth-storage/fabric # create port-channel 13
UCS-A /eth-storage/fabric/port-channel* # enable
UCS-A /eth-storage/fabric/port-channel* # set name portchan13a
```

```
UCS-A /eth-storage/fabric/port-channel* # set pingroupname pingroup1
UCS-A /eth-storage/fabric/port-channel* # set portmode access
UCS-A /eth-storage/fabric/port-channel* # set prio fc
UCS-A /eth-storage/fabric/port-channel* # set speed 2gbps
UCS-A /eth-storage/fabric/port-channel* # commit-buffer
UCS-A /eth-storage/fabric/port-channel #
```

Unconfiguring an Appliance Port Channel

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-storage**	Enters Ethernet storage mode.
Step 2	UCS-A /eth-storage # **scope fabric** {**a** | **b** }	Enters Ethernet storage fabric mode for the specified fabric.
Step 3	UCS-A /eth-storage/fabric # **delete port-channel** *port-num*	Deletes the port channel from the specified Ethernet storage port.
Step 4	UCS-A /eth-storage/fabric # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example unconfigures the port channel on port 13 of fabric A and commits the transaction:

```
UCS-A# scope eth-storage
UCS-A /eth-storage # scope fabric a
UCS-A /eth-storage/fabric # delete port-channel 13
UCS-A /eth-storage/fabric* # commit-buffer
UCS-A /eth-storage/fabric #
```

Enabling or Disabling an Appliance Port Channel

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-storage**	Enters Ethernet storage mode.
Step 2	UCS-A /eth-storage # **scope fabric** {**a** | **b** }	Enters Ethernet storage mode for the specified fabric.
Step 3	UCS-A /eth-storage/fabric # **scope port-channel** *port-chan-name*	Enters Ethernet storage port channel mode.

	Command or Action	Purpose
Step 4	UCS-A /eth-storage/fabric/port-channel # {**enable** | **disable** }	Enables or disables the administrative state of the port channel. The port channel is disabled by default.
Step 5	UCS-A /eth-storage/fabric/port-channel # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example enables port channel 13 on fabric A and commits the transaction:

```
UCS-A# scope eth-storage
UCS-A /eth-storage # scope fabric a
UCS-A /eth-storage/fabric # scope port-channel 13
UCS-A /eth-storage/fabric/port-channel* # enable
UCS-A /eth-storage/fabric/port-channel* # commit-buffer
UCS-A /eth-storage/fabric/port-channel #
```

Adding a Member Port to an Appliance Port Channel

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-storage**	Enters Ethernet storage mode.
Step 2	UCS-A /eth-storage # **scope fabric** {**a** | **b** }	Enters Ethernet storage fabric mode for the specified fabric.
Step 3	UCS-A /eth-storage/fabric # **scope port-channel** *port-num*	Enters Ethernet storage fabric port channel mode for the specified port channel.
Step 4	UCS-A /eth-storage/fabric/port-channel # **create member-port** *slot-num port-num*	Creates the specified member port from the port channel and enters Ethernet storage fabric port channel member port mode.
Step 5	UCS-A /eth-storage/fabric/port-channel # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example adds the member port on slot 1, port 7 to the port channel on port 13 of fabric A and commits the transaction.

```
UCS-A# scope eth-storage
UCS-A /eth-storage # scope fabric a
UCS-A /eth-storage/fabric # scope port-channel 13
UCS-A /eth-storage/fabric/port-channel # create member-port 1 7
UCS-A /eth-storage/fabric/port-channel* # commit-buffer
```

```
UCS-A /eth-storage/fabric/port-channel #
```

Deleting a Member Port from an Appliance Port Channel

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-storage**	Enters Ethernet storage mode.
Step 2	UCS-A /eth-storage # **scope fabric** {**a** | **b** }	Enters Ethernet storage fabric mode for the specified fabric.
Step 3	UCS-A /eth-storage/fabric # **scope port-channel** *port-num*	Enters Ethernet storage fabric port channel mode for the specified port channel.
Step 4	UCS-A /eth-storage/fabric/port-channel # **delete member-port** *slot-num port-num*	Deletes the specified member port from the port channel.
Step 5	UCS-A /eth-storage/fabric/port-channel # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example deletes a member port from the port channel on port 13 of fabric A and commits the transaction:

```
UCS-A# scope eth-storage
UCS-A /eth-storage # scope fabric a
UCS-A /eth-storage/fabric # scope port-channel 13
UCS-A /eth-storage/fabric/port-channel # delete member-port 1 7
UCS-A /eth-storage/fabric/port-channel* # commit-buffer
UCS-A /eth-storage/fabric/port-channel #
```

Fibre Channel Port Channels

A Fibre Channel port channel allows you to group several physical Fibre Channel ports (link aggregation) to create one logical Fibre Channel link to provide fault-tolerance and high-speed connectivity. In Cisco UCS Manager, you create a port channel first and then add Fibre Channel ports to the port channel.

Note Fibre Channel port channels are not compatible with non-Cisco technology.

You can create up to two Fibre Channel port channels in each Cisco UCS domain with Cisco UCS Fabric Interconnects 9108 100G (Cisco UCS X-Series Direct). Each Fibre Channel port channel can include a maximum of four uplink Fibre Channel ports.

For more information, see Port Breakout Functionality on respective fabric interconnects in Network Management Guide.

Ensure that the Fibre Channel port channel on the upstream NPIV switch is configured with its channel mode as **active**. If both the member port(s) and peer port(s) do not have the same channel mode configured, the port channel will not come up. When the channel mode is configured as **active**, the member ports initiate port channel protocol negotiation with the peer port(s) regardless of the channel group mode of the peer port. If the peer port, while configured in a channel group, does not support the port channel protocol, or responds with a nonnegotiable status, it defaults to the On mode behavior. The **active** port channel mode allows automatic recovery without explicitly enabling and disabling the port channel member ports at either end.

This example shows how to configure channel mode as active:

```
switch(config)# int po114
switch(config-if)# channel mode active
```

Configuring a Fibre Channel Port Channel

Note If you are connecting two Fibre Channel port channels, the admin speed for both port channels must match for the link to operate. If the admin speed for one or both of the Fibre Channel port channels is set to auto, Cisco UCS adjusts the admin speed automatically.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters Fibre Channel uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric** {**a** | **b** }	Enters Fibre Channel uplink fabric mode for the specified fabric.
Step 3	UCS-A /fc-uplink/fabric # **create port-channel** *port-num*	Creates a port channel on the specified Fibre Channel uplink port, and enters Fibre Channel uplink fabric port channel mode.
Step 4	(Optional) UCS-A /fc-uplink/fabric/port-channel # {**enable** | **disable**}	Enables or disables the administrative state of the port channel. The port channel is disabled by default.
Step 5	(Optional) UCS-A /fc-uplink/fabric/port-channel # **set name** *port-chan-name*	Specifies the name for the port channel.
Step 6	(Optional) UCS-A /fc-uplink/fabric/port-channel # **set speed** {**1gbps** | **2gbps** | **4gbps** | **8gbps** | **auto**}	Specifies the speed for the port channel.
Step 7	UCS-A /fc-uplink/fabric/port-channel # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example creates port channel 13 on fabric A, sets the name to portchan13a, enables the administrative state, sets the speed to 2 Gbps, and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # create port-channel 13
UCS-A /fc-uplink/fabric/port-channel* # enable
UCS-A /fc-uplink/fabric/port-channel* # set name portchan13a
UCS-A /fc-uplink/fabric/port-channel* # set speed 2gbps
UCS-A /fc-uplink/fabric/port-channel* # commit-buffer
UCS-A /fc-uplink/fabric/port-channel #
```

Configuring a FCoE Port Channel

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters FC Uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric**{**a** | **b**}	Enters FC - Uplink mode for the specific fabric.
Step 3	UCS-A /fc-uplink/fabric # **create fcoe-port-channel** *number*	Creates port channel for the specified FCoE uplink port.
Step 4	UCS-A /fc-uplink/fabric/fabricinterface # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example creates an interface for FCoE uplink port 1 on slot 4 of fabric A and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # create fcoe-port-channel 4
UCS-A /fc-uplink/fabric/fcoe-port-channel* # commit-buffer
UCS-A /fc-uplink/fabric/fcoe-port-channel #
```

Adding Channel Mode Active To The Upstream NPIV Fibre Channel Port Channel

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters Fibre Channel uplink mode.

	Command or Action	Purpose
Step 2	UCS-A /fc-uplink # **scope fabric** {**a** | **b** }	Enters Fibre Channel uplink fabric mode for the specified fabric.
Step 3	UCS-A /fc-uplink/fabric # **create port-channel** *port-num*	Creates a port channel on the specified Fibre Channel uplink port, and enters Fibre Channel uplink fabric port channel mode.
Step 4	(Optional) UCS-A /fc-uplink/fabric/port-channel # {**enable** | **disable**}	Enables or disables the administrative state of the port channel. The port channel is disabled by default.
Step 5	(Optional) UCS-A /fc-uplink/fabric/port-channel # **set name** *port-chan-name*	Specifies the name for the port channel.
Step 6	(Optional) UCS-A /fc-uplink/fabric/port-channel # **scope** *port-chan-name*	Specifies the name for the port channel.
Step 7	(Optional) UCS-A /fc-uplink/fabric/port-channel # **channel mode** {**active**}	Configures the channel-mode active on the upstream NPIV switch.
Step 8	UCS-A /fc-uplink/fabric/port-channel # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example enables channel mode to active:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # create port-channel 13
UCS-A /fc-uplink/fabric/port-channel* # enable
UCS-A /fc-uplink/fabric/port-channel* # set name portchan13a
UCS-A /fc-uplink/fabric/port-channel* # channel mode active
UCS-A /fc-uplink/fabric/port-channel* # commit-buffer
UCS-A /fc-uplink/fabric/port-channel # exit
UCS-A /fc-uplink/fabric/ # show port-channel database

portchan13a
    Administrative channel mode is active
    Operational channel mode is active

UCS-A /fc-uplink/fabric/ #
```

Enabling or Disabling a Fibre Channel Port Channel

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters Fibre Channel uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric** {**a** | **b** }	Enters Fibre Channel uplink mode for the specified fabric.
Step 3	UCS-A /fc-uplink/fabric # **scope port-channel** *port-chan-name*	Enters Fibre Channel uplink port channel mode.
Step 4	UCS-A /fc-uplink/fabric/port-channel # {**enable** | **disable** }	Enables or disables the administrative state of the port channel. The port channel is disabled by default.

Example

The following example enables port channel 13 on fabric A and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # scope port-channel 13
UCS-A /fc-uplink/fabric/port-channel* # enable
UCS-A /fc-uplink/fabric/port-channel* # commit-buffer
UCS-A /fc-uplink/fabric/port-channel #
```

Adding a Member Port to a Fibre Channel Port Channel

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters Fibre Channel uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric** {**a** | **b** }	Enters Fibre Channel uplink fabric mode for the specified fabric.
Step 3	UCS-A /fc-uplink/fabric # **scope port-channel** *port-num*	Enters Fibre Channel uplink fabric port channel mode for the specified port channel.
Step 4	UCS-A /fc-uplink/fabric/port-channel # **create member-port** *slot-num port-num*	Creates the specified member port from the port channel and enters Fibre Channel uplink fabric port channel member port mode.
Step 5	UCS-A /fc-uplink/fabric/port-channel # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example adds the member port on slot 1, port 7 to port channel 13 on fabric A and commits the transaction.

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # scope port-channel 13
UCS-A /fc-uplink/fabric # create member-port 1 7
UCS-A /fc-uplink/fabric/port-channel* # commit-buffer
UCS-A /fc-uplink/fabric/port-channel #
```

Deleting a Member Port from a Fibre Channel Port Channel

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters Fibre Channel uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric** {**a** | **b**}	Enters Fibre Channel uplink fabric mode for the specified fabric.
Step 3	UCS-A /fc-uplink/fabric # **scope port-channel** *port-num*	Enters Fibre Channel uplink fabric port channel mode for the specified port channel.
Step 4	UCS-A /fc-uplink/fabric/port-channel # **delete member-port** *slot-num port-num*	Deletes the specified member port from the port channel.
Step 5	UCS-A /fc-uplink/fabric/port-channel # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example deletes a member port from port channel 13 on fabric A and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # scope port-channel 13
UCS-A /fc-uplink/fabric # delete member-port 1 7
UCS-A /fc-uplink/fabric/port-channel* # commit-buffer
UCS-A /fc-uplink/fabric/port-channel #
```

FCoE Port Channels

An FCoE port channel allows you to group several physical FCoE ports to create one logical FCoE port channel. At a physical level, the FCoE port channel carries FCoE traffic over an Ethernet port channel. So an

FCoE port channel with a set of members is essentially an Ethernet port channel with the same members. This Ethernet port channel is used as a physical transport for FCoE traffic.

For each FCoE port channel, Cisco UCS Manager creates a VFC internally and binds it to an Ethernet port channel. FCoE traffic received from the hosts is sent over the VFC the same way as the FCoE traffic is sent over Fibre Channel uplinks.

Configuring a FCoE Port Channel

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters FC Uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric**{**a** | **b**}	Enters FC - Uplink mode for the specific fabric.
Step 3	UCS-A /fc-uplink/fabric # **create fcoe-port-channel** *number*	Creates port channel for the specified FCoE uplink port.
Step 4	UCS-A /fc-uplink/fabric/fabricinterface # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example creates an interface for FCoE uplink port 1 on slot 4 of fabric A and commits the transaction:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # create fcoe-port-channel 4
UCS-A /fc-uplink/fabric/fcoe-port-channel* # commit-buffer
UCS-A /fc-uplink/fabric/fcoe-port-channel #
```

Adding a Member Port to a FCoE Uplink Port Channel

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters Fibre Channel uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric** {**a** | **b** }	Enters Fibre Channel uplink fabric mode for the specified fabric.
Step 3	UCS-A /fc-uplink/fabric # **scope fcoe-port-channel** *ID*	Enters FCoE uplink port channel mode for the specified port channel.

	Command or Action	Purpose
Step 4	UCS-A /fc-uplink/fabric/fcoe-port-channel # **create member-port** *slot-num port-num*	Creates the specified member port from the port channel and enters FCoE uplink fabric port channel member port mode. **Note** If the FCoE uplink port channel is a unified uplink port channel, you will get the following message: Warning: if this is a unified port channel then member will be added to the ethernet port channel of the same id as well.
Step 5	UCS-A /fc-uplink/fabric/fcoe-port-channel # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example adds the member port on slot 1, port 7 to FCoE port channel 13 on fabric A and commits the transaction.

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # scope fcoe-port-channel 13
UCS-A /fc-uplink/fabric # create member-port 1 7
UCS-A /fc-uplink/fabric/fcoe-port-channel* # commit-buffer
UCS-A /fc-uplink/fabric/fcoe-port-channel #
```

Unified Uplink Port Channel

When you create an Ethernet port channel and an FCoE port channel with the same ID, it is called a unified uplink port channel. When the unified port channel is created, a physical Ethernet port channel and a VFC are created on the fabric interconnect with the specified members. The physical Ethernet port channel is used to carry both Ethernet and FCoE traffic. The VFC binds FCoE traffic to the Ethernet port channel.

The following rules will apply to the member port sets of the unified uplink port channel:

- The Ethernet port channel and FCoE port channel on the same ID, must have the same set of member ports.
- When you add a member port channel to the Ethernet port channel, Cisco UCS Manager adds the same port channel to FCoE port channel as well. Similarly, adding a member to the FCoE port channel adds the member port to the Ethernet port channel.
- When you delete a member port from one of the port channels, Cisco UCS Manager automatically deletes the member port from the other port channel.

If you disable an Ethernet uplink port channel, it disables the underlying physical port channel in a unified uplink port channel. Therefore, even when the FCoE uplink is enabled, the FCoE uplink port channel also

goes down. If you disable an FCoE uplink port channel, only the VFC goes down. If the Ethernet uplink port channel is enabled, it can still function properly in the unified uplink port channel.

Configuring a Unified Uplink Port Channel

To configure a unified uplink port channel, you will convert an existing FCoE uplink port channel as a unified port channel.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric** {**a** | **b**}	Enters Ethernet uplink fabric mode for the specified fabric.
Step 3	UCS-A /eth-uplink/fabric # **create port-channel** *ID*	Creates a port channel for the specified Ethernet uplink port.
Step 4	UCS-A /eth-uplink/fabric/port-channel # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example creates a unified uplink port channel on an existing FCoE port channel:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric b
UCS-A /eth-uplink/fabric # create port-channel 2
UCS-A /eth-uplink/fabric/port-channel* # commit-buffer
UCS-A /eth-uplink/fabric #
```

Event Detection and Action

Cisco UCS Manager uses the statistics collection policy to monitor and trigger an alarm when there are faults in the network interface ports connected from the I/O Module (IOM) to the fabric interconnect.

The error statistics for the network interface ports is called NiErrStats and consists of the following errors:

NiErrStats	Description
frameTx	Collects the TX_FRM_ERROR counter values.
tooLong	Collects the RX_TOOLONG counter values.
tooShort	Collects the sum of RX_UNDERSIZE and RX_FRAGMENT counter values.
Crc	Collects the sum of RX_CRERR_NOT_STOMPED and RX_CRCERR_STOMPED counter values.

InRange	Collects the RX_INRANGEERR counter values.

Note Only active ports collect the network interface port statistics and send the information to Cisco UCS Manager.

Policy-Based Port Error Handling

If Cisco UCS Manager detects any errors on active NI ports, and if the error-disable feature is enabled, Cisco UCS Manager automatically disables the respective FI port that is connected to the NI port that had errors. When a FI port is error disabled, it is effectively shut down and no traffic is sent or received on that port.

The error-disable function serves two purposes:

- It lets you know which FI port is error-disabled and that the connected NI Port has errors.
- It eliminates the possibility that this port can cause other ports, which are connected to the same Chassis/FEX, to fail. Such a failure can occur when the NI port has errors, which can ultimately cause serious network issues. The error-disable function helps prevent these situations.

Creating Threshold Definition

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope eth-server**	Enters Ethernet storage mode.
Step 2	UCS-A/eth-server # **scope stats-threshold-policy default**	Enters statistics threshold policy mode.
Step 3	UCSA/eth-server/stats-threshold-policy # **create class** *class-name*	Creates the specified statistics threshold policy class and enters the organization statistics threshold policy class mode. To see a list of the available class name keywords, enter the **create class ?** command in organization threshold policy mode.
Step 4	UCS-A/eth-server/stats-threshold-policy/class # **create property** *property-name*	Creates the specified statistics threshold policy class property and enters the organization statistics threshold policy class property mode. To see a list of the available property name keywords, enter the **create property ?** command in organization threshold policy class mode.
Step 5	UCS-A/eth-server/stats-threshold-policy/class/property # **set normal-value** *value*	Specifies the normal value for the class property. The *value* format can vary depending on the class property being configured. To see the required format, enter the **set normal-value**

	Command or Action	Purpose
		? command in organization statistics threshold policy class property mode.
Step 6	UCS-A/eth-server/stats-threshold-policy/class/property # **create threshold-value** {*above-normal* | *below-normal*} {*cleared* | *condition* | *critical* | *info* | *major* | *minor* | *warning*}	Creates the specified threshold value for the class property and enters the organization statistics threshold policy class property threshold value mode.
Step 7	UCS-A/eth-server/stats-threshold-policy/class/property/threshold-value # **set** {**deescalating** | **escalating**} *value*	Specifies the deescalating and escalating class property threshold value. The *value* format can vary depending on the class property threshold value being configured. To see the required format, enter the **set deescalating ?** or **set escalating ?** command in the organization statistics threshold policy class property threshold value mode.
Step 8	UCS-A/eth-server/stats-threshold-policy/class/property/threshold-value # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to create a threshold definition:

```
UCS-A # scope eth-server
UCS-A /eth-server # scope stats-threshold-policy default
UCS-A /eth-server/stats-threshold-policy # create class ni-ether-error-stats
UCS-A /eth-server/stats-threshold-policy/class* # create property crc-delta
UCS-A /eth-server/stats-threshold-policy/class/property* # set normal-value 0
UCS-A /eth-server/stats-threshold-policy/class/property* # create threshold-value above-normal
 major
UCS-A /eth-server/stats-threshold-policy/class/property/threshold-value* # set escalating
5
UCS-A /eth-server/stats-threshold-policy/class/property/threshold-value* # set deescalating
 3
UCS-A /eth-server/stats-threshold-policy/class/property/threshold-value* # commit-buffer
```

Configuring Error Disable on a Fabric Interconnect Port

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope eth-server**	Enters Ethernet storage mode.
Step 2	UCS-A/eth-server # **scope stats-threshold-policy default**	Enters statistics threshold policy mode.

	Command or Action	Purpose
Step 3	UCSA/eth-server/stats-threshold-policy # **scope class** *class-name*	Enters the organization statistics threshold policy class mode for the specified statistics threshold policy class.
Step 4	UCS-A/eth-server/stats-threshold-policy/class # **scope property** *property-name*	Enters the organization statistics threshold policy class property mode for the specified statistics threshold policy class property.
Step 5	UCS-A/eth-server/stats-threshold-policy/class/property # **set error-disable-fi-port** {**yes** | **no**}	Specifies the error disable state for the class property. Use the **no** option to disable error disable for the class property.
Step 6	UCS-A/eth-server/stats-threshold-policy/class/property* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to enable error disable on an FI port:

```
UCS-A # scope eth-server
UCS-A /eth-server # scope stats-threshold-policy default
UCS-A /eth-server/stats-threshold-policy # scope class ni-ether-error-stats
UCS-A /eth-server/stats-threshold-policy/class # scope property crc-delta
UCS-A /eth-server/stats-threshold-policy/class/property # set error-disable-fi-port yes
UCS-A /eth-server/stats-threshold-policy/class/property* # commit-buffer
```

Configuring Auto Recovery on a Fabric Interconnect Port

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope eth-server**	Enters Ethernet storage mode.
Step 2	UCS-A/eth-server # **scope stats-threshold-policy default**	Enters statistics threshold policy mode.
Step 3	UCSA/eth-server/stats-threshold-policy # **scope class** *class-name*	Enters the organization statistics threshold policy class mode for the specified statistics threshold policy class.
Step 4	UCS-A/eth-server/stats-threshold-policy/class # **scope property** *property-name*	Enters the organization statistics threshold policy class property mode for the specified statistics threshold policy class property.
Step 5	UCS-A/eth-server/stats-threshold-policy/class/property # **set auto-recovery** {**enabled** | **disabled**}	Specifies the auto recovery state for the class property.

	Command or Action	Purpose
		Use the **disabled** option to disable auto recovery for the class property.
Step 6	UCS-A/eth-server/stats-threshold-policy/class/property* # **set auto-recovery-time** *time*	Specifies the time in minutes after which the port is automatically re-enabled. The auto recovery time can range from 0 minutes to 4294967295 minutes.
Step 7	UCS-A/eth-server/stats-threshold-policy/class/property* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to configure auto recovery on an FI port:

```
UCS-A # scope eth-server
UCS-A /eth-server # scope stats-threshold-policy default
UCS-A /eth-server/stats-threshold-policy # scope class ni-ether-error-stats
UCS-A /eth-server/stats-threshold-policy/class # scope property crc-delta
UCS-A /eth-server/stats-threshold-policy/class/property # set auto-recovery enabled
UCS-A /eth-server/stats-threshold-policy/class/property* # set auto-recovery-time 5
UCS-A /eth-server/stats-threshold-policy/class/property* # commit-buffer
```

Viewing the Network Interface Port Error Counters

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope chassis** *chassis-num*	Enters chassis mode for the specified chassis.
Step 2	UCS-A/chassis # **scope iom** {**a** | **b**}	Enters chassis IOM mode for the specified IOM.
Step 3	UCS-A/chassis/iom # **scope port-group fabric**	Enters the network interface port.
Step 4	UCS-A/chassis/iom/port-group # **scope fabric-if** *fabric-if number*	Enters the specified network interface port number.
Step 5	UCS-A/chassis/iom/port-group/fabric-if # **show stats**	Displays the error counters for the network interface port.

Example

The following example shows how to display the statistics for the network interface ports:

```
UCS-A # scope chassis 1
UCS-A/chassis # scope iom a
UCS-A/chassis/iom # scope port-group fabric
```

```
UCS-A/chassis/iom/port-group # scope faric-if 1
UCS-A/chassis/iom/port-group/fabric-if # show stats
NI Ether Error Stats:
Time Collected: 2014-08-20T15:37:24:688
Monitored Object: sys/chassis-1/slot-1/fabric/port-1/ni-err-stats
Suspect: Yes
Crc (errors): 5000
Frame Tx (errors): 0
Too Long (errors): 0
Too Short (errors): 0
In Range (errors): 0
Thresholded: 0
```

Adapter Port Channels

An adapter port channel groups into one logical link all the physical links going from a Cisco UCS Virtual Interface Card (VIC) into an I/O.

Adapter port channels are created and managed internally by Cisco UCS Manager when it detects that the correct hardware is present. Adapter port channels cannot be configured manually. Adapter port channels are viewable using the Cisco UCS Manager GUI or the Cisco UCS Manager CLI.

Viewing Adapter Port Channels

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope chassis** *chassis-num*	Enters chassis mode for the specified chassis.
Step 2	UCS-A /chassis # **scope iom** {**a** **b**}	Enters chassis IOM mode for the specified IOM.
Step 3	UCS-A /chassis/iom # **scope port group**	Enters port group mode for the specified port group.
Step 4	UCS-A /chassis/iom/port group # **show host-port-channel** [**detail** | **expand**]	Displays the adapter port channels on the specified chassis.

Example

This following example shows how to display information on host port channels within a port group mode:

```
UCS-A # scope chassis 1
UCS-A /chassis # scope iom a
UCS-A /chassis/iom # scope port group
UCS-A /chassis/iom/port group # show host-port-channel

Host Port channel:

  Port Channel Id Fabric ID Oper State       State Reason
    --------------- --------- ---------------- ------------
               1289 B         Up
```

```
                1290 B          Up
                1306 B          Up
                1307 B          Up
                1309 B          Up
                1315 B          Up

UCS-A /chassis/iom/port group #
```

Fabric Port Channels

Fabric port channels allow you to group several of the physical links from an IOM and IFM (IOM for Cisco UCS X-Series Servers) to a fabric interconnect into one logical link for redundancy and bandwidth sharing. As long as one link in the fabric port channel remains active, the fabric port channel continues to operate.

If the correct hardware is connected, fabric port channels are created by Cisco UCS Manager in the following ways:

- During chassis discovery according to the settings configured in the chassis discovery policy.
- After chassis discovery according to the settings configured in the chassis connectivity policy for a specific chassis.

For each IOM and IFM (IOM for Cisco UCS X-Series Servers) there is a single fabric port channel. Each uplink connecting an IOM and IFM (IOM for Cisco UCS X-Series Servers) to a fabric interconnect can be configured as a discrete link or included in the port channel, but an uplink cannot belong to more than one fabric port channel. For example, if a chassis with two IOMs is discovered and the chassis discovery policy is configured to create fabric port channels, Cisco UCS Manager creates two separate fabric port channels: one for the uplinks connecting IOM-1 and another for the uplinks connecting IOM-2. No other chassis can join these fabric port channels. Similarly, uplinks belonging to the fabric port channel for IOM-1 cannot join the fabric port channel for IOM-2.

Load Balancing Over Ports

Load balancing traffic among ports between IOMs and fabric interconnects uses the following criteria for hashing.

- For Ethernet traffic:

  Layer 2 source and destination address

  Layer 3 source and destination address

  Layer 4 source and destination ports

- For FCoE traffic:

  Layer 2 source and destination address

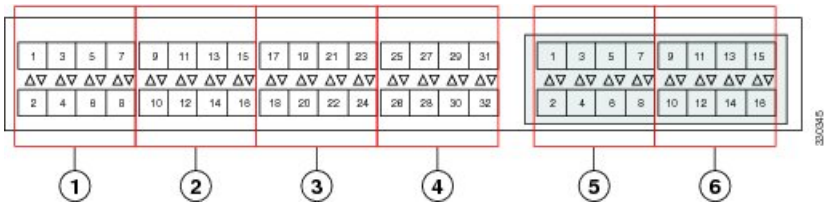
  Source and destination IDs (SID and DID) and Originator Exchange ID (OXID)

Cabling Considerations for Fabric Port Channels

When you configure the links between the Cisco UCS 2200 Series FEX and a Cisco UCS 6200 series fabric interconnect in fabric port channel mode, the available virtual interface namespace (VIF) on the adapter varies depending on where the FEX uplinks are connected to the fabric interconnect ports.

Inside the 6248 fabric interconnect there are six sets of eight contiguous ports, with each set of ports managed by a single chip. When all uplinks from an FEX are connected to a set of ports managed by a single chip, Cisco UCS Manager maximizes the number of VIFs used in service profiles deployed on the blades in the chassis. If uplink connections from an IOM are distributed across ports managed by separate chips, the VIF count is decreased.

Figure 1: Port Groups for Fabric Port Channels



Caution Adding a second link to a fabric-port-channel port group is disruptive and will automatically increase the available amount of VIF namespace from 63 to 118. Adding further links is not disruptive and the VIF namespace stays at 118.

Caution Linking a chassis to two fabric-port-channel port groups does not affect the VIF namespace unless it is manually acknowledged. The VIF namespace is then automatically set to the smaller size fabric port-channel port group usage (either 63 or 118 VIFs) of the two groups.

For high availability cluster-mode applications, we strongly recommend symmetric cabling configurations. If the cabling is asymmetric, the maximum number of VIFs available is the smaller of the two cabling configurations.

For more information on the maximum number of VIFs for your Cisco UCS environment, see the Configuration Limits document for your hardware and software configuration.

Configuring a Fabric Port Channel

Procedure

Step 1 To include all links from the IOM to the fabric interconnect in a fabric port channel during chassis discovery, set the link grouping preference in the chassis discovery policy to port channel.

Step 2 To include links from individual chassis in a fabric port channel during chassis discovery, set the link grouping preference in the chassis connectivity policy to port channel.

Step 3 After chassis discovery, enable or disable additional fabric port channel member ports.

What to do next

To add or remove chassis links from a fabric port channel after making a change to the chassis discovery policy or the chassis connectivity policy, reacknowledge the chassis. Chassis reacknowledgement is not required to enable or disable chassis member ports from a fabric port channel

Viewing Fabric Port Channels

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-server**	Enters Ethernet server mode.
Step 2	UCS-A /eth-server # **scope fabric** {**a** | **b**}	Enters Ethernet server fabric mode for the specified fabric.
Step 3	UCS-A /eth-server/fabric # **show fabric-port-channel** [**detail** | **expand**]	Displays fabric port channels on the specified fabric interconnect.

Example

The following example displays information about configured fabric port channels on fabric interconnect A:

```
UCS-A# scope eth-server
UCS-A /eth-server # scope fabric a
UCS-A /eth-server/fabric # show fabric-port-channel
Fabric Port Channel:
    Port Channel Id Chassis Id Admin State Oper State       State Reason
    --------------- ---------- ----------- ---------------- ------------
               1025 1          Enabled     Failed           No operational members
               1026 2          Enabled     Up

UCS-A /eth-server/fabric #
```

Enabling or Disabling a Fabric Port Channel Member Port

Procedure

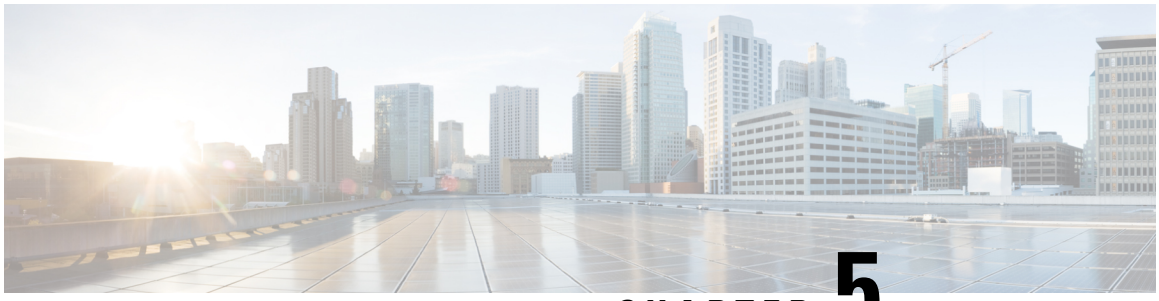
	Command or Action	Purpose
Step 1	UCS-A# **scope eth-server**	Enters Ethernet server mode.
Step 2	UCS-A /eth-server # **scope fabric** {**a** | **b**}	Enters Ethernet server fabric mode for the specified fabric.

	Command or Action	Purpose
Step 3	UCS-A /eth-server/fabric # **scope fabric-port-channel** *port-chan-id*	Enters Ethernet server fabric, fabric port channel mode for the specified fabric.
Step 4	UCS-A /eth-server/fabric/fabric-port-channel # **scope member-port** *slot-id port-id*	Enters Ethernet server fabric, fabric port channel mode for the specified member port.
Step 5	UCS-A /eth-server/fabric/fabric-port-channel # {**enable** | **disable**}	Enables or disables the specified member port.
Step 6	UCS-A /eth-server/fabric/fabric-port-channel # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example disables fabric channel member port 1 31 on fabric port channel 1025 and commits the transaction:

```
UCS-A# scope eth-server
UCS-A /eth-server # scope fabric a
UCS-A /eth-server/fabric # scope fabric-port-channel 1025
UCS-A /eth-server/fabric/fabric-port-channel # scope member-port 1 31
UCS-A /eth-server/fabric/fabric-port-channel/member-port # disable
UCS-A /eth-server/fabric/fabric-port-channel/member-port* # commit-buffer
UCS-A /eth-server/fabric/fabric-port-channel/member-port #
```



CHAPTER 5

VLANs

- VLANs, on page 99
- About the Native VLAN, on page 100
- Named VLANs, on page 100
- Private VLANs, on page 101
- VLAN Port Limitations, on page 102
- Configuring Named VLANs, on page 104
- Configuring Private VLANs, on page 110
- Community VLANs , on page 117
- Viewing the VLAN Port Count, on page 121
- VLAN Port Count Optimization, on page 121
- VLAN Groups, on page 124
- VLAN Permissions, on page 128
- Fabric Port-Channel vHBA, on page 130
- VIC QinQ Tunneling, on page 132

VLANs

A VLAN is a switched network that is logically segmented by function, project team, or application, without regard to the physical locations of the users. VLANs have the same attributes as physical LANs, but you can group end stations even if they are not physically located on the same LAN segment.

Any switch port can belong to a VLAN. Unicast, broadcast, and multicast packets are forwarded and flooded only to end stations in the VLAN. Each VLAN is considered a logical network, and packets destined for stations that do not belong to the VLAN must be forwarded through a router or bridge.

VLANs are typically associated with IP subnetworks. For example, all of the end stations in a particular IP subnet belong to the same VLAN. To communicate between VLANs, you must route the traffic. By default, a newly created VLAN is operational. Additionally, you can configure VLANs to be in the active state, which is passing traffic, or in the suspended state, in which the VLANs are not passing packets. By default, the VLANs are in the active state and pass traffic.

About the Native VLAN

The Native VLAN and the default VLAN serve different purposes within a network. Native VLAN refers to the VLAN that handles untagged traffic—Ethernet frames transmitted without an 802.1Q VLAN tag. Native VLAN traffic is untagged, and its frames are transmitted without modification. The Native VLAN can either be assigned to a specific VLAN or left unconfigured.

It is possible to tag all VLAN traffic and eliminate the use of a Native VLAN across your network. By default, VLAN 1 is assigned as the Native VLAN on switches, but this setting can be modified to meet specific network requirements.

The UCS Manager LAN Uplink Manager allows you to configure VLANs and change the Native VLAN setting.

Changing the Native VLAN triggers a single port flap, resulting in a temporary connectivity loss of approximately 20–40 seconds. This port flap is necessary for the change to take effect. However, continuous port flapping is not expected and may indicate underlying configuration issues that require troubleshooting.

Native VLAN Guidelines

- Native VLANs can only be configured on trunk ports.
- When changing the native VLAN on a UCS vNIC, a port flap will occur, leading to brief traffic interruptions.
- Cisco recommends using the Native VLAN 1 setting to minimize traffic interruptions, particularly when using the Cisco Nexus 1000v switches. Ensure the Native VLAN configuration is consistent between the Nexus 1000v port profiles and the UCS vNIC definition.
- If there is a continuous port flapping, incorrect traffic routing, or outages, verify the configuration of your disjoint Layer 2 network for potential issues.
- Using VLAN 1 for management access across all devices can lead to potential security risks if another switch is connected to the same VLAN as your management devices.

Named VLANs

A named VLAN creates a connection to a specific external LAN. The VLAN isolates traffic to that external LAN, including broadcast traffic.

The name that you assign to a VLAN ID adds a layer of abstraction that allows you to globally update all servers associated with service profiles that use the named VLAN. You do not need to reconfigure the servers individually to maintain communication with the external LAN.

You can create more than one named VLAN with the same VLAN ID. For example, if servers that host business services for HR and Finance need to access the same external LAN, you can create VLANs named HR and Finance with the same VLAN ID. Then, if the network is reconfigured and Finance is assigned to a different LAN, you only have to change the VLAN ID for the named VLAN for Finance.

In a cluster configuration, you can configure a named VLAN to be accessible only to one fabric interconnect or to both fabric interconnects.

Guidelines for VLAN IDs

Important VLAN IDs **3915 to 4042** and **4043 to 4047** are reserved. You cannot use these VLAN IDs to create new VLANs.

For Cisco UCS 6600 Series Fabric Interconnect, Cisco UCS Fabric Interconnects 9108 100G, Cisco UCS 6500 and 6400 Series Fabric Interconnects, VLAN IDs from 1002 to 1005 are reserved for VLAN Trunking Protocol (VTP).

The VLAN IDs you specify must also be supported on the switch that you are using. For example, on Cisco Nexus 5000 Series switches, the VLAN ID range from 3968 to 4029 is reserved. Before you specify the VLAN IDs in Cisco UCS Manager, make sure that the same VLAN IDs are available on your switch.

VLANs in the LAN cloud and FCoE VLANs in the SAN cloud must have different IDs. Using the same ID for a VLAN and an FCoE VLAN in a VSAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that VLAN. Ethernet traffic is dropped on any VLAN which has an ID that overlaps with an FCoE VLAN ID.

VLAN 4048 is user configurable. However, Cisco UCS Manager uses VLAN 4048 for the following default values. If you want to assign 4048 to a VLAN, you must reconfigure these values:

- After an upgrade to Cisco UCS, Release 2.0—The FCoE storage port native VLAN uses VLAN 4048 by default. If the default FCoE VSAN was set to use VLAN 1 before the upgrade, you must change it to a VLAN ID that is not used or reserved. For example, consider changing the default to 4049 if that VLAN ID is not in use.
- After a fresh install of Cisco UCS, Release 2.0—The FCoE VLAN for the default VSAN uses VLAN 4048 by default. The FCoE storage port native VLAN uses VLAN 4049.

The VLAN name is case sensitive.

Private VLANs

A private VLAN (PVLAN) partitions the Ethernet broadcast domain of a VLAN into subdomains, and allows you to isolate some ports. Each subdomain in a PVLAN includes a primary VLAN and one or more secondary VLANs. All secondary VLANs in a PVLAN must share the same primary VLAN. The secondary VLAN ID differentiates one subdomain from another.

Isolated and Community VLANs

All secondary VLANs in a Cisco UCS domain can be Isolated or Community VLANs.

Note You cannot configure an isolated VLAN to use with a regular VLAN.

Ports on Isolated VLANs

Communications on an isolated VLAN can only use the associated port in the primary VLAN. These ports are isolated ports and are not configurable in Cisco UCS Manager. A primary VLAN can have only one isolated VLAN, but multiple isolated ports on the same isolated VLAN are allowed. These isolated ports

cannot communicate with each other. The isolated ports can communicate only with a regular trunk port or promiscuous port that allows the isolated VLAN.

An isolated port is a host port that belongs to an isolated secondary VLAN. This port has complete isolation from other ports within the same private VLAN domain. PVLANs block all traffic to isolated ports except traffic from promiscuous ports. Traffic received from an isolated port is forwarded only to promiscuous ports. You can have more than one isolated port in a specified isolated VLAN. Each port is completely isolated from all other ports in the isolated VLAN.

Guidelines for Uplink Ports

When you create PVLANs, use the following guidelines:

- The uplink Ethernet port channel cannot be in promiscuous mode.
- Each primary VLAN can have only one isolated VLAN.
- VIFs on VNTAG adapters can have only one isolated VLAN.

Guidelines for VLAN IDs

Note You cannot create VLANs with IDs from 3915 to 4042. These ranges of VLAN IDs are reserved.

The VLAN IDs you specify must also be supported on the switch that you are using. For example, on Cisco Nexus 5000 Series switches, the VLAN ID range from 3968 to 4029 is reserved. Before you specify the VLAN IDs in Cisco UCS Manager, make sure that the same VLAN IDs are available on your switch.

VLANs in the LAN cloud and FCoE VLANs in the SAN cloud must have different IDs. Using the same ID for a VLAN and an FCoE VLAN in a VSAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that VLAN. Ethernet traffic is dropped on any VLAN which has an ID that overlaps with an FCoE VLAN ID.

VLAN 4048 is user configurable. However, Cisco UCS Manager uses VLAN 4048 for the following default values. If you want to assign 4048 to a VLAN, you must reconfigure these values:

- After an upgrade to Cisco UCS, Release 2.0—The FCoE storage port native VLAN uses VLAN 4048 by default. If the default FCoE VSAN was set to use VLAN 1 before the upgrade, you must change it to a VLAN ID that is not used or reserved. For example, consider changing the default to 4049 if that VLAN ID is not in use.
- After a fresh install of Cisco UCS, Release 2.0—The FCoE VLAN for the default VSAN uses VLAN 4048 by default. The FCoE storage port native VLAN uses VLAN 4049.

The VLAN name is case sensitive.

VLAN Port Limitations

Cisco UCS Manager limits the number of VLAN port instances that you can configure under border and server domains on a fabric interconnect.

Types of Ports Included in the VLAN Port Count

The following types of ports are counted in the VLAN port calculation:

- Border uplink Ethernet ports
- Border uplink Ether-channel member ports
- FCoE ports in a SAN cloud
- Ethernet ports in a NAS cloud
- Static and dynamic vNICs created through service profiles
- VM vNICs created as part of a port profile in a hypervisor in hypervisor domain

Based on the number of VLANs configured for these ports, Cisco UCS Manager tracks the cumulative count of VLAN port instances and enforces the VLAN port limit during validation. Cisco UCS Manager reserves some pre-defined VLAN port resources for control traffic. These include management VLANs configured under HIF and NIF ports.

VLAN Port Limit Enforcement

Cisco UCS Manager validates VLAN port availability during the following operations:

- Configuring and unconfiguring border ports and border port channels
- Adding or removing VLANs from a cloud
- Configuring or unconfiguring SAN or NAS ports
- Associating or disassociating service profiles that contain configuration changes
- Configuring or unconfiguring VLANs under vNICs or vHBAs
- Receiving creation or deletion notifications from a VMWare vNIC and from an ESX hypervisor

Note This is outside the control of the Cisco UCS Manager.

- Fabric interconnect reboot
- Cisco UCS Manager upgrade or downgrade

Cisco UCS Manager strictly enforces the VLAN port limit on service profile operations. If Cisco UCS Manager detects that the VLAN port limit is exceeded, the service profile configuration fails during deployment.

Exceeding the VLAN port count in a border domain is less disruptive. When the VLAN port count is exceeded in a border domain Cisco UCS Manager changes the allocation status to Exceeded. To change the status back to **Available**, complete one of the following actions:

- Unconfigure one or more border ports
- Remove VLANs from the LAN cloud
- Unconfigure one or more vNICs or vHBAs

Configuring Named VLANs

Creating a Named VLAN Accessible to Both Fabric Interconnects (Uplink Ethernet Mode)

Important VLAN IDs **3915 to 4042** and **4043 to 4047** are reserved. You cannot use these VLAN IDs to create new VLANs.

For Cisco UCS 6600 Series Fabric Interconnect, Cisco UCS Fabric Interconnects 9108 100G, Cisco UCS 6500 and 6400 Series Fabric Interconnects, VLAN IDs from 1002 to 1005 are reserved for VLAN Trunking Protocol (VTP).

The VLAN IDs you specify must also be supported on the switch that you are using. For example, on Cisco Nexus 5000 Series switches, the VLAN ID range from 3968 to 4029 is reserved. Before you specify the VLAN IDs in Cisco UCS Manager, make sure that the same VLAN IDs are available on your switch.

VLANs in the LAN cloud and FCoE VLANs in the SAN cloud must have different IDs. Using the same ID for a VLAN and an FCoE VLAN in a VSAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that VLAN. Ethernet traffic is dropped on any VLAN which has an ID that overlaps with an FCoE VLAN ID.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **create vlan** *vlan-name vlan-id*	Creates a named VLAN, specifies the VLAN name and VLAN ID, and enters Ethernet uplink VLAN mode. The VLAN name is case sensitive.
Step 3	UCS-A /eth-uplink/fabric/vlan # **set sharing** {**isolated** | **none** | **primary**}	Sets the sharing for the specified VLAN. This can be one of the following: • **isolated** —This is a secondary VLAN associated with a primary VLAN. This VLAN is private. • **none** —This VLAN does not have any secondary or private VLANs. • **primary** —This VLAN can have one or more secondary VLANs.
Step 4	UCS-A /eth-uplink/vlan # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example creates a named VLAN for both fabric interconnects, names the VLAN accounting, assigns the VLAN ID 2112, sets the sharing to none, and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # create vlan accounting 2112
UCS-A /eth-uplink/vlan* # set sharing none
UCS-A /eth-uplink/vlan* # commit-buffer
UCS-A /eth-uplink/vlan #
```

Creating a Named VLAN Accessible to Both Fabric Interconnects (Ethernet Storage Mode)

Important VLAN IDs **3915 to 4042** and **4043 to 4047** are reserved. You cannot use these VLAN IDs to create new VLANs.

For Cisco UCS 6600 Series Fabric Interconnect, Cisco UCS Fabric Interconnects 9108 100G, Cisco UCS 6500 and 6400 Series Fabric Interconnects, VLAN IDs from 1002 to 1005 are reserved for VLAN Trunking Protocol (VTP).

The VLAN IDs you specify must also be supported on the switch that you are using. For example, on Cisco Nexus 5000 Series switches, the VLAN ID range from 3968 to 4029 is reserved. Before you specify the VLAN IDs in Cisco UCS Manager, make sure that the same VLAN IDs are available on your switch.

VLANs in the LAN cloud and FCoE VLANs in the SAN cloud must have different IDs. Using the same ID for a VLAN and an FCoE VLAN in a VSAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that VLAN. Ethernet traffic is dropped on any VLAN which has an ID that overlaps with an FCoE VLAN ID.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-storage**	Enters Ethernet storage mode.
Step 2	UCS-A /eth-storage # **create vlan** *vlan-name vlan-id*	Creates a named VLAN, specifies the VLAN name and VLAN ID, and enters Ethernet storage VLAN mode. The VLAN name is case sensitive.
Step 3	UCS-A /eth-storage/vlan # **create member-port** {**a** | **b**} *slot-id port-id*	Creates a member port for the specified VLAN on the specified fabric.
Step 4	UCS-A /eth-storage/vlan/member-port # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example creates a named VLAN for both fabric interconnects, names the VLAN accounting, assigns the VLAN ID 2112, creates a member port on slot 2, port 20, and commits the transaction:

```
UCS-A# scope eth-storage
UCS-A /eth-storage # create vlan accounting 2112
UCS-A /eth-storage/vlan* # create member-port a 2 20
UCS-A /eth-storage/vlan/member-port* # commit-buffer
UCS-A /eth-storage/vlan/member-port #
```

Creating a Named VLAN Accessible to One Fabric Interconnect (Uplink Ethernet Mode)

Important VLAN IDs **3915 to 4042** and **4043 to 4047** are reserved. You cannot use these VLAN IDs to create new VLANs.

For Cisco UCS 6600 Series Fabric Interconnect, Cisco UCS Fabric Interconnects 9108 100G, Cisco UCS 6500 and 6400 Series Fabric Interconnects, VLAN IDs from 1002 to 1005 are reserved for VLAN Trunking Protocol (VTP).

The VLAN IDs you specify must also be supported on the switch that you are using. For example, on Cisco Nexus 5000 Series switches, the VLAN ID range from 3968 to 4029 is reserved. Before you specify the VLAN IDs in Cisco UCS Manager, make sure that the same VLAN IDs are available on your switch.

VLANs in the LAN cloud and FCoE VLANs in the SAN cloud must have different IDs. Using the same ID for a VLAN and an FCoE VLAN in a VSAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that VLAN. Ethernet traffic is dropped on any VLAN which has an ID that overlaps with an FCoE VLAN ID.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric** {**a** | **b**}	Enters Ethernet uplink fabric interconnect mode for the specified fabric interconnect (A or B).
Step 3	UCS-A /eth-uplink/fabric # **create vlan** *vlan-name* *vlan-id*	Creates a named VLAN, specifies the VLAN name and VLAN ID, and enters Ethernet uplink fabric interconnect VLAN mode. The VLAN name is case sensitive.
Step 4	UCS-A /eth-uplink/fabric/vlan # **set sharing** {**isolated** | **none** | **primary**}	Sets the sharing for the specified VLAN. This can be one of the following:

	Command or Action	Purpose
		• **isolated** —This is a secondary VLAN associated with a primary VLAN. This VLAN is private. • **none** —This VLAN does not have any secondary or private VLANs. • **primary** —This VLAN can have one or more secondary VLANs.
Step 5	UCS-A /eth-uplink/fabric/vlan # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example creates a named VLAN for fabric interconnect A, names the VLAN finance, assigns the VLAN ID 3955, sets the sharing to none, and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric a
UCS-A /eth-uplink/fabric # create vlan finance 3955
UCS-A /eth-uplink/fabric/vlan* # set sharing none
UCS-A /eth-uplink/fabric/vlan* # commit-buffer
UCS-A /eth-uplink/fabric/vlan #
```

Creating a Secondary VLAN for a Private VLAN (Accessible to One Fabric Interconnect)

Important VLAN IDs **3915 to 4042** and **4043 to 4047** are reserved. You cannot use these VLAN IDs to create new VLANs.

For Cisco UCS 6600 Series Fabric Interconnect, Cisco UCS Fabric Interconnects 9108 100G, Cisco UCS 6500 and 6400 Series Fabric Interconnects, VLAN IDs from 1002 to 1005 are reserved for VLAN Trunking Protocol (VTP).

The VLAN IDs you specify must also be supported on the switch that you are using. For example, on Cisco Nexus 5000 Series switches, the VLAN ID range from 3968 to 4029 is reserved. Before you specify the VLAN IDs in Cisco UCS Manager, make sure that the same VLAN IDs are available on your switch.

VLANs in the LAN cloud and FCoE VLANs in the SAN cloud must have different IDs. Using the same ID for a VLAN and an FCoE VLAN in a VSAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that VLAN. Ethernet traffic is dropped on any VLAN which has an ID that overlaps with an FCoE VLAN ID.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric** {**a** | **b**}	Enters Ethernet uplink fabric interconnect mode for the specified fabric interconnect (A or B).
Step 3	UCS-A /eth-uplink/fabric # **create vlan** *vlan-name* *vlan-id*	Creates a named VLAN, specifies the VLAN name and VLAN ID, and enters Ethernet uplink fabric interconnect VLAN mode. The VLAN name is case sensitive.
Step 4	UCS-A /eth-uplink/fabric/vlan # **set sharing isolated**	Sets the VLAN as the secondary VLAN.
Step 5	UCS-A /eth-uplink/fabric/vlan # **set pubnwname** *primary-vlan-name*	Specifies the primary VLAN to be associated with this secondary VLAN.
Step 6	UCS-A /eth-uplink/fabric/vlan/member-port # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example creates a named VLAN for fabric interconnect A, names the VLAN finance, assigns the VLAN ID 3955, makes this VLAN the secondary VLAN, associates the secondary VLAN with the primary VLAN, and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric a
UCS-A /eth-uplink/fabric # create vlan finance 3955
UCS-A /eth-uplink/fabric/vlan* # set sharing isolated
UCS-A /eth-uplink/fabric/vlan* # set pubnwname pvlan1000
UCS-A /eth-uplink/fabric/vlan* # commit-buffer
UCS-A /eth-uplink/fabric/vlan #
```

Deleting a Named VLAN

If Cisco UCS Manager includes a named VLAN with the same VLAN ID as the one you delete, the VLAN is not removed from the fabric interconnect configuration until all named VLANs with that ID are deleted.

If you are deleting a private primary VLAN, ensure that you reassign the secondary VLANs to another working primary VLAN.

Before you begin

Before you delete a VLAN from a fabric interconnect, ensure that the VLAN was removed from all vNICs and vNIC templates.

Note If you delete a VLAN that is assigned to a vNIC or vNIC template, the vNIC might allow that VLAN to flap.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	(Optional) UCS-A /eth-uplink # **scope fabric** {**a** | **b**}	Enters Ethernet uplink fabric mode. Use this command when you want to delete a named VLAN only from the specified fabric (a or b).
Step 3	UCS-A /eth-uplink # **delete vlan** *vlan-name*	Deletes the specified named VLAN.
Step 4	UCS-A /eth-uplink # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example deletes a named VLAN accessible to both fabric interconnects and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # delete vlan accounting
UCS-A /eth-uplink* # commit-buffer
UCS-A /eth-uplink #
```

The following example deletes a named VLAN accessible to one fabric interconnect and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric a
UCS-A /eth-uplink/fabric # delete vlan finance
UCS-A /eth-uplink/fabric* # commit-buffer
UCS-A /eth-uplink/fabric #
```

Configuring Private VLANs

Creating a Primary VLAN for a Private VLAN (Accessible to Both Fabric Interconnects)

Important VLAN IDs **3915 to 4042** and **4043 to 4047** are reserved. You cannot use these VLAN IDs to create new VLANs.

For Cisco UCS 6600 Series Fabric Interconnect, Cisco UCS Fabric Interconnects 9108 100G, Cisco UCS 6500 and 6400 Series Fabric Interconnects, VLAN IDs from 1002 to 1005 are reserved for VLAN Trunking Protocol (VTP).

The VLAN IDs you specify must also be supported on the switch that you are using. For example, on Cisco Nexus 5000 Series switches, the VLAN ID range from 3968 to 4029 is reserved. Before you specify the VLAN IDs in Cisco UCS Manager, make sure that the same VLAN IDs are available on your switch.

VLANs in the LAN cloud and FCoE VLANs in the SAN cloud must have different IDs. Using the same ID for a VLAN and an FCoE VLAN in a VSAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that VLAN. Ethernet traffic is dropped on any VLAN which has an ID that overlaps with an FCoE VLAN ID.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **create vlan** *vlan-name vlan-id*	Creates a named VLAN, specifies the VLAN name and VLAN ID, and enters Ethernet uplink VLAN mode. The VLAN name is case sensitive.
Step 3	UCS-A /eth-uplink/vlan # **set sharing primary**	Sets the VLAN as the primary VLAN.
Step 4	UCS-A /eth-uplink/vlan # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example creates a named VLAN for both fabric interconnects, names the VLAN accounting, assigns the VLAN ID 2112, makes this VLAN the primary VLAN, and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # create vlan accounting 2112
UCS-A /eth-uplink/vlan* # set sharing primary
```

```
UCS-A /eth-uplink/vlan* # commit-buffer
UCS-A /eth-uplink/vlan #
```

Creating a Primary VLAN for a Private VLAN (Accessible to One Fabric Interconnect)

Important VLAN IDs **3915 to 4042** and **4043 to 4047** are reserved. You cannot use these VLAN IDs to create new VLANs.

For Cisco UCS 6600 Series Fabric Interconnect, Cisco UCS Fabric Interconnects 9108 100G, Cisco UCS 6500 and 6400 Series Fabric Interconnects, VLAN IDs from 1002 to 1005 are reserved for VLAN Trunking Protocol (VTP).

The VLAN IDs you specify must also be supported on the switch that you are using. For example, on Cisco Nexus 5000 Series switches, the VLAN ID range from 3968 to 4029 is reserved. Before you specify the VLAN IDs in Cisco UCS Manager, make sure that the same VLAN IDs are available on your switch.

VLANs in the LAN cloud and FCoE VLANs in the SAN cloud must have different IDs. Using the same ID for a VLAN and an FCoE VLAN in a VSAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that VLAN. Ethernet traffic is dropped on any VLAN which has an ID that overlaps with an FCoE VLAN ID.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric** {**a** | **b**}	Enters Ethernet uplink fabric interconnect mode for the specified fabric interconnect.
Step 3	UCS-A /eth-uplink/fabric # **create vlan** *vlan-name* *vlan-id*	Creates a named VLAN, specifies the VLAN name and VLAN ID, and enters Ethernet uplink fabric interconnect VLAN mode. The VLAN name is case sensitive.
Step 4	UCS-A /eth-uplink/fabric/vlan # **set sharing primary**	Sets the VLAN as the primary VLAN.
Step 5	UCS-A /eth-uplink/fabric/vlan # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example creates a named VLAN for fabric interconnect A, names the VLAN finance, assigns the VLAN ID 3955, makes this VLAN the primary VLAN, and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric a
```

```
UCS-A /eth-uplink/fabric # create vlan finance 3955
UCS-A /eth-uplink/fabric/vlan* # set sharing primary
UCS-A /eth-uplink/fabric/vlan* # commit-buffer
UCS-A /eth-uplink/fabric/vlan #
```

Creating a Secondary VLAN for a Private VLAN (Accessible to Both Fabric Interconnects)

Important VLAN IDs **3915 to 4042** and **4043 to 4047** are reserved. You cannot use these VLAN IDs to create new VLANs.

For Cisco UCS 6600 Series Fabric Interconnect, Cisco UCS Fabric Interconnects 9108 100G, Cisco UCS 6500 and 6400 Series Fabric Interconnects, VLAN IDs from 1002 to 1005 are reserved for VLAN Trunking Protocol (VTP).

The VLAN IDs you specify must also be supported on the switch that you are using. For example, on Cisco Nexus 5000 Series switches, the VLAN ID range from 3968 to 4029 is reserved. Before you specify the VLAN IDs in Cisco UCS Manager, make sure that the same VLAN IDs are available on your switch.

VLANs in the LAN cloud and FCoE VLANs in the SAN cloud must have different IDs. Using the same ID for a VLAN and an FCoE VLAN in a VSAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that VLAN. Ethernet traffic is dropped on any VLAN which has an ID that overlaps with an FCoE VLAN ID.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **create vlan** *vlan-name vlan-id*	Creates a named VLAN, specifies the VLAN name and VLAN ID, and enters Ethernet uplink VLAN mode. The VLAN name is case sensitive.
Step 3	UCS-A /eth-uplink/vlan # **set sharing isolated**	Sets the VLAN as the secondary VLAN.
Step 4	UCS-A /eth-uplink/vlan # **set pubnwname** *primary-vlan-name*	Specifies the primary VLAN to be associated with this secondary VLAN.
Step 5	UCS-A /eth-uplink/vlan # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example creates a named VLAN for both fabric interconnects, names the VLAN accounting, assigns the VLAN ID 2112, makes this VLAN the secondary VLAN, associates the secondary VLAN with the primary VLAN, and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # create vlan accounting 2112
UCS-A /eth-uplink/vlan* # set sharing isolated
UCS-A /eth-uplink/vlan* # set pubnwname pvlan1000
UCS-A /eth-uplink/vlan* # commit-buffer
UCS-A /eth-uplink/vlan #
```

Creating a Secondary VLAN for a Private VLAN (Accessible to One Fabric Interconnect)

Important VLAN IDs **3915 to 4042** and **4043 to 4047** are reserved. You cannot use these VLAN IDs to create new VLANs.

For Cisco UCS 6600 Series Fabric Interconnect, Cisco UCS Fabric Interconnects 9108 100G, Cisco UCS 6500 and 6400 Series Fabric Interconnects, VLAN IDs from 1002 to 1005 are reserved for VLAN Trunking Protocol (VTP).

The VLAN IDs you specify must also be supported on the switch that you are using. For example, on Cisco Nexus 5000 Series switches, the VLAN ID range from 3968 to 4029 is reserved. Before you specify the VLAN IDs in Cisco UCS Manager, make sure that the same VLAN IDs are available on your switch.

VLANs in the LAN cloud and FCoE VLANs in the SAN cloud must have different IDs. Using the same ID for a VLAN and an FCoE VLAN in a VSAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that VLAN. Ethernet traffic is dropped on any VLAN which has an ID that overlaps with an FCoE VLAN ID.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric** {**a** | **b**}	Enters Ethernet uplink fabric interconnect mode for the specified fabric interconnect (A or B).
Step 3	UCS-A /eth-uplink/fabric # **create vlan** *vlan-name* *vlan-id*	Creates a named VLAN, specifies the VLAN name and VLAN ID, and enters Ethernet uplink fabric interconnect VLAN mode. The VLAN name is case sensitive.
Step 4	UCS-A /eth-uplink/fabric/vlan # **set sharing isolated**	Sets the VLAN as the secondary VLAN.
Step 5	UCS-A /eth-uplink/fabric/vlan # **set pubnwname** *primary-vlan-name*	Specifies the primary VLAN to be associated with this secondary VLAN.
Step 6	UCS-A /eth-uplink/fabric/vlan/member-port # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example creates a named VLAN for fabric interconnect A, names the VLAN finance, assigns the VLAN ID 3955, makes this VLAN the secondary VLAN, associates the secondary VLAN with the primary VLAN, and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric a
UCS-A /eth-uplink/fabric # create vlan finance 3955
UCS-A /eth-uplink/fabric/vlan* # set sharing isolated
UCS-A /eth-uplink/fabric/vlan* # set pubnwname pvlan1000
UCS-A /eth-uplink/fabric/vlan* # commit-buffer
UCS-A /eth-uplink/fabric/vlan #
```

Allowing PVLANs on vNICs

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org /**	Enters root organization mode.
Step 2	UCS-A /org # **scope service-profile** *profile-name*	Commits the transaction to the system configuration.
Step 3	UCS-A /org/service-profile # **scope vnic** *vnic-name*	Enters command mode for the specified vNIC.
Step 4	UCS-A /org/service-profile/vnic # **create eth-if** *community-vlan-name*	Allows the community VLAN to access the specified vNIC.
Step 5	UCS-A /org/service-profile/vnic/eth-if* # **exit**	Exits the interface configuration mode for the specified vNIC.
Step 6	UCS-A /org/service-profile/vnic* # **create eth-if** *primary-vlan-name*	Allows the primary VLAN to access the specified vNIC.
Step 7	UCS-A /org/service-profile/vnic # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to assign the community VLAN cVLAN102 and the primary VLAN primaryVLAN100 to the vNIC vnic_1 and commits the transaction.

```
UCS-A# scope org /
UCS-A /org # scope service-profile GSP1
UCS-A /org/service-profile # scope vnic vnic_1
UCS-A /org/service-profile/vnic # create eth-if cVLAN102
UCS-A /org/service-profile/vnic/eth-if* # exit
UCS-A /org/service-profile/vnic # create eth-if primaryVLAN100
UCS-A /org/service-profile/vnic* # commit-buffer
```

Creating a Primary VLAN for a Private VLAN on an Appliance Cloud

Important VLAN IDs **3915 to 4042** and **4043 to 4047** are reserved. You cannot use these VLAN IDs to create new VLANs.

For Cisco UCS 6600 Series Fabric Interconnect, Cisco UCS Fabric Interconnects 9108 100G, Cisco UCS 6500 and 6400 Series Fabric Interconnects, VLAN IDs from 1002 to 1005 are reserved for VLAN Trunking Protocol (VTP).

The VLAN IDs you specify must also be supported on the switch that you are using. For example, on Cisco Nexus 5000 Series switches, the VLAN ID range from 3968 to 4029 is reserved. Before you specify the VLAN IDs in Cisco UCS Manager, make sure that the same VLAN IDs are available on your switch.

VLANs in the LAN cloud and FCoE VLANs in the SAN cloud must have different IDs. Using the same ID for a VLAN and an FCoE VLAN in a VSAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that VLAN. Ethernet traffic is dropped on any VLAN which has an ID that overlaps with an FCoE VLAN ID.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-storage**	Enters Ethernet storage mode.
Step 2	UCS-A /eth-storage # **create vlan** *vlan-name vlan-id*	Creates a named VLAN, specifies the VLAN name and VLAN ID, and enters Ethernet storage VLAN mode. The VLAN name is case sensitive.
Step 3	UCS-A /eth-storage/vlan* # **set sharing primary**	Sets the VLAN as the primary VLAN.
Step 4	UCS-A /eth-storage/vlan* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example creates a named VLAN for fabric interconnect A, names the VLAN, assigns the VLAN ID, makes this VLAN the primary VLAN, and commits the transaction:

```
UCS-A# scope eth-storage
UCS-A /eth-storage # create vlan primaryvlan500 500
UCS-A /eth-storage/vlan* # set sharing primary
UCS-A /eth-storage/vlan* # commit-buffer
UCS-A /eth-storage/vlan #
```

Creating a Secondary VLAN for a Private VLAN on an Appliance Cloud

Important VLAN IDs **3915 to 4042** and **4043 to 4047** are reserved. You cannot use these VLAN IDs to create new VLANs.

For Cisco UCS 6600 Series Fabric Interconnect, Cisco UCS Fabric Interconnects 9108 100G, Cisco UCS 6500 and 6400 Series Fabric Interconnects, VLAN IDs from 1002 to 1005 are reserved for VLAN Trunking Protocol (VTP).

The VLAN IDs you specify must also be supported on the switch that you are using. For example, on Cisco Nexus 5000 Series switches, the VLAN ID range from 3968 to 4029 is reserved. Before you specify the VLAN IDs in Cisco UCS Manager, make sure that the same VLAN IDs are available on your switch.

VLANs in the LAN cloud and FCoE VLANs in the SAN cloud must have different IDs. Using the same ID for a VLAN and an FCoE VLAN in a VSAN results in a critical fault and traffic disruption for all vNICs and uplink ports using that VLAN. Ethernet traffic is dropped on any VLAN which has an ID that overlaps with an FCoE VLAN ID.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-storage**	Enters Ethernet storage mode.
Step 2	UCS-A /eth-storage # **create vlan** *vlan-name vlan-id*	Creates a named VLAN, specifies the VLAN name and VLAN ID, and enters Ethernet storage VLAN mode. The VLAN name is case sensitive.
Step 3	UCS-A /eth-storage/vlan* # **set sharing isolated**	Sets the VLAN as the secondary VLAN.
Step 4	UCS-A /eth-storage/vlan* # **set pubnwname** *primary-vlan-name*	Specifies the primary VLAN to be associated with this secondary VLAN.
Step 5	UCS-A /eth-storage/vlan* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example creates a named VLAN for fabric interconnect A, names the VLAN, assigns the VLAN ID, makes this VLAN the secondary VLAN, associates the secondary VLAN with the primary VLAN, and commits the transaction:

```
UCS-A# scope eth-storage
UCS-A /eth-storage # create vlan isovlan501 501
UCS-A /eth-storage/vlan* # set sharing isolated
UCS-A /eth-storage/vlan* # set pubnwname primaryvlan500
UCS-A /eth-storage/vlan* # commit-buffer
UCS-A /eth-storage/vlan #  #
```

Community VLANs

Cisco UCS Manager supports Community VLANs in UCS Fabric Interconnects. Community ports communicate with each other and with promiscuous ports. Community ports have Layer 2 isolation from all other ports in other communities, or isolated ports within the PVLAN. Broadcasts are transmitted between the community ports associated with the PVLAN only and the other promiscuous ports. A promiscuous port can communicate with all interfaces, including the isolated and community ports within a PVLAN.

Creating a Community VLAN

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**.	Enters Ethernet uplink mode.
Step 2	UCS-A# /eth-uplink/ # **create vlan** *ID* .	Create a VLAN with the specified VLAN ID. **Note** VLAN with IDs from 1002 to 1005 are reserved for NX-OS in Cisco UCS Fabric Interconnects 9108 100G and Cisco UCS 6500 Series Fabric Interconnects.
Step 3	UCS-A# /eth-uplink/ vlan # **set sharing** *Type* .	Specifies the VLAN type.
Step 4	UCS-A# /eth-uplink/ vlan # **set pubnwname** *Name* .	Specifies the primary VLAN association.
Step 5	UCS-A# /eth-uplink/ vlan # **commit-buffer**.	Commits the transaction to the system configuration.

Example

The following example shows how to create a Community VLAN:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # create vlan vlan203 203
UCS-A /eth-uplink/vlan* # set sharing community
UCS-A /eth-uplink/vlan* # set pubname vlan200
UCS-A /eth-uplink/vlan* # commit-buffer
UCS-A /eth-uplink/vlan* # exit
UCS-A /vlan-group #
```

Viewing Community VLANS

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters Cisco UCS Manager organization.
Step 2	UCS-A /org # **show vlan**	Displays the available groups in the organization.

Example

The following example shows the available VLAN groups in the root org:

```
UCS-A# scope org
UCS-A# /org/# show vlan
VLAN Group:

    Name         VLAN ID         Fabric ID     Native VLAN    Sharing Type    Primary Vlan
    ------------------------------------------------------------------------------------

    vlan100      100             Dual          No             Primary         vlan100
    vlan100      101             Dual          No             Isolated        vlan100
    vlan100      203             Dual          No             Community       vlan200
```

Allowing Community VLANs on vNICs

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*.
Step 2	UCS-A /org # **scope service-profile** *profile-name*	Commits the transaction to the system configuration.
Step 3	UCS-A /org/service-profile # **scope vnic** *vnic-name*	Enters command mode for the specified vNIC.
Step 4	UCS-A /org/service-profile/vnic # **create eth-if** *community-vlan-name*	Allows the community VLAN to access the specified vNIC.
Step 5	UCS-A /org/service-profile/vnic # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to assign the community VLAN cVLAN101 to the vNIC vnic_1 and commits the transaction.

```
UCS-A# scope org /
UCS-A /org # scope service-profile GSP1
UCS-A /org/service-profile # scope vnic vnic_1
UCS-A /org/service-profile/vnic # create eth-if cVLAN101
UCS-A /org/service-profile/vnic* # commit-buffer
```

Allowing PVLAN on Promiscuous Access or Trunk Port

For a promiscuous access port, the isolated and community VLANs must be associated to the same primary VLAN.

For a promiscuous trunk port, isolated and community VLANs belonging to different primary VLANs are allowed, as well as regular VLANs.

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope eth-storage**	Enters Ethernet storage mode.
Step 2	UCS-A /eth-storage # **scope vlan** *iso-vlan-name*	Enters the specified isolated VLAN.
Step 3	UCS-A /eth-storage/vlan # **create member-port** *fabric slot- num port- num*	Creates the member port for the specified fabric, assigns the slot number and port number, and enters member port configuration scope.
Step 4	UCS-A /eth-storage/vlan/member-port # **exit**	Returns to VLAN mode.
Step 5	UCS-A /eth-storage/vlan # **exit**	Returns to Ethernet storage mode.
Step 6	UCS-A /eth-storage # **scope vlan** *comm-vlan-name*	Enters the specified community VLAN.
Step 7	UCS-A /eth-storage/vlan # **create member-port** *fabric slot- num port- num*	Creates the member port for the specified fabric, assigns the slot number and port number, and enters member port configuration scope.
Step 8	UCS-A /eth-storage/vlan/member-port # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to assign the isolated and community associated with the same primary VLAN to the same appliance port and commits the transaction.

```
UCS-A# scope eth-storage
UCS-A /eth-storage # scope vlan isovlan501
UCS-A /eth-storage/vlan # create member-port a 1 2
```

```
UCS-A /eth-storage/vlan/member-port* # exit
UCS-A /eth-storage/vlan* # exit
UCS-A /eth-storage* # scope vlan cvlan502
UCS-A /eth-storage/vlan* # create member-port a 1 2
UCS-A /eth-storage/vlan/member-port* # commit-buffer
UCS-A /eth-storage/vlan/member-port #
```

Deleting a Community VLAN

If Cisco UCS Manager includes a named VLAN with the same VLAN ID as the one you delete, the VLAN is not removed from the fabric interconnect configuration until all named VLANs with that ID are deleted.

If you are deleting a private primary VLAN, ensure that you reassign the secondary VLANs to another working primary VLAN.

Before you begin

Before you delete a VLAN from a fabric interconnect, ensure that the VLAN was removed from all vNICs and vNIC templates.

Note If you delete a VLAN that is assigned to a vNIC or vNIC template, the vNIC might allow that VLAN to flap.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	(Optional) UCS-A /eth-uplink # **scope fabric** {**a** | **b**}	Enters Ethernet uplink fabric mode. Use this command when you want to delete a named VLAN only from the specified fabric (a or b).
Step 3	UCS-A /eth-uplink # **delete community vlan** *vlan-name*	Deletes the specified community VLAN.
Step 4	UCS-A /eth-uplink # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example deletes a Community VLAN and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # delete community vlan vlan203
UCS-A /eth-uplink* # commit-buffer
UCS-A /eth-uplink #
```

Viewing the VLAN Port Count

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fabric-interconnect** {**a** | **b**}	Enters fabric interconnect mode for the specified fabric interconnect.
Step 2	UCS-A /fabric-interconnect # **show vlan-port-count**	Displays the VLAN port count.

Example

The following example displays the VLAN port count for fabric interconnect A:

```
UCS-A# scope fabric-interconnect a
UCS-A /fabric-interconnect # show vlan-port-count

VLAN-Port Count:
VLAN-Port Limit     Access VLAN-Port Count     Border VLAN-Port Count     Alloc Status
----------              ---------------           ----------------          ----------
6000                          3                         0                      Available
```

VLAN Port Count Optimization

VLAN port count optimization enables mapping the state of multiple VLANs into a single internal state. When you enable the VLAN port count optimization, Cisco UCS Manager logically groups VLANs based on the port VLAN membership. This grouping increases the port VLAN count limit. VLAN port count optimization also compresses the VLAN state and reduces the CPU load on the fabric interconnect. This reduction in the CPU load enables you to deploy more VLANs over more vNICs. Optimizing VLAN port count does not change any of the existing VLAN configuration on the vNICs.

VLAN port count optimization is disabled by default. You can enable or disable the option based on your requirements.

Important

- Enabling VLAN port count optimization increases the number of available VLAN ports for use. If the port VLAN count exceeds the maximum number of VLANs in a non-optimized state, you cannot disable the VLAN port count optimization.
- On the Cisco UCS Fabric Interconnects, the VLAN port count optimization is performed when the PV count exceeds 16000.

Note When the Cisco UCS 6400 Series Fabric Interconnect is in Ethernet switching mode:

- The FI does not support **VLAN Port Count Optimization Enabled**
- The FI supports 16000 PVs, similar to EHM mode, when **VLAN Port Count Optimization** is **Disabled**.

The following table illustrates the Port VLAN (PV) Count with VLAN port count optimization enabled and disabled:

Fabric Interconnect Model	PV Count with VLAN Port Count Optimization Disabled	PV Count with VLAN Port Count Optimization Enabled
Cisco UCS 6400 Series FI (6454 FI & 64108 FI)	16000	108000
Cisco UCS 6500 Series FI (6536 FI)	16000	108000
Cisco UCS Fabric Interconnects 9108 100G (UCS X-Series Direct/UCSX-S9108-100G)	16000	108000
Cisco UCS 6600 Series Fabric Interconnect (6664 FI)	16000	108000

Enabling Port VLAN Count Optimization

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink# **set vlan-port-count-optimization enable**	Enables VLAN Port Count Optimization.
Step 3	UCS-A /eth-uplink* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows the fabric port-channel vHBA reset configuration:

```
Sample Command:
UCS-A# scope eth-uplink
UCS-A /eth-uplink # set vlan-port-count-optimization enable
UCS-A /eth-uplink* # commit-buffer
UCS-A /eth-uplink#
------------------
Sample Output:

Ethernet Uplink:
Mode: End Host
MAC Table Aging Time (dd:hh:mm:ss): Mode Default
VLAN Port Count Optimization: Enabled
Fabric Port Channel vHBA reset: Disabled
service for unsupported transceivers: Disabled
```

Disabling Port VLAN Count Optimization

If you have more Port VLAN count than that is allowed in the non port VLAN port count optimization state, you cannot disable the optimization.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink# **set vlan-port-count-optimization disable**	Disables the port VLAN count optimization.
Step 3	UCS-A /eth-uplink # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to disable VLAN port count optimization:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # set vlan-port-count-optimization disable
UCS-A /eth-uplink* # commit-buffer
UCS-A /eth-uplink#
```

Viewing the Port VLAN Count Optimization Groups

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink# **show vlan-port-count-optimization group**	Displays the vlan for port VLAN count optimization groups.

Example

The following example shows port VLAN count optimization group in fabric a and b:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # show vlan-port-count-optimization group
VLAN Port Count Optimization Group:
    Fabric ID  Group ID   VLAN ID
    --------   -------    -------
    A          5          6
    A          5          7
    A          5          8
    B          10         100
    B          10         101
```

VLAN Groups

VLAN groups allow you to group VLANs on Ethernet uplink ports, by function or by VLANs that belong to a specific network. You can define VLAN membership and apply the membership to multiple Ethernet uplink ports on the fabric interconnect.

Note Cisco UCS Manager supports a maximum of 200 VLAN Groups. If Cisco UCS Manager determines that you create more than 200 VLAN groups, the system disables VLAN compression.

You can configure inband and out-of-band (OOB) VLAN groups to use to access the Cisco Integrated Management Interface (CIMC) on blade and rack servers. Cisco UCS Manager supports OOB IPv4 and inband IPv4 and IPv6 VLAN groups for use with the uplink interfaces or uplink port channels.

Note Inband Management is not supported on VLAN 2 or VLAN 3.

After you assign a VLAN to a VLAN group, any changes to the VLAN group are applied to all Ethernet uplink ports that are configured with the VLAN group. The VLAN group also enables you to identify VLAN overlaps between disjoint VLANs.

You can configure uplink ports under a VLAN group. When you configure an uplink port for a VLAN group, that uplink port will support all the VLANs that are part of the associated VLAN groups and individual VLANs

that are associated with the uplink using LAN Uplinks Manager, if any. Further, any uplink that is not selected for association with that VLAN group will stop supporting the VLANs that are part of that VLAN group.

You can create VLAN groups from the **LAN Cloud** or from the **LAN Uplinks Manager**.

Creating a VLAN Group

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**.	Enters Ethernet uplink mode. The VLAN Group name is case sensitive.
Step 2	UCS-A# /eth-uplink/ #**create vlan-group***Name* .	Create a VLAN group with the specified name. This name can be between 1 and 32 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and you cannot change this name after the object is saved.
Step 3	UCS-A# /eth-uplink/ vlan-group#**create member-vlan***ID* .	Adds the specified VLANs to the created VLAN group.
Step 4	UCS-A# /eth-uplink/vlan-group #**create member-port** [member-port-channel] .	Assigns the uplink Ethernet ports to the VLAN group.
Step 5	UCS-A#/vlan-group* # **commit-buffer**.	Commits the transaction to the system configuration.

Example

The following example shows how to create a VLAN group:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # create vlan-group eng
UCS-A /eth-uplink/vlan-group* # create member-vlan 3
UCS-A /eth-uplink/vlan-group* # commit-buffer
UCS-A /vlan-group #
```

Creating an Inband VLAN Group

Configure inband VLAN groups to provide access to remote users via an inband service profile.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth uplink**	Enters Ethernet uplink configuration mode.
Step 2	UCS-A /eth-uplink # **create vlan-group** *inband-vlan-name*	Creates a VLAN group with the specified name and enters VLAN group configuration mode.
Step 3	UCS-A /eth-uplink/vlan-group # **create member-vlan***inband-vlan-nameinband-vlan-id*	Adds the specified VLAN to the VLAN group and enters VLAN group member configuration mode.
Step 4	UCS-A /eth-uplink/vlan-group/member-vlan # **exit**	Exits VLAN group member configuration mode.
Step 5	UCS-A /eth-uplink/vlan-group # **create member-port***fabricslot-numport-num*	Creates the member port for the specified fabric, assigns the slot number, and port number and enters member port configuration.
Step 6	UCS-A /eth-uplink/vlan-group/member-port # **commit-buffer**	Commits the transaction.

Example

The example below creates a VLAN group named inband-vlan-group, creates a member of the group named Inband_VLAN and assigns VLAN ID 888, creates member ports for Fabric A and Fabric B, and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # create vlan-group inband-vlan-group
UCS-A /eth-uplink/vlan-group* # create member-vlan Inband_VLAN 888
UCS-A /eth-uplink/vlan-group/member-vlan* # exit
UCS-A /eth-uplink/vlan-group* # create member-port a 1 23
UCS-A /eth-uplink/vlan-group/member-port* # exit
UCS-A /eth-uplink/vlan-group* # create member-port b 1 23
UCS-A /eth-uplink/vlan-group/member-port* # commit-buffer
UCS-A /eth-uplink/vlan-group/member-port # exit
UCS-A /eth-uplink/vlan-group # exit
```

What to do next

Assign the inband VLAN group to an inband service profile.

Viewing VLAN Groups

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters Cisco UCS Manager organization.

	Command or Action	Purpose
Step 2	UCS-A /org # **show vlan-group**	Displays the available groups in the organization.

Example

The following example shows the available VLAN groups in the root org:

```
UCS-A# scope org
UCS-A# /org/# show vlan-group
VLAN Group:
    Name
    ----
    eng
    hr
    finance
```

Deleting a VLAN Group

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**.	Enters Ethernet uplink mode.
Step 2	UCS-A# /eth-uplink/ #**delete vlan-group***Name* .	Deletes the specified VLAN group.
Step 3	UCS-A#/eth-uplink* # **commit-buffer**.	Commits the transaction to the system configuration.

Example

The following example shows how to delete a VLAN group:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # delete vlan-group eng
UCS-A /eth-uplink* # commit-buffer
UCS-A /eth-uplink #
```

Modifying the Reserved VLAN

This task describes how to modify the reserved VLAN ID. Modifying the reserved VLAN makes transitioning from Cisco UCS 6200 Series Fabric Interconnects to the Cisco UCS 6454 Fabric Interconnect more flexible with preexisting network configurations. The reserved VLAN block is configurable by assigning a contiguous block of 128 unused VLANs, rather than reconfiguring the currently existing VLANs that conflict with the default range. For example, if the reserved VLAN is changed to 3912, then the new VLAN block range spans 3912 to 4039. You can select any contiguous block of 128 VLAN IDs, with the start ID ranging from 2 to

3915. Changing the reserved VLAN requires a reload of the 6454 Fabric Interconnect for the new values to take effect.

For Cisco UCS 6600, 6500 Series Fabric Interconnects, VLAN IDs from 1002 to 1005 are reserved for NX-OS.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**.	Enters Ethernet uplink mode.
Step 2	UCS-A# /eth-uplink/ #**show reserved-vlan** .	This displays the reserved VLAN IDs.
Step 3	UCS-A# /eth-uplink/ #**scope reserved-vlan**.	Enters reserved VLAN ID specification mode.
Step 4	UCS-A# /eth-uplink/reserved-vlan #**set start-vlan-id** [vlan-id] .	Assigns the new reserved VLAN starting ID. The reserved VLAN range ID can be specified from 2-3915.
Step 5	UCS-A# /eth-uplink/reserved-vlan* # **commit-buffer**.	Commits the transaction to the system configuration.

Example

The following example shows how to modify the reserved VLAN ID:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # show reserved-vlan
UCS-A /eth-uplink/ # scope reserved-vlan
UCS-A /eth-uplink/reserved-vlan # set start-vlan-id 3912
UCS-A /eth-uplink/reserved-vlan/* # commit-buffer
```

VLAN Permissions

VLAN permissions restrict access to VLANs based on specified organizations and on the service profile organizations to which the VLANs belong. VLAN permissions also restrict the set of VLANs that you can assign to service profile vNICs. VLAN permissions is an optional feature and is disabled by default. You can enable or disable the feature based on your requirements. If you disable the feature, all of the VLANs are globally accessible to all organizations.

Note If you enable the org permission in **LAN** > **LAN Cloud** > **Global Policies** > **Org Permissions**, when you create a VLAN, the **Permitted Orgs for VLAN(s)** option displays in the **Create VLANs** dialog box. If you do not enable the **Org Permissions**, the **Permitted Orgs for VLAN(s)** option does not display.

Enabling the org permission allows you to specify the organizations for the VLAN. When you specify the organizations, the VLAN becomes available to that specific organization and all of the sub organizations below the structure. Users from other organizations cannot access this VLAN. You can also modify the VLAN permission anytime based on changes to your VLAN access requirements.

Caution When you assign the VLAN org permission to an organization at the root level, all sub organizations can access the VLANs. After assigning the org permission at the root level, and you change the permission for a VLAN that belongs to a sub organization, that VLAN becomes unavailable to the root level organization.

Creating VLAN Permissions

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**.	Enters the Cisco UCS Manager VLAN organization.
Step 2	UCS-A# /org/ #**create vlan-permit***VLAN permission name*.	Creates the specified VLAN permission and assigns VLAN access permission to the organization.
Step 3	UCS-A#/org* # **commit-buffer**.	Commits the transaction to the system configuration.

Example

The following example shows how to create a VLAN permission for an organization:

```
UCS-A# scope org
UCS-A /org # create vlan-permit dev
UCS-A /org* # commit-buffer
UCS-A /org #
```

Viewing VLAN Permissions

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters Cisco UCS Manager organization.
Step 2	UCS-A /org # **show vlan-permit**	Displays the available permissions in the organization.

Example

The following example shows the VLAN groups that have permission to access this VLAN:

```
UCS-A# scope org
UCS-A# /org/# show vlan-permit
```

```
VLAN Group:
    Name
    ----
    eng
    hr
    finance
```

Deleting a VLAN Permission

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**.	Enters the Cisco UCS Manager VLAN organization.
Step 2	UCS-A# /org/ #**delete vlan-permit***VLAN permission name*.	Deletes the access permission to the VLAN.
Step 3	UCS-A#/org* # **commit-buffer**.	Commits the transaction to the system configuration.

Example

The following example shows how to delete a VLAN permission from an organization:

```
UCS-A# scope org
UCS-A /org # delete vlan-permit dev
UCS-A /org* # commit-buffer
UCS-A /org #
```

Fabric Port-Channel vHBA

A virtual host bus adapter (vHBA) logically connects a virtual machine to a virtual interface on the fabric interconnect and allows the virtual machine to send and receive traffic through that interface. This is currently accomplished by using the fibre channel modes (end-host mode/switch mode).

The port-channel operations that involves addition or removal of a member link between fabric interconnect and I/O Module (IOM). Such operations may result in a long I/O pause or connection drop from virtual machines to its targets and require a vHBA reset support

With the fabric port-channel vHBA reset is set to enabled, when the Cisco UCS IOM port-channel membership changes, the fabric interconnect sends a Registered State Change Notification (RSCN) packet to each vhba configured via that Cisco UCS IOM. The RSCN enables the virtual interface card (VIC) or VIC Driver to reset the fabric port-channel vHBA and to restore the connectivity.

By default, the fabric port-channel vHBA reset is set to disabled. This configuration supports additional bandwidth and provides greater resilience.

Enabling Fabric Port Channel vHBA reset

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink# **set fabric-pc-vhba-reset enabled**	Sets the fabric port-channel vHBA reset state as enabled.
Step 3	UCS-A /eth-uplink* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to enable fabric port-channel vHBA reset:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # set fabric-pc-vhba-reset enabled
UCS-A /eth-uplink* # commit-buffer
UCS-A /eth-uplink#
```

Disabling fabric port channel vHBA reset

You can disable the fabric port-channel vHBA reset.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink# **set fabric-pc-vhba-reset disabled**	Sets the fabric port-channel vHBA reset state as disabled. This is the default state.
Step 3	UCS-A /eth-uplink # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to disable the fabric port-channel vHBA reset:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # set fabric-pc-vhba-reset disabled
UCS-A /eth-uplink* # commit-buffer
UCS-A /eth-uplink#
```

Viewing the Fabric Port Channel vHBA Reset

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink# **show detail**	Displays the fabric port-channel vHBA reset configuration.

Example

The following example shows the fabric port-channel vHBA reset configuration:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # show detail

Ethernet Uplink:
    Mode: End Host
    MAC Table Aging Time (dd:hh:mm:ss): Mode Default
    VLAN Port Count Optimization: Disabled
    Fabric Port Channel vHBA reset: Disabled
    service for unsupported transceivers: Disabled
```

VIC QinQ Tunneling

Starting with release 4.3(2a), Cisco UCS Manager introduces support for VIC Q-in-Q tunneling configuration. A Q-in-Q (802.1Q-in-802.1Q) tunnel allows to segregate the traffic in the infrastructure and helps to expand the VLAN space through the addition of 802.1Q tag to 802.1Q-tagged packets.

To configure VIC QinQ Tunneling, ensure **Q-in-Q Forwarding** is enabled. For more information, see Q-in-Q Forwarding, on page 55.

To know more about supported combinations and limitations of VIC QinQ Tunneling: see VIC QinQ Tunneling - Supported Combinations and Limitations, on page 139.

Enabling and Managing QinQ

Enabling QinQ on a vNIC of a Service Profile

To enable QinQ on a vNIC in a service profile, do the following:

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters root organization mode.

	Command or Action	Purpose
Step 2	UCS-A /org **scope service-profile profile name**	Enters service-profile specified.
Step 3	UCS-A /org **scope vnic vnic 02**	Enters command mode for the specified vNIC.
Step 4	UCS-A /org/service-profile/vnic **Set QinQ** {**enabled** | **disabled** } **enabled**	QinQ is enabled on the specified vNIC *vnic 02*. **Note** QinQ VLAN selection on a vNIC is considered only when *Set QinQ* is enabled. For more information, see Adding a VLAN on a vNIC of a Service Profile, on page 142.
Step 5	UCS-A /org/service-profile/vnic/ **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to set QinQ on the vNIC17 in the service profile SP3 and commits the transaction:

```
UCS-A# scope org
UCS-A /org # scope service-profile SP3
UCS-A /org/service-profile # scope vnic vnic17
UCS-A /org/service-profile/vnic* #Set QinQ Enabled
UCS-A /org/service-profile/vnic* #commit-buffer
UCS-A /org/service-profile/vnic #
```

Disabling QinQ on a vNIC of a Service Profile

To disable QinQ on a vNIC of a service profile, do the following:

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters root organization mode.
Step 2	UCS-A /org **scope service-profile profile name**	Enters service-profile specified.
Step 3	UCS-A /org **scope vnic vnic 01**	Enters command mode for the specified vNIC *vnic 01*.
Step 4	UCS-A /org/service-profile/vnic **Set QinQ** {**enabled** | **disabled** } **disabled**	QinQ is disabled on the specified vNIC. **Note** QinQ VLAN selection on a vNIC is considered only when QinQ is enabled. Hence, ensure to re-enable QinQ VLAN when required. For

	Command or Action	Purpose
		more information, see Enabling QinQ on a vNIC of a Service Profile, on page 132.
Step 5	UCS-A /org/service profile/vnic/ **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to disable *QinQ VLAN* on the *vNIC 33* in the service profile *SP1* and commits the transaction:

```
UCS-A# scope org
UCS-A /org # scope service-profile SP1
UCS-A /org/service-profile # scope vnic vnic 33
UCS-A /org/service-profile/vnic* #QinQ Offload disabled
UCS-A /org/service-profile/vnic* #commit-buffer
UCS-A /org/service-profile/vnic # show detail
------------------
Vnic:
Name: vnic 33
Type: Initial Template
Fabric ID: A
Target: Adapter
Host Interface Ethernet MTU: 1500
CDN Source: vNIC Name
Ethernet Interface Admin CDN Name:
MAC Pool:
Oper MAC Pool:
Pin Group:
QoS Policy:
Oper QoS Policy:
Network Control Policy:
Oper Network Control Policy: org-root/nwctrl-default
Stats Policy: default
Oper Stats Policy: org-root/thr-policy-default
Policy Owner: Local
Redundancy Type: No Redundancy
Redundancy Peer Template Name:
Oper Redundancy Peer Template Name:
QinQ Offload: Disabled
```

Enabling QinQ on a vNIC of LAN Connectivity Policy

To enable QinQ on a vNIC through LAN Connectivity Policy, do the following:

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters root organization mode.
Step 2	UCS-A /org **scope lann conn policy LAN Policy 12**	Enters LAN Connectivity Policy **LAN Policy 12**.

	Command or Action	Purpose
Step 3	UCS-A /org **scope vnic vnic 01**	Enters command mode of the vNIC **vnic 01**.
Step 4	UCS-A /org/lann conn pol/vnic **Set QinQ** {**enabled** | **disabled** } **enabled**	QinQ is enabled on the specified vNIC. **Note** QinQ VLAN selection on a vNIC is considered only when *Set QinQ* is enabled. For more information, see Adding a VLAN on a vNIC of LAN Connectivity Policy, on page 141.
Step 5	UCS-A /org/lann conn pol/vnic/ **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to set QinQ on the vNIC17 in the *LAN Connectivity Policy 22* and commits the transaction:

```
UCS-A# scope org
UCS-A /org # scope lann conn policy 22
UCS-A /org/lann conn pol # scope vnic vnic17
UCS-A /org/lann conn pol/vnic* #Set QinQ Enabled
UCS-A /org/lann conn pol/vnic* #commit-buffer
UCS-A /org/lann conn pol/vnic #
```

Disabling QinQ on a vNIC of LAN Connectivity Policy

To disable QinQ VLAN on a vNIC through LAN Connectivity Policy, do the following:

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters root organization mode.
Step 2	UCS-A /org **scope lann conn policy LAN Policy 12**	Enters Lan Connectivity Policy **LAN Policy 12**.
Step 3	UCS-A /org/lann conn pol **scope vnic vnic 01**	Enters command mode for the specified vNIC.
Step 4	UCS-A /org/lann conn pol/vnic **Set QinQ** {**enabled** | **disabled** } **disabled**	QinQ is disabled on the specified vNIC. **Note** QinQ VLAN selection on a vNIC is considered only when QinQ is enabled. Hence, ensure to re-enable QinQ when required. For more information, see Enabling QinQ on a vNIC of LAN Connectivity Policy, on page 134.

	Command or Action	Purpose
Step 5	UCS-A /org/lann conn pol/vnic/ **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to disable QinQ on the vNIC 33 in the Lan Connectivity Policy *LP1* and commits the transaction:

```
UCS-A# scope org
UCS-A /org # scope lann conn policy LP1
UCS-A /org/lann conn pol # scope vnic vnic 33
UCS-A /org/lann conn pol/vnic* #QinQ Offload disabled
UCS-A /org/lann conn pol/vnic* #commit-buffer
UCS-A /org/lann conn pol/vnic # show detail
------------------
Vnic:
Name: vnic 33
Type: Initial Template
Fabric ID: A
Target: Adapter
Host Interface Ethernet MTU: 1500
CDN Source: vNIC Name
Ethernet Interface Admin CDN Name:
MAC Pool:
Oper MAC Pool:
Pin Group:
QoS Policy:
Oper QoS Policy:
Network Control Policy:
Oper Network Control Policy: org-root/nwctrl-default
Stats Policy: default
Oper Stats Policy: org-root/thr-policy-default
Policy Owner: Local
Redundancy Type: No Redundancy
Redundancy Peer Template Name:
Oper Redundancy Peer Template Name:
QinQ Offload: Disabled
```

Enabling QinQ on a vNIC Template

To enable QinQ on a specified vNIC template, do the following:

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters root organization mode.
Step 2	UCS-A /org **scope vnic-templ 22**	Enters command mode for the specified vNIC template.
Step 3	UCS-A /org/vnic-templ/eth-if # **Set QinQ** {**enabled** | **disabled** } **enabled**	QinQ is enabled on the specified vNIC template. **Note**

	Command or Action	Purpose
		QinQ VLAN selection on a vNIC is considered only when *Set QinQ* is enabled. For more information, see Adding a VLAN on a vNIC Template, on page 140.
Step 4	UCS-A /org/vnic-templ/eth-if # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example adds a VLAN 10 on the vNIC template 01, sets the VLAN as a native VLAN, enables QinQ on the vNIC, and commits the transaction:

```
UCS-A# scope org
UCS-A /org # scope vnic-templ 01
UCS-A /org/vnic-templ/eth-if# set qinq enabled
UCS-A /org/service-profile/eth-if* #commit-buffer
UCS-A /org/vnic-templ/eth-if
```

Disabling QinQ on a vNIC Template

To disable QinQ on a specified vNIC template, do the following:

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters root organization mode.
Step 2	UCS-A /org **scope vnic-templ 22**	Enters command mode for the specified vNIC template.
Step 3	UCS-A /org/vnic-templ/eth-if # **Set QinQ** {**enabled** | **disabled** } **disabled**	QinQ is disabled on the specified vNIC template. **Note** QinQ VLAN selection on a vNIC is considered only when QinQ is enabled. Hence, ensure to re-enable QinQ when required. For more information, see Enabling QinQ on a vNIC Template, on page 136.
Step 4	UCS-A /org/vnic-templ/eth-if # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to disable QinQ on the vNIC template 01 and commits the transaction:

```
UCS-A# scope org
UCS-A /org # scope vnic-templ 01
UCS-A /org/vnic-templ/eth-if# set qinq disabled
UCS-A /org/service-profile/eth-if* #commit-buffer
UCS-A /org/lann conn pol/vnic # show detail
------------------
Vnic:
Name: vnic 33
Type: Initial Template
Fabric ID: A
Target: Adapter
Host Interface Ethernet MTU: 1500
CDN Source: vNIC Name
Ethernet Interface Admin CDN Name:
MAC Pool:
Oper MAC Pool:
Pin Group:
QoS Policy:
Oper QoS Policy:
Network Control Policy:
Oper Network Control Policy: org-root/nwctrl-default
Stats Policy: default
Oper Stats Policy: org-root/thr-policy-default
Policy Owner: Local
Redundancy Type: No Redundancy
Redundancy Peer Template Name:
Oper Redundancy Peer Template Name:
QinQ Offload: Disabled
```

Viewing QinQ

To view QinQ VLAN on vNIC of a service profile, do the following:

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters root organization mode.
Step 2	UCS-A /org **scope service-profile profile name**	Enters service-profile specified.
Step 3	UCS-A /org **scope vnic vnic 01**	Enters command mode for the specified vNIC.
Step 4	UCS-A /org/service-profile/vnic # **show detail**	Displays the details including QinQ configuration status on the vNIC.

Example

The following example shows how to view the QinQ configuration status on vNIC through service profile. The example output displays QinQ Offload status as Enabled:

```
UCS-A# scope org
UCS-A /org # scope service-profile SP1
UCS-A /org/service-profile # scope vnic vnic 01
UCS-A /org/service-profile/vnic* #Show detail
UCS-A /org/service-profile/vnic #
----------------------------------------
Vnic:
Name: vnic01
Type: Initial Template
Fabric ID: A
Target: Adapter
Host Interface Ethernet MTU: 1500
CDN Source: vNIC Name
Ethernet Interface Admin CDN Name:
MAC Pool:
Oper MAC Pool:
Pin Group:
QoS Policy:
Oper QoS Policy:
Network Control Policy:
Oper Network Control Policy: org-root/nwctrl-default
Stats Policy: default
Oper Stats Policy: org-root/thr-policy-default
Policy Owner: Local
Redundancy Type: No Redundancy
Redundancy Peer Template Name:
Oper Redundancy Peer Template Name:
QinQ Offload: Enabled
```

Important You can use the **show detail** command to view the QinQ status on a vNIC Template and on a vNIC in a Lan Connectivity Policy.

VIC QinQ Tunneling - Supported Combinations and Limitations

Following are the supported combinations for VIC QinQ Tunneling:

- QinQ VLAN selection is considered only when the **Enable QinQ** check box is selected on a vNIC Interface.
- QinQ Configuration supports a maximum of two VLANs on a vNIC Interface. A QinQ VLAN can be a Native or a non-Native VLAN. You can configure a Native VLAN and a Non-Native VLAN as a QinQ VLAN on the vNIC.

  When using the Native VLAN as QinQ VLAN, no additional VLAN can be configured on the vNIC.
- For Cisco UCS VIC 15000 series adapters, QinQ and Geneve Offload can be enabled on a vNIC Interface.

Following are the limitations of VIC QinQ Tunneling:

- QinQ configuration on a vNIC Interface is not supported on Cisco UCS VIC 1300 series adapters.

- The default VLAN (VLAN ID: 1) is not supported as a QinQ VLAN on a vNIC Interface.
- When a Native VLAN and a QinQ VLAN are configured on a vNIC Interface, a new VLAN configuration is not supported and results in Server Profile association failures when selected. To accommodate a new VLAN, either the Native VLAN or QinQ VLAN must be removed.
- When the QinQ VLAN is the same as the Native VLAN on a vNIC Interface, a new VLAN configuration is not supported and results in Server Profile association failures when selected. To accommodate a new VLAN, either the Native VLAN or QinQ VLAN must be modified.
- For Cisco UCS VIC 1400, 14000, and 15000 series adapters, LAN (or PXE) Boot and QinQ cannot be configured on a vNIC interface and result in configuration failures when enabled.
- For Cisco UCS VIC 1400, 14000, and 15000 series adapters, iSCSI Boot and QinQ cannot be configured on a vNIC interface and result in configuration failures when enabled.
- For Cisco UCS VIC 1400 and 14000 series adapters, QinQ and Geneve Offload cannot be configured on a vNIC interface and result in configuration failures when enabled.
- For Cisco UCS VIC 1400, 14000, and 15000 series adapters, QinQ and VMMQ cannot be configured on a vNIC interface and result in configuration failures when enabled.
- For Cisco UCS VIC 1400, 14000, and 15000 series adapters, QinQ and RDMA V2 cannot be configured on a vNIC interface and result in configuration failures when enabled.
- For Cisco UCS 6454, 64108, 6536 Fabric Interconnects, Cisco UCS Fabric Interconnects 9108 100G, and Cisco UCS 6600 Series Fabric Interconnect, QinQ must be enabled at LAN > Global Policies to support QinQ VLAN on a VIC adapter.
- For Cisco UCS VIC 1400, 14000, and 15000 series adapters, QinQ and SR-IOV cannot be configured on a vNIC interface and result in configuration failures when enabled.
- When the Service Profile is already associated, you cannot enable or disable QinQ on a B-Series server.
- For Cisco UCS 6454, Cisco UCS 64108, Cisco UCS 6536 Fabric Interconnects, Cisco UCS Fabric Interconnects 9108 100G, and Cisco UCS 6600 Series Fabric Interconnect, QinQ configuration for Fabric Interconnects in Global Policy > LAN Connectivity Policy must be enabled to configure QinQ on a vNIC interface.
- QinQ and usNIC cannot be enabled together on a vNIC interface.
- When VIC QinQ Tunneling is enabled, you cannot downgrade to lower release versions.

Managing VLANs

Adding a VLAN on a vNIC Template

To create a VLAN on a vNIC template, do the following:

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters root organization mode.

	Command or Action	Purpose
Step 2	UCS-A /org # **scope vnic-templ 01**	Enters command mode for the specified vNIC template.
Step 3	UCS-A /org/vnic-templ/eth-if # **create eth-if** *vlan 20*	Creates a VLAN on the specified vNIC template. The VLAN name is case sensitive.
Step 4	UCS-A /org/vnic-templ/eth-if # **set default-net** {**yes** | **no** } **yes**	Sets the VLAN 10 as a Native VLAN on the vNIC template.
Step 5	UCS-A /org/vnic-templ/eth-if # **set qinq-vlan** {**yes** | **no** } **yes**	Enables VIC QinQ Tunneling on the vNIC Template. The supported QinQ VLAN ID range is 2 to 4094. A QinQ VLAN can be a Native or a Non-Native VLAN. You can configure a Native VLAN and a Non-Native VLAN as a QinQ VLAN on the vNIC. When using the Native VLAN as QinQ VLAN, no additional VLAN can be configured on the vNIC. **Note** VIC QinQ Tunneling is considered only when QinQ is enabled. For more information, see Enabling QinQ on a vNIC Template, on page 136.
Step 6	UCS-A /org/vnic-templ/vnic/eth-if # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example adds a VLAN 10 on the vNIC template 01, sets the VLAN as a native VLAN, enables QinQ on the VLAN, and commits the transaction:

```
UCS-A# scope org
UCS-A /org # scope vnic-templ 01
UCS-A /org/vnic-templ# create eth-if VLAN 10
UCS-A /org/vnic-templ/eth-if# set default-net yes
UCS-A /org/vnic-templ/eth-if# set qinq-vlan yes
UCS-A /org/vnic-templ/eth-if* #commit-buffer
UCS-A /org/vnic-templ/eth-if
```

Adding a VLAN on a vNIC of LAN Connectivity Policy

To add a VLAN on a vNIC of LAN Connectivity Policy, do the following:

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters root organization mode.
Step 2	UCS-A /org # **scope lann conn policy Lan Policy 01**	Enters LAN Connectivity Policy **Lan Policy 01**.
Step 3	UCS-A /org # **scope vnic vnic 11**	Enters command mode for the specified vNIC.
Step 4	UCS-A /org/lann conn policy/vnic **create eth-if** *vlan 10*	Creates a VLAN 10 on the specified vNIC. The VLAN name is case sensitive.
Step 5	UCS-A /org/lann conn policy/vnic/eth-if#/ **set default-net** {**yes** | **no** } **yes**	Sets the VLAN 10 as native VLAN in the service profile.
Step 6	UCS-A /org/lann conn policy/vnic/eth-if/ **set qinq-vlan** {**yes** | **no** } **yes**	Enables VIC QinQ Tunneling on the VLAN in the vNIC. The supported QinQ VLAN ID range is 2 to 4094. QinQ VLAN can be a Native or a Non-Native VLAN. You can configure a Native VLAN and a Non-Native VLAN as a QinQ VLAN on the vNIC. When using the Native VLAN as QinQ VLAN, no additional VLAN can be configured on the vNIC. **Note** VIC QinQ Tunneling is considered only when QinQ is enabled. For more information, see Enabling QinQ on a vNIC of LAN Connectivity Policy, on page 134.
Step 7	UCS-A /org/lann conn policy/vnic/eth-if/ **commit-buffer**	Commits the transaction to the system configuration.

Adding a VLAN on a vNIC of a Service Profile

To add a VLAN on a vNIC through service profile, do the following:

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters root organization mode.
Step 2	UCS-A /org # **scope service-profile profile name**	Enters service-profile named **profile name**

	Command or Action	Purpose
Step 3	UCS-A /org # **scope vnic vnic 01**	Enters command mode for the specified vNIC **vnic 01** .
Step 4	UCS-A /org/service profile/vnic **create eth-if** *vlan 10*	Creates VLAN 10 on the specified vNIC vnic 01. The VLAN name is case sensitive.
Step 5	UCS-A /org/service profile/vnic/eth-if#/ **set default-net** {**yes** | **no** } **yes**	Sets the VLAN 10 as native VLAN in the service profile.
Step 6	UCS-A /org/service profile/vnic/eth-if/ **set qinq-vlan** {**yes** | **no** } **yes**	Enables VIC QinQ Tunneling on the VLAN 10 in the vNIC 01. The supported QinQ VLAN ID range is 2 to 4094. QinQ VLAN can be a Native or a Non-Native VLAN. You can configure a Native VLAN and a Non-Native VLAN as a QinQ VLAN on the vNIC. When using the Native VLAN as QinQ VLAN, no additional VLAN can be configured on the vNIC. **Note** VIC QinQ Tunneling is considered only when QinQ is enabled. For more information, see Enabling QinQ on a vNIC of a Service Profile, on page 132.
Step 7	UCS-A /org/service profile/vnic/eth-if/ **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example creates a VLAN 20 on the vNIC 01 in a service profile, sets the VLAN as a native VLAN, enables QinQ on the VLAN, and commits the transaction:

```
UCS-A# scope org
UCS-A /org # scope service-profile SP1
UCS-A /org/service-profile # scope vnic vnic 01
UCS-A /org/service-profile/vnic # create eth-if VLAN 20
UCS-A /org/service-profile/vnic/eth-if* # set default-net no
UCS-A /org/service-profile/vnic/eth-if* # set qinq vlan yes
UCS-A /org/service-profile/vnic/eth-if* # exit
UCS-A /org/service-profile/vnic* #commit-buffer
UCS-A /org/service-profile/vnic#
```

Deleting a VLAN in a VNIC template

To delete a VLAN on the specified vNIC template, do the following:

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters root organization mode.
Step 2	UCS-A /org # **scope vnic-templ template01**	Enters vNIC template specified.
Step 3	UCS-A /org/vnic-templ/ # **delete eth-if vlan 33**	Deletes the VLAN 33 and its configuration on the specified vNIC template.
Step 4	UCS-A /org/vnic-templ/ # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to delete a VLAN 22 on the vNIC template 37, and commit the transaction:

```
UCS-A# scope org
UCS-A /org # scope vnic-templ template 37
UCS-A /org/vnic-templ/ # delete eth-if vlan 22
UCS-A /org/vnic-templ/ eth-if* # exit
UCS-A /org/vnic-templ* #commit-buffer
UCS-A /org/vnic-templ*
```

Deleting a VLAN on a vNIC of LAN Connectivity Policy

To delete a VLAN on a vNIC of LAN Connectivity Policy, do the following:

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters root organization mode.
Step 2	UCS-A /org # **scope lann conn policy lan policy 01**	Enters LAN Connectivity Policy lan policy 01 specified.
Step 3	UCS-A /org/lann conn policy # **scope vnic vnic 01**	Enters command mode for the specified vNIC *vnic 01*.
Step 4	UCS-A /org/lann conn policy/vnic # **delete eth-if vlan 23**	Deletes the VLAN 23 and its configuration on the specified vNIC.
Step 5	UCS-A /org/lann conn policy/vnic # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to delete a *VLAN 22* on the *vNIC 37* in a LAN Connectivity Policy *01* and commit the transaction:

```
UCS-A# scope org
UCS-A /org # scope lann conn policy Lan Policy 01
UCS-A /org/lann conn policy # scope vnic vnic 37
UCS-A /org/lann conn policy/vnic # delete eth-if VLAN 22
UCS-A /org/lann conn policy/vnic/eth-if* # exit
UCS-A /org/lann conn policy/vnic* #commit-buffer
UCS-A /org/lann conn policy/vnic#
```

Deleting a VLAN on a vNIC of a Service Profile

To delete a VLAN on the specified vNIC in a service profile, do the following:

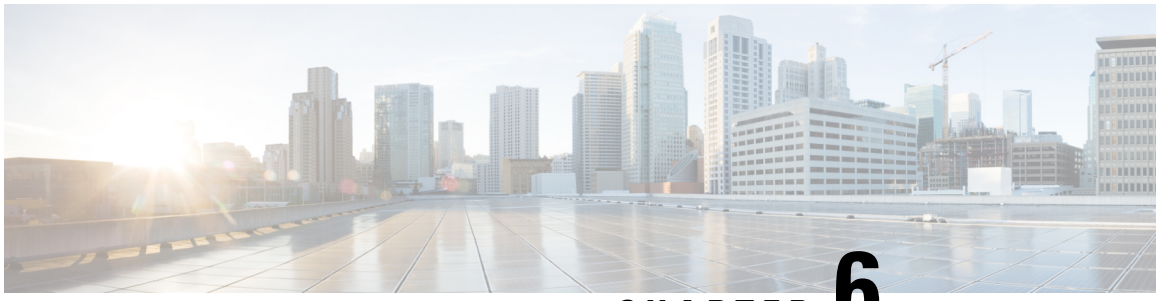
Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters root organization mode.
Step 2	UCS-A /org # **scope service-profile profile name**	Enters service-profile specified.
Step 3	UCS-A /org # **scope vnic vnic 01**	Enters command mode for the specified vNIC.
Step 4	UCS-A /org/service profile/vnic # **delete eth-if vlan 23**	Deletes the VLAN 23 and its configuration on the specified vNIC.
Step 5	UCS-A /org/service profile/vnic # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to delete a VLAN 22 on the vNIC 37 in a service profile SP2 and commit the transaction:

```
UCS-A# scope org
UCS-A /org # scope service-profile SP2
UCS-A /org/service-profile # scope vnic vnic 37
UCS-A /org/service-profile/vnic # delete eth-if VLAN 22
UCS-A /org/service-profile/vnic/eth-if* # exit
UCS-A /org/service-profile/vnic* #commit-buffer
UCS-A /org/service-profile/vnic#
```



CHAPTER 6

LAN PIN Groups

- LAN Pin Groups, on page 147
- Configuring a LAN Pin Group, on page 147

LAN Pin Groups

Cisco UCS uses LAN pin groups to pin Ethernet traffic from a vNIC on a server to an uplink Ethernet port or port channel on the fabric interconnect. You can use this pinning to manage the distribution of traffic from the servers.

To configure pinning for a server, you must include the LAN pin group in a vNIC policy. The vNIC policy is then included in the service profile assigned to that server. All traffic from the vNIC travels through the I/O module to the specified uplink Ethernet port.

Note If you do not assign a pin group to a server interface through a vNIC policy, Cisco UCS Manager chooses an uplink Ethernet port or port channel for traffic from that server interface dynamically. This choice is not permanent. A different uplink Ethernet port or port channel may be used for traffic from that server interface after an interface flap or a server reboot.

If an uplink is part of a LAN pin group, the uplink is not necessarily reserved for only that LAN pin group. Other vNIC's policies that do not specify a LAN pin group can use the uplink as a dynamic uplink.

Configuring a LAN Pin Group

In a system with two fabric interconnects, you can associate the pin group with only one fabric interconnect or with both fabric interconnects.

Before you begin

Configure the ports and port channels with which you want to configure the pin group. You can only include ports and port channels configured as uplink ports in a LAN pin group.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **create pin-group** *pin-group-name*	Creates an Ethernet (LAN) pin group with the specified name, and enters Ethernet uplink pin group mode.
Step 3	(Optional) UCS-A /eth-uplink/pin-group # **set descr** *description*	Provides a description for the pin group. **Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output.
Step 4	(Optional) UCS-A /eth-uplink/pin-group # **set target** {**a** | **b** | **dual**} {**port** *slot-num* / *port-num* | **port-channel** *port-num*}	Sets the Ethernet pin target to the specified fabric and port, or fabric and port channel.
Step 5	UCS-A /eth-uplink/pin-group # **commit-buffer**	Commits the transaction to the system configuration.

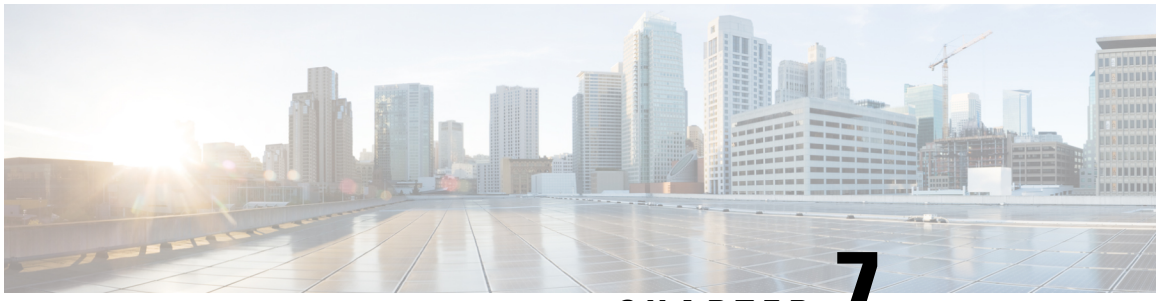
Example

The following example creates a LAN pin group named pingroup54 on fabric A, provides a description for the pin group, sets the pin group target to port channel 28, and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # create pin-group pingroup54
UCS-A /eth-uplink/pin-group* # set descr "This is my pin group #54"
UCS-A /eth-uplink/pin-group* # set target a port-channel 28
UCS-A /eth-uplink/pin-group* # commit-buffer
UCS-A /eth-uplink/pin-group #
```

What to do next

Include the pin group in a vNIC template.



CHAPTER 7

MAC Pools

- MAC Pools, on page 149
- Creating a MAC Pool, on page 149
- Deleting a MAC Pool, on page 151

MAC Pools

A MAC pool is a collection of network identities, or MAC addresses, that are unique in their Layer 2 environment and are available to be assigned to vNICs on a server. If you use MAC pools in service profiles, you do not have to manually configure the MAC addresses to be used by the server associated with the service profile.

In a system that implements multitenancy, you can use the organizational hierarchy to ensure that MAC pools can be used only by specific applications or business services. Cisco UCS uses the name resolution policy to assign MAC addresses from the pool.

To assign a MAC address to a server, you must include the MAC pool in a vNIC policy. The vNIC policy is then included in the service profile assigned to that server.

You can specify your own MAC addresses or use a group of MAC addresses provided by Cisco.

Creating a MAC Pool

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*.
Step 2	UCS-A /org # **create mac-pool** *mac-pool-name*	Creates a MAC pool with the specified name, and enters organization MAC pool mode. This name can be between 1 and 32 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen),

	Command or Action	Purpose
		_ (underscore), : (colon), and . (period), and you cannot change this name after the object is saved.
Step 3	(Optional) UCS-A /org/mac-pool # **set descr** *description*	Provides a description for the MAC pool. **Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output.
Step 4	UCS-A /org/mac-pool # **set assignmentorder** {**default** | **sequential**}	This can be one of the following: • **default**—Cisco UCS Manager selects a random identity from the pool. • **sequential**—Cisco UCS Manager selects the lowest available identity from the pool.
Step 5	UCS-A /org/mac-pool # **create block** *first-mac-addr last-mac-addr*	Creates a block (range) of MAC addresses, and enters organization MAC pool block mode. You must specify the first and last MAC addresses in the address range using the form *nn*:*nn*:*nn*:*nn*:*nn*:*nn*, with the addresses separated by a space. **Note** A MAC pool can contain more than one MAC address block. To create multiple MAC address blocks, you must enter multiple **create block** commands from organization MAC pool mode.
Step 6	UCS-A /org/mac-pool # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to create a MAC pool named pool37, provide a description for the pool, define a MAC address block by specifying the first and last MAC addresses in the block, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # create mac-pool pool37
UCS-A /org/mac-pool* # set descr "This is my MAC pool"
UCS-A /org/mac-pool* # create block 00:A0:D7:42:00:01 00:A0:D7:42:01:00
UCS-A /org/mac-pool/block* # commit-buffer
UCS-A /org/mac-pool/block #
```

What to do next

Include the MAC pool in a vNIC template.

Deleting a MAC Pool

If you delete a pool, Cisco UCS ManagerCisco UCS Central does not reallocate any addresses from that pool that were assigned to vNICs or vHBAs in Cisco UCS Manager. All assigned addresses from a deleted pool remain with the vNIC or vHBA to which they are assigned until one of the following occurs:

- The associated service profiles are deleted.
- The vNIC or vHBA to which the address is assigned is deleted.
- The vNIC or vHBA is assigned to a different pool.

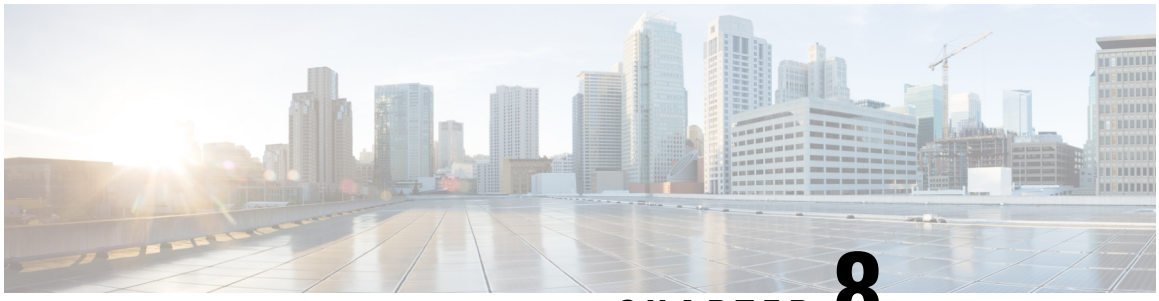
Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*.
Step 2	UCS-A /org # **delete mac-pool** *pool-name*	Deletes the specified MAC pool.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to delete the MAC pool named pool4 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # delete mac-pool pool4
UCS-A /org* # commit-buffer
UCS-A /org #
```



CHAPTER 8

Quality of Service

- Quality of Service, on page 153
- Configuring System Classes, on page 154
- Configuring Quality of Service Policies, on page 158
- Configuring Flow Control Policies, on page 160
- Configuring Slow Drain, on page 163
- Priority Flow Control Watchdog Interval, on page 166

Quality of Service

Cisco UCS provides the following methods to implement quality of service:

- System classes that specify the global configuration for certain types of traffic across the entire system
- QoS policies that assign system classes for individual vNICs
- Flow control policies that determine how uplink Ethernet ports handle pause frames

Global QoS changes made to the QoS system class may result in brief data-plane interruptions for all traffic. Some examples of such changes are:

- Changing the MTU size for an enabled class
- Changing packet drop for an enabled class
- Changing the CoS value for an enabled class

Guidelines and Limitations for Quality of Service on Cisco UCS 6600 Series Fabric Interconnect, Cisco UCS Fabric Interconnects 9108 100G, Cisco UCS 6536 Fabric Interconnects, Cisco UCS 6400 Series Fabric Interconnects

- Multicast optimization is not supported.
- For all QoS system classes except for Fibre Channel, the default MTU is 1500 bytes. The MTU for Fiber Channel class is not configurable and is set to 2240 bytes internally. All classes (excluding Fibre Channel) allow for MTU configuration up to a maximum of 9216 bytes.

Note The maximum MTU for a QoS class on the Fabric Interconnect is 9216 bytes, while the maximum MTU that can be set on a vNIC is 9158 bytes. The vNIC MTU is configured through the adapter policy.

- The MTU size for fibre channel is always 2240 bytes.
- Multicast is not supported on any no-drop QoS class.

Configuring System Classes

System Classes

Cisco UCS uses Data Center Ethernet (DCE) to handle all traffic inside a Cisco UCS domain. This industry standard enhancement to Ethernet divides the bandwidth of the Ethernet pipe into eight virtual lanes. Two virtual lanes are reserved for internal system and management traffic. You can configure quality of service (QoS) for the other six virtual lanes. System classes determine how the DCE bandwidth in these six virtual lanes is allocated across the entire Cisco UCS domain.

Each system class reserves a specific segment of the bandwidth for a specific type of traffic, which provides a level of traffic management, even in an oversubscribed system. For example, you can configure the **Fibre Channel Priority** system class to determine the percentage of DCE bandwidth allocated to FCoE traffic.

The following table describes the system classes that you can configure.

Table 8: System Classes

System Class	Description
Platinum Gold Silver Bronze	A configurable set of system classes that you can include in the QoS policy for a service profile. Each system class manages one lane of traffic. All properties of these system classes are available for you to assign custom settings and policies.
Best Effort	A system class that sets the quality of service for the lane reserved for basic Ethernet traffic. Some properties of this system class are preset and cannot be modified. For example, this class has a drop policy that allows it to drop data packets if required. The default MTU for the Best Effort class is 1500. You cannot disable this system class.

System Class	Description
Fibre Channel	A system class that sets the quality of service for the lane reserved for Fibre Channel over Ethernet traffic. Some properties of this system class are preset and cannot be modified. For example, this class has a no-drop policy that ensures it never drops data packets. You cannot disable this system class. **Note** FCoE traffic has a reserved QoS system class that should not be used by any other type of traffic. If any other type of traffic has a CoS value that is used by FCoE, the value is remarked to 0.

Configuring a System Class

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-server**	Enters Ethernet server mode.
Step 2	UCS-A /eth-server # **scope qos**	Enters Ethernet server QoS mode.
Step 3	UCS-A /eth-server/qos # **scope eth-classified** {**bronze** | **gold** | **platinum** | **silver**}	Enters Ethernet server QoS Ethernet classified mode for the specified system class.
Step 4	UCS-A /eth-server/qos/eth-classified # **enable**	Enables the specified system class.
Step 5	UCS-A /eth-server/qos/eth-classified # **set cos** *cos-value*	Specifies the class of service for the specified system class. Valid class of service values are 0 to 6. **Important** Use the same CoS values on UCS and N5K for all the no-drop policies. To insure that end-to-end PFC works correctly, have the same QoS policy configured on all intermediate switches. **Note** When the CoS value is set to 0 in any QoS class, this causes the adapter to use the same queue for best effort and the QoS class. When traffic congestion occurs, best effort and the QoS class will share the bandwidth equally instead of using the weight configured in the QoS class.
Step 6	UCS-A /eth-server/qos/eth-classified # **set drop** {**drop** | **no-drop**}	Specifies whether the channel can drop packets or not. For Cisco UCS Mini, packet drop can

	Command or Action	Purpose
		only be disabled on the platinum and gold classes. **Note** Changes saved to the drop displays the following warning message: Warning: The operation will cause momentary disruption to traffic forwarding.
Step 7	UCS-A /eth-server/qos/eth-classified # **set mtu** {*mtu-value* | **fc** | **normal**}	The maximum transmission unit, or packet size to be used. The maximum configurable MTU for a QoS system class is 9216 bytes. Fabric Interconnects (FI) are capable of forwarding packets with an MTU of up to 9216 bytes, as determined by the overall QoS configuration. However, the maximum configurable MTU for vNICs remains 9000 bytes. **Note** If the vNIC has an associated QoS policy, the MTU specified here must be equal to or less than the MTU specified in the associated QoS system class. If this MTU value exceeds the MTU value in the QoS system class, packets might get dropped during data transmission. **Important** Some properties may not be configurable for all system classes. The maximum configurable MTU for a QoS system class is 9216 bytes. However, the Fabric Interconnects can forward packets with an MTU of up to 9216 bytes, as determined by the overall QoS configuration. Changes saved to the MTU displays the following warning message: Warning: The operation will cause momentary disruption to traffic forwarding.
Step 8	UCS-A /eth-server/qos/eth-classified # **set weight** {*weight-value* | **best-effort** | **none**}	Specifies the relative weight for the specified system class. Valid weight values are 0 to 10.
Step 9	UCS-A /eth-server/qos/eth-classified # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to enable the platinum system class, allow the channel to drop packets, set the class of service to 6, set the MTU to normal, set the relative weight to 5, and commit the transaction:

```
UCS-A# scope eth-server
UCS-A /eth-server # scope qos
UCS-A /eth-server/qos # scope eth-classified platinum
UCS-A /eth-server/qos/eth-classified # enable
UCS-A /eth-server/qos/eth-classified* # set drop drop
Warning: The operation will cause momentary disruption to traffic forwarding
UCS-A /eth-server/qos/eth-classified* # set cos 6
UCS-A /eth-server/qos/eth-classified* # set mtu normal
Warning: The operation will cause momentary disruption to traffic forwarding
UCS-A /eth-server/qos/eth-classified* # set weight 5
UCS-A /eth-server/qos/eth-classified* # commit-buffer
UCS-A /eth-server/qos/eth-classified #
```

Disabling a System Class

If you disable a system class that is used in a QoS policy, Cisco UCS Manager uses the system class configured with CoS 0 for traffic on servers that are configured with the QoS policy. If no system class is configured as CoS 0, the Best Effort system class is used. You cannot disable the Best Effort or Fibre Channel system classes.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-server**	Enters Ethernet server mode.
Step 2	UCS-A /eth-server # **scope qos**	Enters Ethernet server QoS mode.
Step 3	UCS-A /eth-server/qos # **scope eth-classified** {**bronze** | **gold** | **platinum** | **silver**}	Enters Ethernet server QoS Ethernet classified mode for the specified system class.
Step 4	UCS-A /eth-server/qos/eth-classified # **disable**	Disables the specified system class.
Step 5	UCS-A /eth-server/qos/eth-classified # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example disables the platinum system class and commits the transaction:

```
UCS-A# scope eth-server
UCS-A /eth-server # scope qos
UCS-A /eth-server/qos # scope eth-classified platinum
UCS-A /eth-server/qos/eth-classified # disable
UCS-A /eth-server/qos/eth-classified* # commit-buffer
UCS-A /eth-server/qos/eth-classified #
```

Configuring Quality of Service Policies

Quality of Service Policy

A quality of service (QoS) policy assigns a system class to the outgoing traffic for a vNIC or vHBA. This system class determines the quality of service for that traffic. For certain adapters, you can also specify additional controls on the outgoing traffic, such as burst and rate.

You must include a QoS policy in a vNIC policy or vHBA policy and then include that policy in a service profile to configure the vNIC or vHBA.

Configuring a QoS Policy

Procedure

	Command or Action	Purpose
Step 1	Switch-A# **scope org** *org-name*	Enters org mode for the specified organization. To enter the default org mode, type **/** as the *org-name* .
Step 2	Switch-A /org # **create qos-policy** *policy-name*	Creates the specified QoS policy, and enters org QoS policy mode.
Step 3	Switch-A /org/qos-policy # **create egress-policy**	Creates the egress policy (for both vNICs and vHBAs) to be used by the QoS policy, and enters org QoS policy egress policy mode.
Step 4	Switch-A /org/qos-policy/egress-policy # **set host-cos-control** {**full** | **none**}	(Optional) Specifies whether the host or Cisco UCS Manager controls the class of service (CoS) for a vNIC. This setting has no effect on a vHBA. Use the **full** keyword to have the host control the CoS. If the packet has a valid CoS value, the host uses that value. Otherwise, it uses the CoS value associated with the specified class priority. Use the **none** keyword to have Cisco UCS Manager use the CoS value associated with the specified priority.
Step 5	Switch-A /org/qos-policy/egress-policy # **set prio** *sys-class-name*	Specifies the system class to be used for the egress policy. The *sys-class-name* argument can be one of the following class keywords: • **FC**—Use this priority for QoS policies that control vHBA traffic only. • **Platinum**—Use this priority for QoS policies that control vNIC traffic only.

	Command or Action	Purpose
		• **Gold**—Use this priority for QoS policies that control vNIC traffic only. • **Silver**—Use this priority for QoS policies that control vNIC traffic only. • **Bronze**—Use this priority for QoS policies that control vNIC traffic only. • **Best Effort**—Do not use this priority. It is reserved for the Basic Ethernet traffic lane. If you assign this priority to a QoS policy and configure another system class as CoS 0, Cisco UCS Manager does not default to this system class. It defaults to the priority with CoS 0 for that traffic.
Step 6	Switch-A /org/qos-policy/egress-policy # **set rate** {**line-rate** | *kbps*} **burst** *bytes*	Specifies the expected average rate of traffic. Traffic that falls under this rate will always conform. The default is **line-rate**, which equals a value of 10,000,000. The minimum value is 8, and the maximum value is 40,000,000. The Cisco Cisco UCS M81KR Virtual Interface Card, Cisco UCS VIC 1300 series, UCS VIC 1400 series, and UCS VIC 15000 series adapters support rate limiting on both vNICs and vHBAs. On the Cisco UCS VIC 1200 series adapters, rate limiting is supported only on vNICs.
Step 7	Switch-A /org/qos-policy/egress-policy # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example creates a QoS policy for vNIC traffic, assigns the platinum system class and sets the rate limit (traffic rate and burst size) for the egress policy, and commits the transaction:

```
Switch-A# scope org /
Switch-A /org # create qos-policy VnicPolicy34
Switch-A /org/qos-policy* # create egress-policy
Switch-A /org/qos-policy/egress-policy* # set prio platinum
Switch-A /org/qos-policy/egress-policy* # set rate 5000000 burst 65000
Switch-A /org/qos-policy/egress-policy* # commit-buffer
Switch-A /org/qos-policy/egress-policy #
```

The following example creates a QoS policy for vHBA traffic, assigns the fc (Fibre Channel) system class and sets the rate limit (traffic rate and burst size) for the egress policy, and commits the transaction:

```
Switch-A# scope org /
Switch-A /org # create qos-policy VhbaPolicy12
Switch-A /org/qos-policy* # create egress-policy
```

```
Switch-A /org/qos-policy/egress-policy* # set prio fc
Switch-A /org/qos-policy/egress-policy* # set rate 5000000 burst 65000
Switch-A /org/qos-policy/egress-policy* # commit-buffer
Switch-A /org/qos-policy/egress-policy #
```

What to do next

Include the QoS policy in a vNIC or vHBA template.

Deleting a QoS Policy

If you delete a QoS policy that is in use or you disable a system class that is used in a QoS policy, any vNIC or vHBA that uses that QoS policy is assigned to the Best Effort system class or to the system class with a CoS of 0. In a system that implements multitenancy, Cisco UCS Manager first attempts to find a matching QoS policy in the organization hierarchy.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 2	UCS-A /org # **delete qos-policy** *policy-name*	Deletes the specified QoS policy.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following deletes the QoS policy named QosPolicy34 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # delete qos-policy QosPolicy34
UCS-A /org* # commit-buffer
UCS-A /org #
```

Configuring Flow Control Policies

Flow Control Policy

Flow control policies determine whether the uplink Ethernet ports in a Cisco UCS domain send and receive IEEE 802.3x pause frames when the receive buffer for a port fills. These pause frames request that the transmitting port stop sending data for a few milliseconds until the buffer clears.

For flow control to work between a LAN port and an uplink Ethernet port, you must enable the corresponding receive and send flow control parameters for both ports. For Cisco UCS, the flow control policies configure these parameters.

When you enable the send function, the uplink Ethernet port sends a pause request to the network port if the incoming packet rate becomes too high. The pause remains in effect for a few milliseconds before traffic is reset to normal levels. If you enable the receive function, the uplink Ethernet port honors all pause requests from the network port. All traffic is halted on that uplink port until the network port cancels the pause request.

Because you assign the flow control policy to the port, changes to the policy have an immediate effect on how the port reacts to a pause frame or a full receive buffer.

Configuring a Flow Control Policy

Before you begin

Configure the network port with the corresponding setting for the flow control that you need. For example, if you enable the send setting for flow-control pause frames in the policy, ensure that the receive parameter in the network port is set to on or to desired. If you want the Cisco UCS port to receive flow-control frames, ensure that the send parameter is set to on or to desire on the network port. If you do not want to use flow control, you can set the send and receive parameters on the network port to off.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope flow-control**	Enters Ethernet uplink flow control mode.
Step 3	UCS-A /eth-uplink/flow-control # **create policy** *policy-name*	Creates the specified flow control policy.
Step 4	UCS-A /eth-uplink/flow-control/policy # **set prio** *prio-option*	Specifies one of the following flow control priority options: • **auto** —The Cisco UCS system and the network negotiate whether PPP will be used on this fabric interconnect. • **on** —PPP is enabled on this fabric interconnect.
Step 5	UCS-A /eth-uplink/flow-control/policy # **set receive** *receive-option*	Specifies one of the following flow control receive options: • **off** —Pause requests from the network are ignored and traffic flow continues as normal. • **on** —Pause requests are honored and all traffic is halted on that uplink port until the network cancels the pause request.

	Command or Action	Purpose
Step 6	UCS-A /eth-uplink/flow-control/policy # **set send** *send-option*	Specifies one of the following flow control send options: • **off** —Traffic on the port flows normally regardless of the packet load. • **on** —The Cisco UCS system sends a pause request to the network if the incoming packet rate becomes too high. The pause remains in effect for a few milliseconds before traffic is reset to normal levels.
Step 7	UCS-A /eth-uplink/flow-control/policy # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following configures a flow control policy and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope flow-control
UCS-A /eth-uplink/flow-control # create policy FlowControlPolicy23
UCS-A /eth-uplink/flow-control/policy* # set prio auto
UCS-A /eth-uplink/flow-control/policy* # set receive on
UCS-A /eth-uplink/flow-control/policy* # set send on
UCS-A /eth-uplink/flow-control/policy* # commit-buffer
UCS-A /eth-uplink/flow-control/policy #
```

What to do next

Associate the flow control policy with an uplink Ethernet port or port channel.

Deleting a Flow Control Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope flow-control**	Enters Ethernet uplink flow control mode.
Step 3	UCS-A /eth-uplink/flow-control # **delete policy** *policy-name*	Deletes the specified flow control policy.
Step 4	UCS-A /eth-uplink/flow-control # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example deletes the flow control policy named FlowControlPolicy23 and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope flow-control
UCS-A /eth-uplink/flow-control # delete policy FlowControlPolicy23
UCS-A /eth-uplink/flow-control* # commit-buffer
UCS-A /eth-uplink/flow-control #
```

Configuring Slow Drain

QoS Slow Drain Device Detection and Mitigation

All data traffic between end devices in the fabric is carried by Fibre Channel services that use link-level, per-hop-based, and buffer-to-buffer flow control. These classes of service do not support end-to-end flow control. When slow devices are attached to the fabric, the end devices do not accept the frames at the configured or negotiated rate. The slow devices lead to an Inter-Switch Link (ISL) credit shortage in the traffic that is destined for these devices, and they congest the links. The credit shortage affects the unrelated flows in the fabric that use the same ISL link even though destination devices do not experience a slow drain.

Similarly, in End-Host Mode, if a server that is directly attached to the Fabric Interconnect receives traffic slowly, it may congest the uplink port shared by other servers. If a slow server is attached to a HIF port on FEX/IOM, it may congest the fabric port and/or uplink port.

Cisco UCS Manager Release 4.0(2) introduces the QoS Slow Drain Detection and Mitigation feature on Cisco UCS 6454 Fabric Interconnects. This feature provides various enhancements that enable you to detect slow drain devices that cause congestion in the network, and also mitigate it. The enhancements are mainly on the edge ports and core ports that connect to the slow drain devices. This is done to minimize the frames stuck condition in the edge and core ports due to slow drain devices that are causing an ISL blockage. To avoid or minimize the stuck condition, you can configure smaller frame timeout for the ports. A smaller frame timeout value helps to alleviate the slow drain condition that affects the fabric by dropping the packets on the edge ports sooner than the time they actually get timed out. This function frees the buffer space in ISL, which can be used by other unrelated flows that do not experience the slow drain condition. Cisco UCS Manager Release 4.1 extends support of this feature to Cisco UCS 64108 Fabric Interconnects.

Note Another way of mitigating network congestion is to use the watchdog timer function, supported on Cisco UCS 6400 Series Fabric Interconnects starting with Cisco UCS Manager 4.2. However, the slow drain and watchdog timer functions are mutually exclusive.

In this release, slow drain detection and mitigation is supported on the following ports:

- FCoE
- Back-plane

Configuring Slow Drain Detection

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-server**	Enters Ethernet server mode.
Step 2	UCS-A /eth-server # **scope qos**	Enters Ethernet server QoS mode.
Step 3	UCS-A /eth-server/qos # **scope slow-drain**	Enters Ethernet server QoS slow drain mode.
Step 4	UCS-A /eth-server/qos/slow-drain #**set fcoe-admin-state** {**disable** | **enable**}	Sets the FCoE admin state to one of the following: • disable—Slow drain detection is disabled • enable—Slow drain detection is enabled
Step 5	UCS-A /eth-server/qos/slow-drain* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example enables slow drain detection on FCoE ports and commits the transaction:

```
UCS-A# scope eth-server
UCS-A /eth-server # scope qos
UCS-A /eth-server/qos # scope slow-drain
UCS-A /eth-server/qos/slow-drain # set fcoe-admin-state enable
UCS-A /eth-server/qos/slow-drain* # commit-buffer
UCS-A /eth-server/qos/slow-drain #
```

Configuring Slow Drain Timers

While configuring slow drain timeout timers, you can select the timeout value from the list of allowed values. You cannot configure custom timeout values.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-server**	Enters Ethernet server mode.
Step 2	UCS-A /eth-server # **scope qos**	Enters Ethernet server QoS mode.
Step 3	UCS-A /eth-server/qos # **scope slow-drain**	Enters Ethernet server QoS slow drain mode.
Step 4	UCS-A /eth-server/qos/slow-drain #**set core-port-timer** {**100** | **200** | **300** | **400** | **500** | **600** | **700** | **800** | **900** | **1000**}	Sets the core FCoE port timeout to one of the listed values.

	Command or Action	Purpose
		The default timeout value is 500 ms.
Step 5	UCS-A /eth-server/qos/slow-drain* #**set edge-port-timer** {**100** | **200** | **300** | **400** | **500** | **600** | **700** | **800** | **900** | **1000**}	Sets the edge FCoE port timeout to one of the listed values. The default timeout value is 500 ms.
Step 6	UCS-A /eth-server/qos/slow-drain* #**set backplane-port-timer** { **200** | **300** | **400** | **500** | **600** | **700** | **800** | **900** | **1000**}	Sets the backplane port timeout to one of the listed values. The default timeout value is 1000 ms.
Step 7	UCS-A /eth-server/qos/slow-drain* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example configures the slow drain timers and commits the transaction:

```
UCS-A# scope eth-server
UCS-A /eth-server # scope qos
UCS-A /eth-server/qos # scope slow-drain
UCS-A /eth-server/qos/slow-drain # set core-port-timer 500
UCS-A /eth-server/qos/slow-drain* # set edge-port-timer 500
UCS-A /eth-server/qos/slow-drain* # set backplane-port-timer 1000
UCS-A /eth-server/qos/slow-drain* # commit-buffer
UCS-A /eth-server/qos/slow-drain #
```

Displaying Slow Drain Settings

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-server**	Enters Ethernet server mode.
Step 2	UCS-A /eth-server # **scope qos**	Enters Ethernet server QoS mode.
Step 3	UCS-A /eth-server/qos # **show slow-drain**	Displays QoS slow drain settings.

Example

The following example displays the slow drain settings:

```
UCS-A# scope eth-server
UCS-A /eth-server # scope qos
UCS-A /eth-server/qos # show slow-drain

QoS Slow Drain:
```

```
    Admin State for QoS Slow Drain for Physical FCoE Ports: Enabled
    QoS Slow Drain: Timer value for Core Physical FCoE Ports: 100
    QoS Slow Drain: Timer value for Edge Physical FCoE Ports: 100
    QoS Slow Drain: Timer value for Backplane Ports: 1000
UCS-A /eth-server/qos #
```

Priority Flow Control Watchdog Interval

A PFC storm may occur in the network from a malfunctioning NIC or switch, where the Priority Flow Control (PFC) frames are propagated to all senders causing a complete stall in traffic in the network. To mitigate a PFC storm, a PFC watchdog can be used. A PFC watchdog interval can be configured to detect whether packets in a no-drop queue are being drained within a specified time period. If packets are present in buffer longer than the configured time period and after the time period expires, all outgoing packets are dropped on the interfaces that match the PFC queue that is not being drained.

Note For VIC 6332 Fabric Interconnects, Priority Flow Watchdog functionality does not operate on all 6332 Fabric Interconnect ports, due to ASIC limitations. These port limitations are as follows:

- For VIC 6332, it will not operate on Ports 1/28-32 (40G uplink-only ports).
- For VIC 6332-16UP it will not operate on the following ports: Ethernet1/1-16 (Combined Ethernet/FC ports) or 1/35-40 (40G uplink-only ports).

For VIC 6332 with Priority Flow Control Watchdog, use only supported ports as needed.

Starting with Cisco UCS Manager 4.2(1d), the watchdog timer is enabled by default. The slow drain and watchdog timer functions are mutually exclusive.

- Configuring a Priority Flow Control Watchdog Interval, on page 166
- Viewing the Watchdog Settings, on page 167

Configuring a Priority Flow Control Watchdog Interval

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-server**	Enters the Ethernet server mode.
Step 2	UCS-A /eth-server # **scope pfc**	Enters the Ethernet server PFC mode.
Step 3	UCS-A /eth-server/pfc # **set wd-admin-state {on | off}**	Globally enables or disables the PFC watchdog interval for all interfaces. The default value is **on**.
Step 4	UCS-A /eth-server/pfc # **set wd-interval 500**	Specifies the watchdog interval value. The valid range is from 100 to 1000 milliseconds. The default value is 100.

	Command or Action	Purpose
Step 5	UCS-A /eth-server/pfc # **set wd-shutdown-multiplier 1**	Specifies when to declare the PFC queue as struck. The valid range is from 1 to 10. The default value is 1.
Step 6	UCS-A /eth-server/pfc* # commit-buffer	Commits the transaction to the system configuration.

The watchdog interval, polling interval, and shutdown multiplier are configured.

Example

The following example shows how to configure the watchdog interval, polling interval, and shutdown multiplier, and then commit the transaction.

```
UCS-A# scope eth-server
UCS-A /eth-server # scope pfc
UCS-A /eth-server/pfc # set wd-admin-state on
 UCS-A /eth-server/pfc # set wd-interval 500
UCS-A /eth-server/pfc # set wd-shutdown-multiplier 1
UCS-A /eth-server/pfc* # commit-buffer
```

Viewing the Watchdog Settings

Procedure

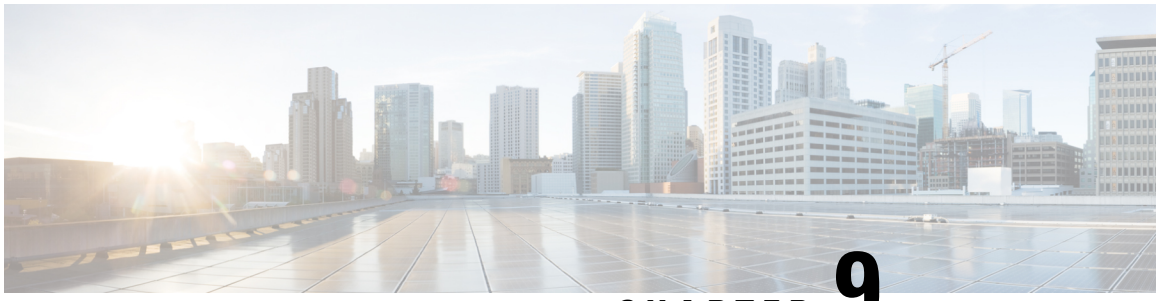
	Command or Action	Purpose
Step 1	UCS-A# **scope eth-server**	Enters the Ethernet server mode.
Step 2	UCS-A /eth-server # **show pfc details**	Displays the PFC watchdog settings.

Example

The following example displays the watchdog settings:

```
UCS-A# scope eth-server
UCS-A /eth-server # show pfc details

    Global PFC watchdog configuration details:
    PFC watchdog interval: On
    PFC watchdog poll interval: 500
    PFC watchdog shutdown multiplier: 1
    Current Task:
```



CHAPTER 9

Configuring Port Security

- Port Security Overview, on page 169
- Port Security Violations, on page 170
- Guidelines for Port Security on UCS 6454 Fabric Interconnects, on page 170
- Configuring Port Security, on page 171

Port Security Overview

The port security feature allows you to restrict input to an interface by limiting and identifying MAC addresses of the workstations that are allowed to access the port. It helps you to control the learning and storing of MAC addresses for each interface. It is used to protect against CAM overflow attacks and rogue equipment, such as hubs and switches, being plugged in. A port security enabled port is called a secure port, and the MAC addresses allowed on that port are called secure MAC addresses.When you assign secure MAC addresses to a secure port, the port does not forward packets with source addresses outside the group of defined addresses. If you limit the number of secure MAC addresses to one and assign a single secure MAC address to a secure port, the workstation attached to that port is assured the full bandwidth of the port.

After you have set the maximum number of secure MAC addresses on a port, you can include secure MAC addresses in an address table in one of these ways:

- Configure all secure MAC addresses by using the switchport port-security mac-address *mac_address* interface configuration command.
- Allow the port to dynamically configure secure MAC addresses with the MAC addresses of connected devices.
- Configure a number of addresses and allow the rest to be dynamically configured.

Note If the port shuts down, all dynamically learned addresses are removed.

- 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. If these addresses are saved in the configuration file, the interface does not need to dynamically relearn them when the switch restarts. Although sticky secure addresses can be manually configured, it is not recommended.

MAC Learning

After port security is enabled on an interface and a new MAC address is seen on the interface, a security validation is done for the new MAC address. Based on this validation, the MAC address will be added to the address table - either as a normal entry or a drop entry.

Port Security Violations

A port security violation occurs in either of these situations:

- When the maximum number of secure MAC addresses is reached on a secure port and the source MAC address of the ingress traffic is different from any of the identified secure MAC addresses, port security applies the configured violation mode.
- If traffic with a secure MAC address that is configured or learned on one secure port attempts to access another secure port in the same VLAN, port security applies the configured violation mode. This is also known as a MAC move violation.

There are three violation actions for port security. You can configure the port for one of these violation actions:

- **Shutdown**—A port security violation causes the port to shut down immediately.
- **Restrict**—A port security violation restricts data, causes the SecurityViolation counter to increment, and causes an SNMP Trap to be generated. In the Restrict action, learning is disabled on the port after 10 violations. Restrict is the default action for port security violations.
- **Protect**—A port security violation causes data from unknown MAC addresses to be dropped. The SecurityViolation counter is not incremented, and no SNMP Trap is generated.

Guidelines for Port Security on UCS 6454 Fabric Interconnects

The following guidelines apply when you configure port security for UCS 6454 Fabric Interconnect ports:

- Port security can be configured only on NIV ports. It is not supported on BIF ports.
- Only one MAC address per VLAN can be secured for an NIV port.
- For port security violations on virtual interfaces, Restrict is the default violation action.
- MAC learning is disabled on a secure port after 10 violations.
- Secure MAC addresses never age out.
- The maximum number of secure MAC addresses that can be configured are as follows:
  - On a Device—A maximum of 8000 secure MAC addresses in addition to one MAC address per port
  - On an Interface—A maximum of 1000 MAC addresses per interface
  - In a VLAN—Only one secure MAC address per port for a VLAN

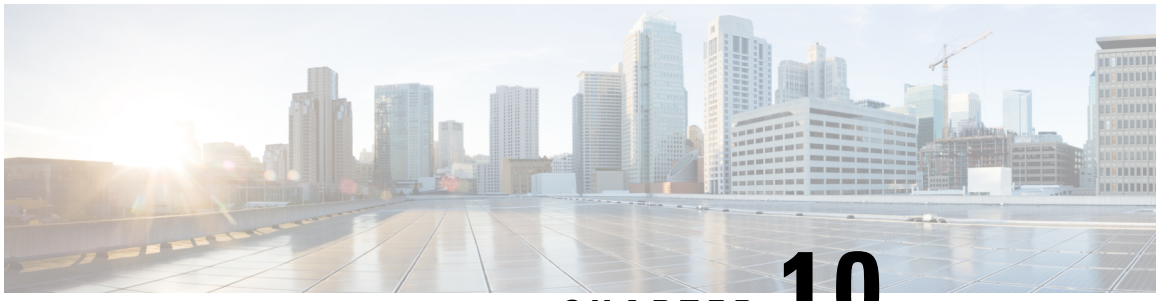
Configuring Port Security

To restrict traffic through a port by limiting and identifying MAC addresses of the workstations allowed to access the port, perform this task:

Procedure

	Command or Action	Purpose
Step 1	switch(config)# **interface** *interface_id*	Enters interface configuration mode.
Step 2	switch(config-if)# **switchport mode access**	Sets the interface mode as access; an interface in the default mode (dynamic desirable) 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 a secure port, use the no switchport port-security interface configuration command.
Step 4	switch(config-if)# **switchport port-security maximum** *value*	Sets the maximum number of secure MAC addresses for the interface. The range is 1 to 1000. To return the interface to the default number of secure MAC addresses, use the no switchport port-security maximum *value* interface configuration command.
Step 5	switch(config-if)# **switchport port-security violation** {**restrict** | **shutdown** | **protect**}	Sets the action to be taken when a security violation is detected. The action can be one of the following: • **Shutdown**—A port security violation causes the port to shut down immediately. • **Restrict**—A port security violation restricts data, causes the SecurityViolation counter to increment, and causes an SNMP Trap to be generated. In the Restrict action, learning is disabled on the port after 10 violations. Restrict is the default action for port security violations. • **Protect**—A port security violation causes data from unknown MAC addresses to be dropped. The SecurityViolation counter is not incremented, and no SNMP Trap is generated. To return the violation mode to the default condition (restrict), use the no switchport

	Command or Action	Purpose
		port-security violation {restrict | shutdown | protect} interface configuration command.
Step 6	switch(config-if)# **switchport port-security mac-address** *mac_address*	Enters a secure MAC address for the interface. You can use this command to enter the maximum number of secure MAC addresses. If you configure fewer secure MAC addresses than the maximum, the remaining MAC addresses are dynamically learned. To delete a MAC address from the address table, use the no switchport port-security mac-address *mac_address* interface configuration command.



CHAPTER 10

Upstream Disjoint Layer-2 Networks

- Upstream Disjoint Layer-2 Networks, on page 173
- Guidelines for Configuring Upstream Disjoint L2 Networks, on page 174
- Upstream Disjoint L2 Networks Pinning Considerations, on page 176
- Configuring Cisco UCS for Upstream Disjoint L2 Networks, on page 177
- Assigning Ports and Port Channels to VLANs, on page 178
- Removing Ports and Port Channels from VLANs , on page 179
- Viewing Ports and Port Channels Assigned to VLANs, on page 180

Upstream Disjoint Layer-2 Networks

Upstream disjoint layer-2 networks (disjoint L2 networks) are required if you have two or more Ethernet clouds that never connect, but must be accessed by servers or virtual machines located in the same Cisco UCS domain. For example, you could configure disjoint L2 networks if you require one of the following:

- Servers or virtual machines to access a public network and a backup network
- Servers or virtual machines for more than one customer are located in the same Cisco UCS domain, and that need to access the L2 networks for both customers in a multi-tenant system

Note By default, data traffic in Cisco UCS works on a principle of mutual inclusion. All traffic for all VLANs and upstream networks travels along all uplink ports and port channels. If you have upgraded from a release that does not support upstream disjoint layer-2 networks, you must assign the appropriate uplink interfaces to your VLANs, or traffic for those VLANs continues to flow along all uplink ports and port channels.

The configuration for disjoint L2 networks works on a principle of selective exclusion. Traffic for a VLAN that is designated as part of a disjoint network can only travel along an uplink Ethernet port or port channel that is specifically assigned to that VLAN, and is selectively excluded from all other uplink ports and port channels. However, traffic for VLANs that are not specifically assigned to an uplink Ethernet port or port channel can still travel on all uplink ports or port channels, including those that carry traffic for the disjoint L2 networks.

Note The behavior described above in a disjoint Layer-2 configuration means VLAN traffic is carried for VLANs that are not explicitly assigned to uplink Ethernet ports or port channels. To ensure complete traffic segregation and allow only explicitly assigned VLANs on disjoint Layer-2 networks, it is recommended to use VLAN groups in UCS Manager. VLAN groups ensure that traffic from unassigned VLANs does not flow over the disjoint Layer-2 network.

In Cisco UCS, the VLAN represents the upstream disjoint L2 network. When you design your network topology for disjoint L2 networks, you must assign uplink interfaces to VLANs not the reverse.

For information about the maximum number of supported upstream disjoint L2 networks, see the appropriate *Cisco UCS Configuration Limits for Cisco UCS Manager Guide*.

Guidelines for Configuring Upstream Disjoint L2 Networks

When you plan your configuration for upstream disjoint L2 networks, consider the following:

Ethernet Switching Mode Must Be End-Host Mode

Cisco UCS only supports disjoint L2 networks when the Ethernet switching mode of the fabric interconnects is configured for end-host mode. You cannot connect to disjoint L2 networks if the Ethernet switching mode of the fabric interconnects is switch mode.

Symmetrical Configuration Is Recommended for High Availability

If a Cisco UCS domain is configured for high availability with two fabric interconnects, we recommend that both fabric interconnects are configured with the same set of VLANs.

VLAN Validity Criteria Are the Same for Uplink Ethernet Ports and Port Channels

The VLAN used for the disjoint L2 networks must be configured and assigned to an uplink Ethernet port or uplink Ethernet port channel. If the port or port channel does not include the VLAN, Cisco UCS Manager considers the VLAN invalid and does the following:

- Displays a configuration warning in the **Status Details** area for the server.
- Ignores the configuration for the port or port channel and drops all traffic for that VLAN.

Note The validity criteria are the same for uplink Ethernet ports and uplink Ethernet port channels. Cisco UCS Manager does not differentiate between the two.

Overlapping VLANs Are Not Supported

Cisco UCS does not support overlapping VLANs in disjoint L2 networks. You must ensure that each VLAN only connects to one upstream disjoint L2 domain.

Each vNIC Can Only Communicate with One Disjoint L2 Network

A vNIC can only communicate with one disjoint L2 network. If a server needs to communicate with multiple disjoint L2 networks, you must configure a vNIC for each of those networks.

To communicate with more than two disjoint L2 networks, a server must have a Cisco VIC adapter that supports more than two vNICs.

Appliance Port Must Be Configured with the Same VLAN as Uplink Ethernet Port or Port Channel

For an appliance port to communicate with a disjoint L2 network, you must ensure that at least one uplink Ethernet port or port channel is in the same network and is therefore assigned to the same VLANs that are used by the appliance port. If Cisco UCS Manager cannot identify an uplink Ethernet port or port channel that includes all VLANs that carry traffic for an appliance port, the appliance port experiences a pinning failure and goes down.

For example, a Cisco UCS domain includes a global VLAN named vlan500 with an ID of 500. vlan500 is created as a global VLAN on the uplink Ethernet port. However, Cisco UCS Manager does not propagate this VLAN to appliance ports. To configure an appliance port with vlan500, you must create another VLAN named vlan500 with an ID of 500 for the appliance port. You can create this duplicate VLAN in the **Appliances** node on the **LAN** tab of the Cisco UCS Manager GUI or the **eth-storage** scope in the Cisco UCS Manager CLI. If you are prompted to check for VLAN Overlap, accept the overlap and Cisco UCS Manager creates the duplicate VLAN for the appliance port.

Default VLAN 1 Cannot Be Configured Explicitly on an Uplink Ethernet Port or Port Channel

Cisco UCS Manager implicitly assigns default VLAN 1 to all uplink ports and port channels. Even if you do not configure any other VLANs, Cisco UCS uses default VLAN 1 to handle data traffic for all uplink ports and port channels.

Note After you configure VLANs in a Cisco UCS domain, default VLAN 1 remains implicitly on all uplink ports and port channels. You cannot explicitly assign default VLAN 1 to an uplink port or port channel, nor can you remove it from an uplink port or port channel.

If you attempt to assign default VLAN 1 to a specific port or port channel, Cisco UCS Manager raises an Update Failed fault.

Therefore, if you configure a Cisco UCS domain for disjoint L2 networks, do not configure any vNICs with default VLAN 1 unless you want all data traffic for that server to be carried on all uplink Ethernet ports and port channels and sent to all upstream networks.

VLANs for Both FIs Must be Concurrently Assigned

When you assign a port to a global VLAN, the VLAN is removed from all of the ports that are not explicitly assigned to the VLAN on both fabric interconnects. The ports on both FIs must be configured at the same time. If the ports are only configured on the first FI, traffic on the second FI will be disrupted.

Upstream Disjoint L2 Networks Pinning Considerations

Communication with an upstream disjoint L2 network requires that you ensure that the pinning is properly configured. Whether you implement soft-pinning or hard-pinning, a VLAN membership mismatch causes traffic for one or more VLANs to be dropped.

Soft-Pinning

Soft-pinning is the default behavior in Cisco UCS. If you plan to implement soft-pinning, you do not need to create LAN pin groups to specify a pin target for a vNIC. Instead, Cisco UCS Manager pins the vNIC to an uplink Ethernet port or port channel according to VLAN membership criteria.

With soft-pinning, Cisco UCS Manager validates data traffic from a vNIC against the VLAN membership of all uplink Ethernet ports and port channels. If you have configured disjoint L2 networks, Cisco UCS Manager must be able to find an uplink Ethernet port or port channel that is assigned to all VLANS on the vNIC. If no uplink Ethernet port or port channel is configured with all VLANs on the vNIC, Cisco UCS Manager does the following:

- Brings the link down.
- Drops the traffic for all of the VLANs on the vNIC.
- Raises the following faults:
  - Link Down
  - VIF Down

Cisco UCS Manager does not raise a fault or warning about the VLAN configuration.

For example, a vNIC on a server is configured with VLANs 101, 102, and 103. Interface 1/3 is assigned only to VLAN 102. Interfaces 1/1 and 1/2 are not explicitly assigned to a VLAN, which makes them available for traffic on VLANs 101 and 103. As a result of this configuration, the Cisco UCS domain does not include a border port interface that can carry traffic for all three VLANS for which the vNIC is configured. As a result, Cisco UCS Manager brings down the vNIC, drops traffic for all three VLANs on the vNIC, and raises the Link Down and VIF Down faults.

hard-pinning

hard-pinning occurs when you use LAN pin groups to specify the pinning target for the traffic intended for the disjoint L2 networks. In turn, the uplink Ethernet port or port channel that is the pinning target must be configured to communicate with the appropriate disjoint L2 network.

With hard-pinning, Cisco UCS Manager validates data traffic from a vNIC against the VLAN membership of all uplink Ethernet ports and port channels, and validates the LAN pin group configuration to ensure it includes the VLAN and the uplink Ethernet port or port channel. If the validation fails at any point, Cisco UCS Manager does the following:

- Raises a Pinning VLAN Mismatch fault with a severity of Warning.
- Drops traffic for the VLAN.
- Does not bring the link down, so that traffic for other VLANs can continue to flow along it.

For example, if you want to configure hard-pinning for an upstream disjoint L2 network that uses VLAN 177, do the following:

- Create a LAN pin group with the uplink Ethernet port or port channel that carries the traffic for the disjoint L2 network.
- Configure at least one vNIC in the service profile with VLAN 177 and the LAN pin group.
- Assign VLAN 177 to an uplink Ethernet port or port channel included in the LAN pin group

If the configuration fails at any of these three points, then Cisco UCS Manager warns of a VLAN mismatch for VLAN 177 and drops the traffic for that VLAN only.

Note If changes are made to soft-pinning configurations resulting in vNIC VLANs not resolving with disjoint L2 uplink, a warning dialog box is displayed. The warning dialog box allows you to proceed with your configuration or cancel it. If you decide to proceed with the mis- configuration, you will experience a reduction is server traffic performance.

Configuring Cisco UCS for Upstream Disjoint L2 Networks

When you configure a Cisco UCS domain to connect with upstream disjoint L2 networks, you need to ensure that you complete all of the following steps.

Before you begin

Before you begin this configuration, ensure that the ports on the fabric interconnects are properly cabled to support your disjoint L2 networks configuration.

Procedure

	Command or Action	Purpose
Step 1	Configure Ethernet switching mode for both fabric interconnects in Ethernet End-Host Mode.	The Ethernet switching mode must be in End-Host Mode for Cisco UCS to be able to communicate with upstream disjoint L2 networks. For more information, see the *LAN Ports and Port Channel* chapter in this guide..
Step 2	Configure the ports and port channels that you require to carry traffic for the disjoint L2 networks.	
Step 3	(Optional) Configure the LAN pin groups required to pin the traffic for the appropriate uplink Ethernet ports or port channels.	For more information, see Configuring a LAN Pin Group, on page 147.
Step 4	Create one or more VLANs.	These can be named VLANs or private VLANs. For a cluster configuration, we recommend that you create the VLANs accessible to both fabric interconnects. For more information, see *VLAN*

	Command or Action	Purpose
		and *Upstream Disjointed Layer-2 Networks* chapters in this guide.
Step 5	Assign the desired ports or port channels to the VLANs for the disjoint L2 networks.	When this step is complete, traffic for these VLANs is be sent through the trunks for the assigned ports and/or port channels.
Step 6	Ensure that the service profiles for all servers that need to communicate with the disjoint L2 networks include the correct LAN connectivity configuration. This configuration ensures that the vNICs direct the traffic to the appropriate VLAN.	You can complete this configuration through one or more vNIC templates, or when you configure the networking options for the service profile. For more information about vNIC templates and service profiles, see the *Cisco UCS Manager Storage Management Guide*.

Assigning Ports and Port Channels to VLANs

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope vlan** *vlan-name*	Enters Ethernet uplink VLAN mode for the specified VLAN.
Step 3	UCS-A /eth-uplink/vlan # **create member-port** *fabric-interconnect slot-id port-id*	Assigns the specified VLAN to the specified uplink Ethernet port.
Step 4	UCS-A /eth-uplink/vlan # **create member-port-channel** *fabric-interconnect member-port-chan-id*	Assigns the specified VLAN to the specified uplink Ethernet port channel.
Step 5	UCS-A /eth-uplink/vlan # **commit-buffer**	Commits the transaction to the system configuration. After a port or port channel is assigned to one or more VLANs, it is removed from all other VLANs.

Example

The following example assigns uplink Ethernet ports to a named VLAN called VLAN100 on fabric interconnect A and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope vlan VLAN100
UCS-A /eth-uplink/vlan # create member-port a 2
UCS-A /eth-uplink/vlan # create member-port a 4
```

```
UCS-A /eth-uplink/vlan* # commit-buffer
UCS-A /eth-uplink/vlan #
```

Removing Ports and Port Channels from VLANs

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope vlan** *vlan-name*	Enters Ethernet uplink VLAN mode for the specified VLAN.
Step 3	UCS-A /eth-uplink/vlan # **delete member-port** *fabric-interconnect slot-id port-id*	Deletes the specified Uplink Ethernet member port assignment from the VLAN.
Step 4	UCS-A /eth-uplink/vlan # **delete member-port-channel** *fabric-interconnect member-port-chan-id*	Deletes the specified Uplink Ethernet port channel assignment from the VLAN.
Step 5	UCS-A /eth-uplink/vlan # **commit-buffer**	Commits the transaction to the system configuration. **Important** If you remove all port or port channel interfaces from a VLAN, the VLAN returns to the default behavior and data traffic on that VLAN flows on all uplink ports and port channels. Based on the configuration in the Cisco UCS domain, this default behavior can cause Cisco UCS Manager to drop traffic for that VLAN. To avoid this occurrence, Cisco recommends that you assign at least one interface to the VLAN or delete the VLAN.

Example

The following example deletes the association between uplink Ethernet port 2 on fabric interconnect A and the named VLAN called MyVLAN and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope vlan MyVLAN
UCS-A /eth-uplink/vlan # delete member-port a 2
UCS-A /eth-uplink/vlan* # commit-buffer
UCS-A /eth-uplink/vlan #
```

Viewing Ports and Port Channels Assigned to VLANs

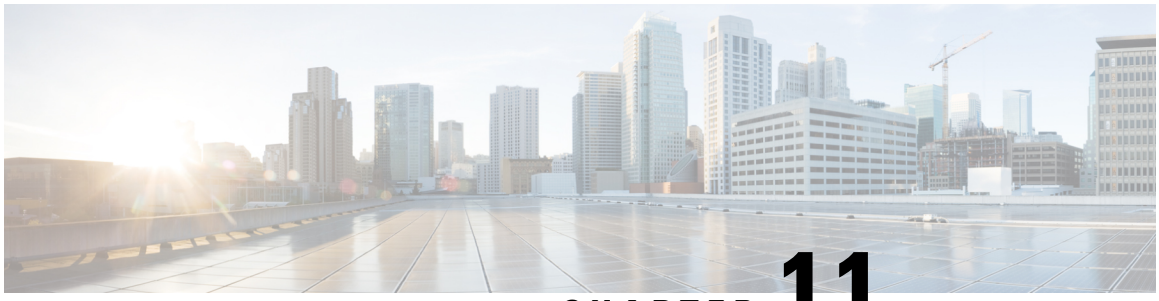
Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope vlan** *vlan-name*	Enters Ethernet uplink VLAN mode for the specified VLAN.
Step 3	UCS-A /eth-uplink/vlan # **show member-port** [**detail** | **expand**]	Shows member ports assigned to the specified VLAN.
Step 4	UCS-A /eth-uplink/vlan # **show member-port-channel** [**detail** | **expand**]	Shows member port channels assigned to the specified VLAN.
Step 5	UCS-A /eth-uplink/vlan # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example displays the full details for uplink Ethernet ports assigned to a named VLAN called MyVLAN:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope vlan MyVLAN
UCS-A /eth-uplink/vlan # show member-port detail
Member Port:
    Fabric ID: A
    Slot ID: 1
    Port ID: 2
    Mark Native Vlan: No
UCS-A /eth-uplink/vlan #
```



CHAPTER 11

Network-Related Policies

- vNIC Template, on page 181
- Ethernet Adapter Policies, on page 188
- Ethernet and Fibre Channel Adapter Policies, on page 197
- Configuring a Default vNIC Behavior Policy, on page 202
- Deleting a vNIC from a LAN Connectivity Policy, on page 203
- Creating a LAN Connectivity Policy, on page 203
- Deleting a LAN Connectivity Policy, on page 204
- About the LAN and SAN Connectivity Policies, on page 205
- Network Control Policy, on page 213
- Creating a Multicast Policy, on page 217
- Deleting a Multicast Policy, on page 218
- Entering Multicast Policy Mode, on page 219
- Enter a Multicast Policy, on page 219
- Assigning a Global VLAN Multicast Policy, on page 220
- Disassociating a Global VLAN Multicast Policy, on page 220
- Disassociating a VLAN Multicast Policy, on page 221
- Configuring SRIOV HPN Connection Policy, on page 222
- GENEVE Offload, on page 225
- Configuring Ethernet Adapter Policies, on page 227
- Configuring the Default vNIC Behavior Policy, on page 230
- Configuring a Network Control Policy, on page 231
- Deleting a Network Control Policy, on page 233
- Configuring Multicast Policies, on page 234
- LACP Policy, on page 239
- Configuring UDLD Link Policies, on page 241
- VMQ Connection Policy, on page 248

vNIC Template

The vNIC LAN connectivity policy defines how a vNIC on a server connects to the LAN.

Cisco UCS ManagerCisco UCS Central does not automatically create a VM-FEX port profile with the correct settings when you create a vNIC template. If you want to create a VM-FEX port profile, you must configure the target of the vNIC template as a VM. You must include this policy in a service profile for it to take effect.

You can select VLAN groups in addition to any individual VLAN while creating a vNIC template.

Note If your server has two Emulex or QLogic NICs (Cisco UCS CNA M71KR-E or Cisco UCS CNA M71KR-Q), you must configure vNIC policies for both adapters in your service profile to get a user-defined MAC address for both NICs. If you do not configure policies for both NICs, Windows still detects both of them in the PCI bus. Then because the second eth is not part of your service profile, Windows assigns it a hardware MAC address. If you then move the service profile to a different server, Windows sees additional NICs because one NIC did not have a user-defined MAC address.

Creating vNIC Template Pairs

Procedure

	Command or Action	Purpose
Step 1	UCS-A/ org # **create vnic-templ** *vnic-primary* .	Creates a Primary vNIC template.
Step 2	UCS-A/ # org vnic-templ **set type updating-template** .	Set the template type to updating, which drives the configurations in the Primary vNIC template for shared configurations to the peer vNIC template. See the shared configurations listed below.
Step 3	UCS-A/ # org vnic-templ [**set fabric** {**a** | **b**}] .	Specifies the fabric for the Primary vNIC template. If you specify Fabric A for the Primary vNIC template, the Secondary vNIC template must be Fabric B or vice versa.
Step 4	UCS-A/ # org vnic-templ **set descr primaryinredundancypair** .	Sets the template as the Primary vNIC template.
Step 5	UCS-A/ # org vnic-templ **set redundancy-type** *primary*.	Sets the redundancy template type as the Primary vNIC template. Following are descriptions of the **Redundancy Types**: **Primary**—Creates configurations that can be shared with the Secondary vNIC template. Any shared changes on the Primary vNIC template are automatically synchronized to the Secondary vNIC template. **Secondary** — All shared configurations are inherited from the Primary template. **No Redundancy**— Legacy vNIC template behavior. Following is a list of shared configurations:

	Command or Action	Purpose
		• **Network Control Policy** • **QoS Policy** • **Stats Threshold Policy** • **Template Type** • **Connection Policies** • **VLANS** • **MTU** Following is a list of non-shared configurations: • **Fabric ID** • **CDN Source** • **MAC Pool** • **Description** • **Pin Group Policy**
Step 6	UCS-A/ # org vnic-templ **exit** .	Exits creating the redundancy template pairing. **Note** Ensure to commit the transaction after linking the Primary vNIC template to a peer Secondary vNIC template to create the redundancy pair.
Step 7	UCS-A/ # org vnic-templ **create vNIC-templ** *vNICsecondary* .	Creates the Secondary vNIC template.
Step 8	UCS-A/ # org vnic-templ **set type updating-template** .	Sets the template type to updating, which automatically inherits the configurations from the Primary vNIC template.
Step 9	UCS-A/ org # vnic-templ [**set fabric** {**a** | **b**}] .	Specifies the fabric for the Secondary vNIC template. If you specify Fabric A for the Primary vNIC template, the Secondary vNIC template must be Fabric B or vice versa.
Step 10	UCS-A/ # org vnic-templ **set descr secondaryredundancypair**.	Sets the secondary vNIC template as a redundancy pair template.
Step 11	UCS-A/ # org vnic-templ **set redundancy-type** *secondary*.	Sets the vNIC template type as Secondary.
Step 12	UCS-A/ # org vnic-templ **set peer-template-name** *vNIC-primary*.	Sets the Primary vNIC template as the peer to the Secondary vNIC template.

	Command or Action	Purpose
Step 13	UCS-A/ # org vnic-templ **commit-buffer** .	Commits the transaction to the system configuration.

Example

The following example configures a vNIC redundancy template pair and commits the transaction:

```
UCS-A /org* # create vnic-template vnic-primary
UCS-A /org/vnic-templ* # set type updating-template
UCS-A /org/vnic-templ* # set fabric a
UCS-A /org/vnic-templ* # set descr primaryinredundancypair
UCS-A /org/vnic-templ* # set redundancy-type primary
UCS-A /org/vnic-templ* # exit
UCS-A /org* # create vnic-templ vnicsecondary
UCS-A /org/vnic-templ* # set fabric b
UCS-A /org/vnic-templ* # set descr secondaryinredundancypair
UCS-A /org/vnic-templ* # set redundancy-type secondary
UCS-A /org/vnic-templ* # set peer-template-name vnic-primary
UCS-A /org/vnic-templ* # commit-buffer
UCS-A /org/vnic-templ #
```

What to do next

After you create the vNIC redundancy template pair, you can use the redundancy template pair to create redundancy vNIC pairs for any service profile in the same organization or sub- organization.

Undo vNIC Template Pairs

You can undo the vNIC template pair by changing the Peer Redundancy Template so that there is no peer template for the Primary or the Secondary template. When you undo a vNIC template pair, the corresponding vNIC pairs also becomes undone.

Procedure

	Command or Action	Purpose
Step 1	UCS-A /org # **scope vnic-templ** *template1*.	Specifies the name of the vNIc template that you want to undo from the template pair.
Step 2	UCS-A /org/ vnic-templ # **set redundancy-type** *no redundancy*.	Removes the paring between the peer Primary or Secondary redundancy template used to perform the template pairing.
Step 3	UCS-A /org/vnic-templ* # **commit-buffer** .	Commits the transaction to the system configuration.

Example

The following example shows how to undo a template pairing:

```
UCS-A /org # scope vnic-templ template1
UCS-A /org/vnic-templ # set redundancy-type no-redundancy
UCS-A /org/vnic-templ* # commit buffer
```

Configuring a vNIC Template

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*.
Step 2	UCS-A /org # **create vnic-templ** *vnic-templ-name* [**eth-if** *vlan-name*] [**fabric** {**a** | **b**}] [**target** [**adapter** | **vm**]]	Creates a vNIC template and enters organization vNIC template mode. The target you choose determines whether or not Cisco UCS Manager automatically creates a VM-FEX port profile with the appropriate settings for the vNIC template. This can be one of the following: • **Adapter**—The vNICs apply to all adapters. No VM-FEX port profile is created if you choose this option. • **VM**—The vNICs apply to all virtual machines. A VM-FEX port profile is created if you choose this option.
Step 3	(Optional) UCS-A /org/vnic-templ # **set descr** *description*	Provides a description for the vNIC template.
Step 4	(Optional) UCS-A /org/vnic-templ # **set fabric** {**a** | **a-b** | **b** | **b-a**}	Specifies the fabric to use for the vNIC. If you did not specify the fabric when creating the vNIC template in Step 2, you have the option to specify it with this command. If you want this vNIC to be able to access the second fabric interconnect if the default one is unavailable, choose **a-b** (A is the primary) or **b-a** (B is the primary) . **Note** Do not enable fabric failover for the vNIC under the following circumstances:

	Command or Action	Purpose
		• If the Cisco UCS domain is running in Ethernet Switch Mode. vNIC fabric failover is not supported in that mode. If all Ethernet uplinks on one fabric interconnect fail, the vNICs do not fail over to the other. • If you plan to associate this vNIC to a server with an adapter that does not support fabric failover, such as the Cisco UCS 82598KR-CI 10-Gigabit Ethernet Adapter. If you do so, Cisco UCS Manager generates a configuration fault when you associate the service profile with the server.
Step 5	UCS-A /org/vnic-templ # **set mac-pool** *mac-pool-name*	The MAC address pool that vNICs created from this vNIC template should use.
Step 6	UCS-A /org/vnic-templ # **set mtu** *mtu-value*	The maximum transmission unit, or packet size, that vNICs created from this vNIC template should use. Enter an integer between 1500 and 9000. For Cisco UCS VIC 15000 Series adapters, enter an integer between 1500 and 9158. **Note** If the vNIC template has an associated QoS policy, the MTU specified here must be equal to or less than the MTU specified in the associated QoS system class. If this MTU value exceeds the MTU value in the QoS system class, packets may be dropped during data transmission. For VIC 1400 Series, VIC 14000 Series, and VIC 15000 Series adapters, you can change the MTU size of the vNIC from the host interface settings. When the Overlay network is configured, make sure that the new value is equal to or less than the MTU specified in the associated QoS system class or packets could be dropped during data transmission.
Step 7	UCS-A /org/vnic-templ # **set nw-control-policy** *policy-name*	The network control policy that vNICs created from this vNIC template should use.
Step 8	UCS-A /org/vnic-templ # **set pin-group** *group-name*	The LAN pin group that vNICs created from this vNIC template should use.

	Command or Action	Purpose
Step 9	UCS-A /org/vnic-templ # **set qos-policy** *policy-name*	The quality of service policy that vNICs created from this vNIC template should use.
Step 10	UCS-A /org/vnic-templ # **set stats-policy** *policy-name*	The statistics collection policy that vNICs created from this vNIC template should use.
Step 11	UCS-A /org/vnic-templ # **set type** {**initial-template** | **updating-template**}	Specifies the vNIC template update type. If you do not want vNIC instances created from this template to be automatically updated when the template is updated, use the **initial-template** keyword; otherwise, use the **updating-template** keyword to ensure that all vNIC instances are updated when the vNIC template is updated.
Step 12	UCS-A /org/vnic-templ # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example configures a vNIC template and commits the transaction:

```
UCS-A# scope org /
UCS-A /org* # create vnic template VnicTempFoo
UCS-A /org/vnic-templ* # set descr "This is a vNIC template example."
UCS-A /org/vnic-templ* # set fabric a
UCS-A /org/vnic-templ* # set mac-pool pool137
UCS-A /org/vnic-templ* # set mtu 8900
UCS-A /org/vnic-templ* # set nw-control-policy ncp5
UCS-A /org/vnic-templ* # set pin-group PinGroup54
UCS-A /org/vnic-templ* # set qos-policy QosPol5
UCS-A /org/vnic-templ* # set stats-policy ServStatsPolicy
UCS-A /org/vnic-templ* # set type updating-template
UCS-A /org/vnic-templ* # commit-buffer
UCS-A /org/vnic-templ #
```

Deleting a vNIC Template

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*.
Step 2	UCS-A /org # **delete vnic-templ** *vnic-templ-name*	Deletes the specified vNIC template.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example deletes the vNIC template named VnicTemp42 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # delete vnic template VnicTemp42
UCS-A /org* # commit-buffer
UCS-A /org #
```

Ethernet Adapter Policies

Configuring an Ethernet Adapter Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type **/** as the *org-name* .
Step 2	UCS-A /org # **create eth-policy** *policy-name*	Creates the specified Ethernet adapter policy and enters organization Ethernet policy mode.
Step 3	(Optional) UCS-A /org/eth-policy # **set arfs accelaratedrfs** {**enabled** | **disabled**}	Configures Accelerated RFS.
Step 4	(Optional) UCS-A /org/eth-policy # **set comp-queue count** *count*	Configures the Ethernet completion queue.
Step 5	(Optional) UCS-A /org/eth-policy # **set descr** *description*	Provides a description for the policy. **Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output.
Step 6	(Optional) UCS-A /org/eth-policy # **set failover timeout** *timeout-sec*	Configures the Ethernet failover.
Step 7	(Optional) UCS-A /org/eth-policy # **set interrupt** {**coalescing-time** *sec* | **coalescing-type** {**idle** | **min**} | **count** *count* | **mode** {**intx** | **msi** | **msi-x**}}	Configures the Ethernet interrupt.
Step 8	(Optional) UCS-A /org/eth-policy # **set nvgre adminstate** {**disabled** | **enabled**}	Configures NVGRE.

	Command or Action	Purpose
Step 9	(Optional) UCS-A /org/eth-policy # **set offload** {**large-receive** | **tcp-rx-checksum** | **tcp-segment** | **tcp-tx-checksum**} {**disabled** | **enabled**}	Configures the Ethernet offload.
Step 10	(Optional) UCS-A /org/eth-policy # **set policy-owner** {**local** | **pending**}	Specifies the owner for the Ethernet adapter policy.
Step 11	(Optional) UCS-A /org/eth-policy # **set recv-queue** {**count** *count* | **ring-size** *size-num*}	Configures the Ethernet receive queue.
Step 12	(Optional) UCS-A /org/eth-policy # **set rss receivesidescaling** {**disabled** | **enabled**}	Configures the RSS. RSS is supported with the Enhanced Network Stack (ENS) on ESXi 8.0 and 9.0.
Step 13	(Optional) UCS-A /org/eth-policy # **set trans-queue** {**count** *count* | **ring-size** *size-num*}	Configures the Ethernet transmit queue.
Step 14	(Optional) UCS-A /org/eth-policy # **set vxlan adminstate** {**disabled** | **enabled**}	Configures VXLAN.
Step 15	(Optional) UCS-A /org/eth-policy # **set geneve adminstate** {**disabled** | **enabled**}	Enables or disables Generic Network Virtualization Encapsulation (GENEVE) offload. **Note** This must be enabled to support **Receive Side Scaling (RSS)** with the **Enhanced Network Stack (ENS)** on ESXi 8.0 and 9.0.
Step 16	UCS-A /org/eth-policy # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example configures an Ethernet adapter policy, and commits the transaction:

```
UCS-A# scope org
UCS-A /org* # create eth-policy EthPolicy19
UCS-A /org/eth-policy* # set comp-queue count 16
UCS-A /org/eth-policy* # set descr "This is an Ethernet adapter policy example."
UCS-A /org/eth-policy* # set failover timeout 300
UCS-A /org/eth-policy* # set interrupt count 64
UCS-A /org/eth-policy* # set offload large-receive disabled
UCS-A /org/eth-policy* # set recv-queue count 32
UCS-A /org/eth-policy* # set rss receivesidescaling enabled
UCS-A /org/eth-policy* # set trans-queue
UCS-A /org/eth-policy* # commit-buffer
UCS-A /org/eth-policy #
```

This example shows how to configure an Ethernet adapter policy to support RSS with the Enhanced Network Stack (ENS) on ESXi 8.0 and 9.0 using a Cisco UCS VIC 15000 Series adapter.

Deleting an Ethernet Adapter Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type **/** as the *org-name* .
Step 2	UCS-A /org # **delete eth-policy** *policy-name*	Deletes the specified Ethernet adapter policy.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example deletes the Ethernet adapter policy named EthPolicy19 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # delete eth-policy EthPolicy19
UCS-A /org* # commit-buffer
UCS-A /org #
```

Receive Side Scaling (RSS)

Configuring an Ethernet Adapter Policy to Enable RSS on Windows Operating Systems

To enable Receive Side Scaling (RSS) and configure an Ethernet Adapter Policy, do the following:

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type **/** as the *org-name* .
Step 2	UCS-A /org # **create eth-policy** *policy-name*	Creates the specified Ethernet adapter policy and enters organization Ethernet policy mode.
Step 3	(Optional) UCS-A /org/eth-policy # **set arfs accelaratedrfs** {**enabled** | **disabled**}	Configures Accelerated RFS.
Step 4	(Optional) UCS-A /org/eth-policy # **set comp-queue count** *count*	Configures the Ethernet completion queue.
Step 5	(Optional) UCS-A /org/eth-policy # **set descr** *description*	Provides a description for the policy. **Note**

	Command or Action	Purpose
		If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output.
Step 6	(Optional) UCS-A /org/eth-policy # **set failover timeout** *timeout-sec*	Configures the Ethernet failover.
Step 7	(Optional) UCS-A /org/eth-policy # **set interrupt** {**coalescing-time** *sec* | **coalescing-type** {**idle** | **min**} | **count** *count* | **mode** {**intx** | **msi** | **msi-x**}}	Configures the Ethernet interrupt. In general, interrupt value should be equal to (Completion Queues + 2) rounded up to nearest power of 2. For RSS, set the coalescing-type as msi-x.
Step 8	(Optional) UCS-A /org/eth-policy # **set nvgre adminstate** {**disabled** | **enabled**}	Configures NVGRE.
Step 9	(Optional) UCS-A /org/eth-policy # **set offload** {**large-receive** | **tcp-rx-checksum** | **tcp-segment** | **tcp-tx-checksum**} {**disabled** | **enabled**}	Configures the Ethernet offload.
Step 10	(Optional) UCS-A /org/eth-policy # **set policy-owner** {**local** | **pending**}	Specifies the owner for the Ethernet adapter policy.
Step 11	(Optional) UCS-A /org/eth-policy # **set recv-queue** {**count** *count* | **ring-size** *size-num*}	Configures the Ethernet receive queue.
Step 12	(Optional) UCS-A /org/eth-policy # **set rss receivesidescaling** {**disabled** | **enabled**}	Configures the RSS. To support RSS , set**enabled**.
Step 13	(Optional) UCS-A /org/eth-policy # **set trans-queue** {**count** *count* | **ring-size** *size-num*}	Configures the Ethernet transmit queue. For RSS, set the trans-queue count as 1.
Step 14	(Optional) UCS-A /org/eth-policy # **set vxlan adminstate** {**disabled** | **enabled**}	Configures VXLAN.
Step 15	UCS-A /org/eth-policy # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to configure an Ethernet adapter policy to support RSS and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # create eth-policy NVGRE
UCS-A /org/eth-policy* # set descr "Ethernet adapter policy with stateless offloads"
UCS-A /org/eth-policy* # set nvgre adminstate enabled
UCS-A /org/eth-policy* # set comp-queue count 9
UCS-A /org/eth-policy* # set interrupt count 16
```

```
UCS-A /org/eth-policy* # set recv-queue count 8
UCS-A /org/eth-policy* # set rss receivesidescaling enabled
UCS-A /org/eth-policy* # set trans-queue 1
UCS-A /org/eth-policy* # set interrupt mode mxi-x
UCS-A /org/eth-policy* # commit-buffer
UCS-A /org/eth-policy #
```

Configuring an Ethernet Adapter Policy to Support RSS and Multiple Transmit Queues on VMware ESXi

This configuration enables Receive Side Scaling (RSS) and multiple transmit (Tx) queues for improved network performance in VMware ESXi 8.0 U3 and later, using Ethernet Adapter Policy in Cisco UCS Manager.

Prerequisites:

- **Cisco UCS Manager:** Supported from Cisco UCS Manager Release 4.3(6a) onwards.
- **VMware ESXi:** Version 8.0 U3 or later
- **nenic driver on ESXi:** Minimum required nenic driver version is 2.0.17.0-1OEM.800.1.0.20613240 (for ESXi 8.0U3).
- **Hardware:** Supported on Cisco UCS 1400, 14000, and 15000 series adapters.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* .
Step 2	UCS-A /org # **create eth-policy** *policy-name*	Creates the specified Ethernet adapter policy and enters organization Ethernet policy mode.
Step 3	Use the following parameters when creating the Ethernet adapter policy:	• Transmit Queues = n (up to 16) • Receive Queues = n (up to 16) • Completion Queues = # of Transmit Queues + # of Receive Queues • Interrupts = (# Completion Queues +2) rounded up to the nearest power of 2 • Receive Side Scaling = Enabled • VMQ Connection Policy = Disabled For more information, see Configuring an Ethernet Adapter Policy, on page 188. **Note** When VMQ is disabled, RSS engines handle the queue distribution, which may result in the

	Command or Action	Purpose
		Rx netqueue count appearing as 1 in ESXi command outputs. If VMQ is enabled, the Rx queue count will reflect the VMQ queues, and RSS engines may not be reported as active. Hence, to support RSS with multiple transmit queues, VMQ must be disabled and RSS must be enabled.
Step 4	UCS-A /org/eth-policy # **commit-buffer**	Commits the transaction to the system configuration.
Step 5	Install the appropriate nenic driver. Example, ESXi version 2.0.17.0 (as specified in prerequisites).	For more information, see https://www.cisco.com/c/en/us/support/servers-unified-computing/ucs-virtual-interface-card/products-installation-guides-list.html.
Step 6	Reboot the server.	

Example

The following example shows how to configure an Ethernet adapter policy named "rss-and-tx-esxi" with 16 Transmit and 16 Receive Queues, enabling RSS, and disabling VMQ:

```
UCS-A# scope org /
UCS-A /org* # create eth-policy rss-and-tx-esxi
UCS-A /org/eth-policy* # set descr "Ethernet adapter policy with RSS and multiple TX queues
 for ESXi"
UCS-A /org/eth-policy* # set trans-queue count 16
UCS-A /org/eth-policy* # set recv-queue count 16
UCS-A /org/eth-policy* # set comp-queue count 32 (16 (TX) + 16 (RX)
UCS-A /org/eth-policy* # set interrupt count 34 (32 (CQ) + 2 = 34)
UCS-A /org/eth-policy* # set rss receivesidescaling enabled
UCS-A /org/eth-policy* # set vmq-conn-policy disabled
UCS-A /org/eth-policy* # set interrupt mode msi-x
UCS-A /org/eth-policy* # commit-buffer
UCS-A /org/eth-policy #
```

Configuring an Ethernet Adapter Policy to Enable Stateless Offloads with NVGRE

Cisco UCS Manager supports stateless offloads with NVGRE only with Cisco UCS VIC 1340, 1380, 1385, 1387 and/or Cisco UCS VIC 1380 adapters that are installed on servers running Windows Server 2012 R2 operating systems. You cannot use NVGREs tateless offloads with Netflow, usNIC, or VM-FEX.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 2	UCS-A /org # **create eth-policy** *policy-name*	Creates the specified Ethernet adapter policy and enters organization Ethernet policy mode.
Step 3	To enable stateless offloads with NVGRE, set the following options:	• Transmit Queues = 1 • Receive Queues = n (up to 8) • Completion Queues = # of Transmit Queues + # of Receive Queues • Interrupts = # Completion Queues + 2 • Network Virtualization using Generic Routing Encapsulation = Enabled • Interrupt Mode = Msi-X **Note** If you set Interrupt Mode as Msi-X, and if pci=nomsi parameter is enabled in /boot/grub/grub.conf on RHEL system, then pci=nomsi would block the eNIC/fNIC driver to run in the Msi-X mode, impacting system performance. For more information on creating an Ethernet adapter policy, see Configuring an Ethernet Adapter Policy, on page 188.
Step 4	UCS-A /org/eth-policy # **commit-buffer**	Commits the transaction to the system configuration.
Step 5	Install an eNIC driver Version 3.0.0.8 or later.	For more information, see http://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/sw/vic_drivers/install/Windows/b_Cisco_VIC_Drivers_for_Windows_Installation_Guide.html.
Step 6	Reboot the server.	

Example

The following example shows how to configure an Ethernet adapter policy to enable stateless offloads with NVGRE and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # create eth-policy NVGRE
UCS-A /org/eth-policy* # set descr "Ethernet adapter policy with stateless offloads"
UCS-A /org/eth-policy* # set nvgre adminstate enabled
UCS-A /org/eth-policy* # set comp-queue count 16
UCS-A /org/eth-policy* # set interrupt count 64
UCS-A /org/eth-policy* # set recv-queue count 32
UCS-A /org/eth-policy* # set rss receivesidescaling enabled
UCS-A /org/eth-policy* # set trans-queue 1
UCS-A /org/eth-policy* # set interrupt mode mxi-x
UCS-A /org/eth-policy* # commit-buffer
UCS-A /org/eth-policy #
```

Configuring an Ethernet Adapter Policy to Enable Stateless Offloads with VXLAN

Cisco UCS Manager supports VXLAN TSO and checksum offloads only with Cisco UCS VIC 1340, 1380, 1385, 1387, adapters that are running on ESXi 5.5 and later releases. Stateless offloads with VXLAN cannot be used with NetFlow, usNIC, VM-FEX, Netqueue, or VMQ.

VXLAN with Receive Side-Scaling (RSS) support starts with the Cisco UCS Manager 3.1(2) release. RSS is supported with VXLAN stateless offload on VIC adapters 1340, 1380, 1385, 1387, and SIOC on Cisco UCS S3260 system for ESXi 5.5 and later releases.

Note VXLAN stateless hardware offloads are not supported with Guest OS TCP traffic over IPv6 on UCS VIC 13xx adapters. To run VXLAN encapsulated TCP traffic over IPV6, disable the VXLAN stateless offloads feature.

- To disable the VXLAN stateless offload feature in UCS Manager, disable the 'Virtual Extensible LAN' field in the Ethernet Adapter Policy.
- To disable the VXLAN stateless offload feature in the CIMC of a Cisco C-Series UCS server, uncheck 'Enable VXLAN' field in the Ethernet Interfaces pane's vNIC properties area.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type **/** as the *org-name* .
Step 2	UCS-A /org # **create eth-policy** *policy-name*	Creates the specified Ethernet adapter policy and enters organization Ethernet policy mode.
Step 3	To enable stateless offloads with VXLAN, set the following options:	• Transmit Queues = 1 • Receive Queues = n (up to 8) • Completion Queues = # of Transmit Queues + # of Receive Queues

	Command or Action	Purpose
		• Interrupts = # Completion Queues + 2 • Virtual Extensible LAN = Enabled • Interrupt Mode = Msi-X **Note** If you set Interrupt Mode as Msi-X, and if pci=nomsi parameter is enabled in /boot/grub/grub.conf on RHEL system, then pci=nomsi would block the eNIC/fNIC driver to run in the Msi-X mode, impacting system performance. • Receive Side Scaling = Enabled For more information on creating an Ethernet adapter policy, see Configuring an Ethernet Adapter Policy, on page 188.
Step 4	UCS-A /org/eth-policy # **commit-buffer**	Commits the transaction to the system configuration.
Step 5	Install an eNIC driver Version 2.3.0.10 or later.	For more information, see http://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/sw/vic_drivers/install/ESX/2-0/b_Cisco_VIC_Drivers_for_ESX_Installation_Guide.html.
Step 6	Reboot the server.	

Example

The following example shows how to configure an Ethernet adapter policy to enable stateless offloads with VXLAN and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # create eth-policy VXLAN
UCS-A /org/eth-policy* # set descr "Ethernet adapter policy with stateless offloads"
UCS-A /org/eth-policy* # set vxlan adminstate enabled
UCS-A /org/eth-policy* # set comp-queue count 16
UCS-A /org/eth-policy* # set interrupt count 32
UCS-A /org/eth-policy* # set recv-queue count 8
UCS-A /org/eth-policy* # set rss receivesidescaling enabled
UCS-A /org/eth-policy* # set trans-queue 1
UCS-A /org/eth-policy* # set interrupt mode mxi-x
UCS-A /org/eth-policy* # commit-buffer
UCS-A /org/eth-policy #
```

Ethernet and Fibre Channel Adapter Policies

These policies govern the host-side behavior of the adapter, including how the adapter handles traffic. For example, you can use these policies to change default settings for the following:

- Queues
- Interrupt handling
- Performance enhancement
- RSS hash
- Failover in a cluster configuration with two fabric interconnects

Note For Fibre Channel adapter policies, the values displayed by Cisco UCS ManagerCisco UCS Central may not match those displayed by applications such as QLogic SANsurfer. For example, the following values may result in an apparent mismatch between SANsurfer and Cisco UCS ManagerCisco UCS Central:

- Max LUNs Per Target—SANsurfer supports a maximum of 256 LUNs. Cisco UCS Manager, 6.0(2b) release version supports a maximum of 16,384 (16K) LUNs per vHBA (FC-Initiator) on Cisco UCS VIC 1400, 14000, and 15000 series adapters when used with Linux operating systems. For Windows and ESXi operating systems, the maximum number of LUNs supported is 4,096 (4K).This parameter is applicable only for FC-Initiator.
- Link Down Timeout—In SANsurfer, you configure the timeout threshold for link down in seconds. In Cisco UCS ManagerCisco UCS Central, you configure this value in milliseconds. Therefore, a value of 5500 ms in Cisco UCS ManagerCisco UCS Central displays as 5s in SANsurfer.
- Max Data Field Size—SANsurfer has allowed values of 512, 1024, and 2048. Cisco UCS ManagerCisco UCS Central allows you to set values of any size. Therefore, a value of 900 in Cisco UCS ManagerCisco UCS Central displays as 512 in SANsurfer.
- LUN Queue Depth—The LUN queue depth setting is available for Windows system FC adapter policies. Queue depth is the number of commands that the HBA can send and receive in a single transmission per LUN. Windows Storport driver sets this to a default value of 20 for physical miniports and to 250 for virtual miniports. This setting adjusts the initial queue depth for all LUNs on the adapter. Valid range for this value is 1 - 254. The default LUN queue depth is 20. This feature only works with Cisco UCS Manager version 3.1(2) and higher. This parameter is applicable only for FC-Initiator.
- IO TimeOut Retry—When the target device does not respond to an IO request within the specified timeout, the FC adapter cancels the pending command then resends the same IO after the timer expires. The FC adapter valid range for this value is 1 - 59 seconds. The default IO retry timeout is 5 seconds. This feature only works with Cisco UCS Manager version 3.1(2) and higher.

Operating System Specific Adapter Policies

By default, Cisco UCS provides a set of Ethernet adapter policies and Fibre Channel adapter policies. These policies include the recommended settings for each supported server operating system. Operating systems are sensitive to the settings in these policies. Storage vendors typically require non-default adapter settings. You can find the details of these required settings on the support list provided by those vendors.

Important We recommend that you use the values in these policies for the applicable operating system. Do not modify any of the values in the default policies unless directed to do so by Cisco Technical Support.

However, if you are creating an Ethernet adapter policy for an OS (instead of using the default adapter policy), you must use the following formulas to calculate values that work for that OS.

Depending on the UCS firmware, your driver interrupt calculations may be different. Newer UCS firmware uses a calculation that differs from previous versions. Later driver release versions on Linux operating systems now use a different formula to calculate the Interrupt Count. In this formula, the Interrupt Count is the maximum of either the Transmit Queue or the Receive Queue plus 2.

Interrupt Count in Linux Adapter Policies

Drivers on Linux operating systems use differing formulas to calculate the Interrupt Count, depending on the eNIC driver version. The UCS 3.2 release increased the number of Tx and Rx queues for the eNIC driver from 8 to 256 each.

Use one of the following strategies, according to your driver version.

For Linux drivers before the UCS 3.2 firmware release, use the following formula to calculate the Interrupt Count.

Completion Queues = Transmit Queues + Receive Queues
Interrupt Count = (Completion Queues + 2) rounded up to nearest power of 2

For example, if Transmit Queues = 1 and Receive Queues = 8 then:

Completion Queues = 1 + 8 = 9
Interrupt Count = (9 + 2) rounded up to the nearest power of 2 = 16

On drivers for UCS firmware release 3.2 and higher, the Linux eNIC drivers use the following formula to calculate the Interrupt Count.

Interrupt Count = Max(Tx, Rx) + 2

For example:

Interrupt Count wq = 32, rq = 32, cq = 64 - then Interrupt Count = Max(32, 32) + 2 = 34
Interrupt Count wq = 64, rq = 8, cq = 72 – then Interrupt Count = Max(64, 8) + 2 = 66
Interrupt Count wq = 1, rq = 16, cq = 17 - then Interrupt count = Max(1, 16) + 2 = 18

Interrupt Count in Windows Adapter Policies

For Windows OS, the recommended adapter policy in UCS Manager for VIC 1400 series and above adapters is Win-HPN and if RDMA is used, the recommended policy is Win-HPN-SMB. For VIC 1400 series and above adapters, the recommended interrupt value setting is 512 and the Windows VIC driver takes care of allocating the required number of Interrupts.

For VIC 1300 and VIC 1200 series adapters, the recommended UCS Manager adapter policy is Windows and the Interrupt would be TX + RX + 2, rounded to closest power of 2. The maximum supported Windows queues is 8 for Rx Queues and 1 for Tx Queues.

Example for VIC 1200 and VIC 1300 series adapters:

Tx = 1, Rx = 4, CQ = 5, Interrupt = 8 (1 + 4 rounded to nearest power of 2), Enable RSS

Example for VIC 1400 series , 14000 series and 15000 series adapters and above adapters:

Tx = 1, Rx = 4, CQ = 5, Interrupt = 512 , Enable RSS

NVMe over Fabrics using Fibre Channel

The NVM Express (NVMe) interface allows host software to communicate with a non-volatile memory subsystem. This interface is optimized for Enterprise non-volatile storage, which is typically attached as a register level interface to the PCI Express (PCIe) interface.

NVMe over Fabrics using Fibre Channel (FC-NVMe) defines a mapping protocol for applying the NVMe interface to Fibre Channel. This protocol defines how Fibre Channel services and specified Information Units (IUs) are used to perform the services defined by NVMe over a Fibre Channel fabric. NVMe initiators can access and transfer information to NVMe targets over Fibre Channel.

FC-NVMe combines the advantages of Fibre Channel and NVMe. You get the improved performance of NVMe along with the flexibility and the scalability of the shared storage architecture. Cisco UCS Manager Release 4.0(2) supports NVMe over Fabrics using Fibre Channel on UCS VIC 1400 Series adapters.

Starting with UCS Manager release 4.3(2b), NVMeoF using RDMA is supported on Cisco UCS VIC 14000 series adapters.

Cisco UCS Manager provides the recommended FC NVME Initiator adapter policies in the list of pre-configured adapter policies. To create a new FC-NVMe adapter policy, follow the steps in the *Creating a Fibre Channel Adapter Policy* section.

NVMe over Fabrics Using RDMA

NVMe over Fabrics (NVMeoF) is a communication protocol that allows one computer to access NVMe namespaces available on another computer. NVMeoF is similar to NVMe, but differs in the network-related steps involved in using the NVMeoF storage devices. The commands for discovering, connecting, and disconnecting a NVMeoF storage device are integrated into the **nvme** utility provided in Linux..

The NVMeoF fabric that Cisco supports is RDMA over Converged Ethernet version 2 (RoCEv2). RoCEv2 is a fabric protocol that runs over UDP. It requires a no-drop policy.

The eNIC RDMA driver works in conjunction with the eNIC driver, which must be loaded first when configuring NVMeoF.

Cisco UCS Manager provides the default Linux-NVMe-RoCE adapter policy for creating NVMe RoCEv2 interfaces. Do not use the default Linux adapter policy. For complete information on configuring RoCEv2 over NVMeoF, refer to the *Cisco UCS Manager Configuration Guide for RDMA over Converged Ethernet (RoCE) v2*.

NVMeoF using RDMA is supported on M5 B-Series or C-Series Servers with Cisco UCS VIC 1400 Series adapters.

Starting with UCS Manager release 4.3(2b), NVMeOF using RDMA is supported on Cisco UCS VIC 14000 series adapters.

Accelerated Receive Flow Steering

Accelerated Receive Flow Steering (ARFS) is hardware-assisted receive flow steering that can increase CPU data cache hit rate by steering kernel level processing of packets to the CPU where the application thread consuming the packet is running.

Using ARFS can improve usage CPU efficiency and reduce network traffic latency. Each receive queue of a CPU has an interrupt associated with it. You can configure the Interrupt Service Routine (ISR) to run on a CPU. The ISR moves the packet from the receive queue to the backlog of one of the current CPUs, which processes the packet later. If the application is not running on this CPU, the CPU must copy the packet to non-local memory, which adds to latency. ARFS can reduce this latency by moving that particular stream to the receive queue of the CPU on which the application is running.

ARFS is disabled by default and can be enabled through Cisco UCS Manager. To configure ARFS, do the following:

1. Create an adapter policy with ARFS enabled.
2. Associate the adapter policy with a service profile.
3. Enable ARFS on a host:
   a. Turn off Interrupt Request Queue (IRQ) balance.
   b. Associate IRQ with different CPUs.
   c. Enable ntuple by using ethtool.

Guidelines and Limitations for Accelerated Receive Flow Steering

- ARFS supports 64 filters per vNIC
- ARFS is supported on the following adapters:
  - Cisco UCS VIC 1300 Series
  - Cisco UCS VIC 1400 Series
  - Cisco UCS VIC 14000 Series
- ARFS is supported on the following Operating Systems:
  - Red Hat Enterprise Linux 8.4 and higher versions
  - Red Hat Enterprise Linux 9.0 and higher versions
  - SUSE Linux Enterprise Server 15 SP4 and higher versions
  - Ubuntu 20.04 and higher versions

Interrupt Coalescing

Adapters typically generate a large number of interrupts that a host CPU must service. Interrupt coalescing reduces the number of interrupts serviced by the host CPU. This is done by interrupting the host CPU only once for multiple occurrences of the same event over a configurable coalescing interval.

When interrupt coalescing is enabled for receive operations, the adapter continues to receive packets, but the host CPU does not immediately receive an interrupt for each packet. A coalescing timer starts when the first packet is received by the adapter. When the configured coalescing interval times out, the adapter generates one interrupt with the packets received during that interval. The NIC driver on the host then services the multiple packets that are received. Reduction in the number of interrupts generated reduces the time spent by

the host CPU on context switches. This means that the CPU has more time to process packets, which results in better throughput and latency.

Adaptive Interrupt Coalescing

Due to the coalescing interval, the handling of received packets adds to latency. For small packets with a low packet rate, this latency increases. To avoid this increase in latency, the driver can adapt to the pattern of traffic flowing through it and adjust the interrupt coalescing interval for a better response from the server.

Adaptive interrupt coalescing (AIC) is most effective in connection-oriented low link utilization scenarios including email server, databases server, and LDAP server. It is not suited for line-rate traffic.

Guidelines and Limitations for Adaptive Interrupt Coalescing

- Adaptive Interrupt Coalescing (AIC) does not provide any reduction in latency when the link utilization is more than 80 percent.
- Enabling AIC disables static coalescing.
- AIC is supported on the following Operating Systems:
  - Red Hat Enterprise Linux 6.4 and higher versions
  - SUSE Linux Enterprise Server 11 SP2 and higher versions
  - XenServer 6.5 and higher versions
  - Ubuntu 14.04.2 and higher versions

RDMA Over Converged Ethernet for SMB Direct

RDMA over Converged Ethernet (RoCE) allows direct memory access over an Ethernet network. RoCE is a link layer protocol, and hence, it allows communication between any two hosts in the same Ethernet broadcast domain. RoCE delivers superior performance compared to traditional network socket implementations because of lower latency, lower CPU utilization and higher utilization of network bandwidth. Windows 2012 and later versions use RDMA for accelerating and improving the performance of SMB file sharing and Live Migration.

Cisco UCS Manager Release 2.2(4) supports RoCE for Microsoft SMB Direct. It sends additional configuration information to the adapter while creating or modifying an Ethernet adapter policy.

Guidelines and Limitations for SMB Direct with RoCE

- Microsoft SMB Direct with RoCE is supported on Microsoft Windows, Release 2012 R2 for Cisco UCS Manager release 2.2(4) and later releases.
- For Microsoft SMB Direct with RoCE support on Microsoft Windows 2016 for Cisco UCS Manager release, check UCS Hardware and Software Compatibility.
- Microsoft SMB Direct with RoCE is supported only with third generation Cisco UCS VIC 1340, 1380, 1385, 1387 adapters. Second generation UCS VIC 1225 and 1227 adapters are not supported.
- RoCE configuration is supported between Cisco adapters. Interoperability between Cisco adapters and third party adapters is not supported.

- Cisco UCS Manager does not support more than 4 RoCE-enabled vNICs per adapter.
- Cisco UCS Manager does not support RoCE with NVGRE, VXLAN, NetFlow, VMQ, or usNIC.
- Maximum number of queue pairs per adapter is 8192.
- Maximum number of memory regions per adapter is 524288.
- If you do not disable RoCE before downgrading Cisco UCS Manager from Release 2.2(4), downgrade will fail.

Configuring a Default vNIC Behavior Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org /**	Enters the root organization mode.
Step 2	UCS-A/org # **scope vnic-beh-policy**	Enters default vNIC behavior policy mode.
Step 3	UCS-A/org/vnic-beh-policy # **set action** {**hw-inherit** [**template_name** *name*] | **none**}	Specifies the default vNIC behavior policy. This can be one of the following: • **hw-inherit**—If a service profile requires vNICs and none have been explicitly defined, Cisco UCS Manager creates the required vNICs based on the adapter installed in the server associated with the service profile. If you specify **hw-inherit**, you can also specify a vNIC template to create the vNICs. • **none**—Cisco UCS Manager does not create default vNICs for a service profile. All vNICs must be explicitly created.
Step 4	UCS-A/org/vnic-beh-policy # **commit-buffer**	Commits the transaction to the system configuration.

Example

This example shows how to set the default vNIC behavior policy to **hw-inherit**:

```
UCS-A # scope org /
UCS-A/org # scope vnic-beh-policy
UCS-A/org/vnic-beh-policy # set action hw-inherit
UCS-A/org/vnic-beh-policy* # commit-buffer
UCS-A/org/vnic-beh-policy #
```

Deleting a vNIC from a LAN Connectivity Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*.
Step 2	UCS-A /org # **scope lan-connectivity-policy** *policy-name*	Enters LAN connectivity policy mode for the specified LAN connectivity policy.
Step 3	UCS-A /org/lan-connectivity-policy # **delete vnic** *vnic-name*	Deletes the specified vNIC from the LAN connectivity policy.
Step 4	UCS-A /org/lan-connectivity-policy # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to delete a vNIC named vnic3 from a LAN connectivity policy named LanConnect42 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope lan-connectivity-policy LanConnect42
UCS-A /org/lan-connectivity-policy # delete vnic vnic3
UCS-A /org/lan-connectivity-policy* # commit-buffer
UCS-A /org/lan-connectivity-policy #
```

Creating a LAN Connectivity Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*.
Step 2	UCS-A /org # **create lan-connectivity-policy** *policy-name*	Creates the specified LAN connectivity policy, and enters organization LAN connectivity policy mode. This name can be between 1 and 16 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and you

	Command or Action	Purpose
		cannot change this name after the object is saved.
Step 3	(Optional) UCS-A /org/lan-connectivity-policy # **set descr** *policy-name*	Adds a description to the policy. We recommend that you include information about where and how the policy should be used. Enter up to 256 characters. You can use any characters or spaces except ` (accent mark), \ (backslash), ^ (carat), " (double quote), = (equal sign), > (greater than), < (less than), or ' (single quote).
Step 4	UCS-A /org/lan-connectivity-policy # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to create a LAN connectivity policy named LanConnect42 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # create lan-connectivity-policy LanConnect42
UCS-A /org/lan-connectivity-policy* # set descr "LAN connectivity policy"
UCS-A /org/lan-connectivity-policy* # commit-buffer
UCS-A /org/lan-connectivity-policy #
```

What to do next

Add one or more vNICs and/or iSCSI vNICs to this LAN connectivity policy.

Deleting a LAN Connectivity Policy

If you delete a LAN connectivity policy that is included in a service profile, it also deletes all vNICs and iSCSI vNICs from that service profile, and disrupt LAN data traffic for the server associated with the service profile.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*.
Step 2	UCS-A /org # **delete lan-connectivity-policy** *policy-name*	Deletes the specified LAN connectivity policy.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to delete the LAN connectivity policy named LanConnectiSCSI42 from the root organization and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # delete lan-connectivity-policy LanConnectiSCSI42
UCS-A /org* # commit-buffer
UCS-A /org #
```

About the LAN and SAN Connectivity Policies

Connectivity policies determine the connections and the network communication resources between the server and the LAN or SAN on the network. These policies use pools to assign MAC addresses, WWNs, and WWPNs to servers and to identify the vNICs and vHBAs that the servers use to communicate with the network.

Note We do not recommend that you use static IDs in connectivity policies, because these policies are included in service profiles and service profile templates and can be used to configure multiple servers.

Privileges Required for LAN and SAN Connectivity Policies

Connectivity policies enable users without network or storage privileges to create and modify service profiles and service profile templates with network and storage connections. However, users must have the appropriate network and storage privileges to create connectivity policies.

Privileges Required to Create Connectivity Policies

Connectivity policies require the same privileges as other network and storage configurations. For example, you must have at least one of the following privileges to create connectivity policies:

- admin—Can create LAN and SAN connectivity policies
- ls-server—Can create LAN and SAN connectivity policies
- ls-network—Can create LAN connectivity policies
- ls-storage—Can create SAN connectivity policies

Privileges Required to Add Connectivity Policies to Service Profiles

After the connectivity policies have been created, a user with ls-compute privileges can include them in a service profile or service profile template. However, a user with only ls-compute privileges cannot create connectivity policies.

Interactions between Service Profiles and Connectivity Policies

You can configure the LAN and SAN connectivity for a service profile through either of the following methods:

- LAN and SAN connectivity policies that are referenced in the service profile
- Local vNICs and vHBAs that are created in the service profile
- Local vNICs and a SAN connectivity policy
- Local vHBAs and a LAN connectivity policy

Cisco UCS maintains mutual exclusivity between connectivity policies and local vNIC and vHBA configuration in the service profile. You cannot have a combination of connectivity policies and locally created vNICs or vHBAs. When you include a LAN connectivity policy in a service profile, all existing vNIC configuration is erased, and when you include a SAN connectivity policy, all existing vHBA configuration in that service profile is erased.

Creating a LAN Connectivity Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*.
Step 2	UCS-A /org # **create lan-connectivity-policy** *policy-name*	Creates the specified LAN connectivity policy, and enters organization LAN connectivity policy mode. This name can be between 1 and 16 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and you cannot change this name after the object is saved.
Step 3	(Optional) UCS-A /org/lan-connectivity-policy # **set descr** *policy-name*	Adds a description to the policy. We recommend that you include information about where and how the policy should be used. Enter up to 256 characters. You can use any characters or spaces except ` (accent mark), \ (backslash), ^ (carat), " (double quote), = (equal sign), > (greater than), < (less than), or ' (single quote).
Step 4	UCS-A /org/lan-connectivity-policy # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to create a LAN connectivity policy named LanConnect42 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org* # create lan-connectivity-policy LanConnect42
UCS-A /org/lan-connectivity-policy* # set descr "LAN connectivity policy"
UCS-A /org/lan-connectivity-policy* # commit-buffer
UCS-A /org/lan-connectivity-policy #
```

What to do next

Add one or more vNICs and/or iSCSI vNICs to this LAN connectivity policy.

Creating a vNIC for a LAN Connectivity Policy

If you are continuing from Creating a LAN Connectivity Policy, on page 203, begin this procedure at Step 3.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*.
Step 2	UCS-A /org # **scope lan-connectivity-policy** *policy-name*	Enters LAN connectivity policy mode for the specified LAN connectivity policy.
Step 3	UCS-A /org/lan-connectivity-policy # **create vnic** *vnic-name* [**eth-if** *eth-if-name*] [**fabric** {**a** | **b**}]	Creates a vNIC for the specified LAN connectivity policy. This name can be between 1 and 16 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and you cannot change this name after the object is saved.
Step 4	UCS-A /org/lan-connectivity-policy/vnic # **set fabric** {**a** | **a-b** | **b** | **b-a**}	Specifies the fabric to use for the vNIC. If you did not specify the fabric when you created the vNIC in Step 3, you have the option to specify it with this command. If you want this vNIC to be able to access the second fabric interconnect if the default one is unavailable, choose **a-b** (A is the primary) or **b-a** (B is the primary) . **Note** Do not enable fabric failover for the vNIC under the following circumstances: • If the Cisco UCS domain is running in Ethernet Switch Mode. vNIC fabric failover is not supported in that mode. If all Ethernet uplinks on one fabric

	Command or Action	Purpose
		interconnect fail, the vNICs do not fail over to the other. • If you plan to associate this vNIC to a server with an adapter that does not support fabric failover, such as the Cisco UCS 82598KR-CI 10-Gigabit Ethernet Adapter. If you do so, Cisco UCS Manager generates a configuration fault when you associate the service profile with the server.
Step 5	UCS-A /org/lan-connectivity-policy/vnic # **set adapter-policy** *policy-name*	Specifies the adapter policy to use for the vNIC.
Step 6	UCS-A /org/lan-connectivity-policy/vnic # **set identity** {**dynamic-mac** {*mac-addr* | **derived**} | **mac-pool** *mac-pool-name*}	Specifies the identity (MAC address) for the vNIC. You can set the identity using one of the following options: • Create a unique MAC address in the form *nn***:** *nn***:***nn***:***nn* **:***nn***:***nn*. • Derive the MAC address from one burned into the hardware at manufacture. • Assign a MAC address from a MAC pool.
Step 7	UCS-A /org/lan-connectivity-policy/vnic # **set mtu** *size-num*	Specifies the maximum transmission unit, or packet size, that this vNIC accepts. Enter an integer between 1500 and 9000. For Cisco UCS VIC 15000 Series adapters, enter an integer between 1500 and 9158. **Note** If the vNIC has an associated QoS policy, the MTU specified here must be equal to or less than the MTU specified in the associated QoS system class. If this MTU value exceeds the MTU value in the QoS system class, packets might get dropped during data transmission.
Step 8	UCS-A /org/lan-connectivity-policy/vnic # **set nw-control-policy** *policy-name*	Specifies the network control policy that the vNIC should use.
Step 9	UCS-A /org/lan-connectivity-policy/vnic # **set order** {*order-num* | **unspecified**}	Specifies the relative order for the vNIC.
Step 10	UCS-A /org/lan-connectivity-policy/vnic # **set pin-group** *group-name*	Specifies the LAN pin group that the vNIC should use.

	Command or Action	Purpose
Step 11	UCS-A /org/lan-connectivity-policy/vnic # **set qos-policy** *policy-name*	Specifies the quality of service policy that the vNIC should use.
Step 12	UCS-A /org/lan-connectivity-policy/vnic # **set stats-policy** *policy-name*	Specifies the statistics collection policy that the vNIC should use.
Step 13	UCS-A /org/lan-connectivity-policy/vnic # **set template-name** *policy-name*	Specifies the dynamic vNIC connectivity policy to use for the vNIC.
Step 14	UCS-A /org/lan-connectivity-policy/vnic # **set vcon** {**1** | **2** | **3** | **4** | **any**}	Assigns the vNIC to the specified vCon. Use the **any** keyword to have Cisco UCS Manager automatically assign the vNIC.
Step 15	UCS-A /org/lan-connectivity-policy/vnic # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to configure a vNIC for a LAN connectivity policy named LanConnect42 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope lan-connectivity-policy LanConnect42
UCS-A /org/lan-connectivity-policy* # create vnic vnic3 fabric a
UCS-A /org/lan-connectivity-policy/vnic* # set fabric a-b
UCS-A /org/lan-connectivity-policy/vnic* # set adapter-policy AdaptPol2
UCS-A /org/lan-connectivity-policy/vnic* # set identity mac-pool MacPool3
UCS-A /org/lan-connectivity-policy/vnic* # set mtu 8900
UCS-A /org/lan-connectivity-policy/vnic* # set nw-control-policy ncp5
UCS-A /org/lan-connectivity-policy/vnic* # set order 0
UCS-A /org/lan-connectivity-policy/vnic* # set pin-group EthPinGroup12
UCS-A /org/lan-connectivity-policy/vnic* # set qos-policy QosPol5
UCS-A /org/lan-connectivity-policy/vnic* # set stats-policy StatsPol2
UCS-A /org/lan-connectivity-policy/vnic* # set template-name VnicConnPol3
UCS-A /org/lan-connectivity-policy/vnic* # set vcon any
UCS-A /org/lan-connectivity-policy/vnic* # commit-buffer
UCS-A /org/lan-connectivity-policy/vnic #
```

What to do next

If desired, add another vNIC or an iSCSI vNIC to the LAN connectivity policy. If not, include the policy in a service profile or service profile template.

Deleting a vNIC from a LAN Connectivity Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*.
Step 2	UCS-A /org # **scope lan-connectivity-policy** *policy-name*	Enters LAN connectivity policy mode for the specified LAN connectivity policy.
Step 3	UCS-A /org/lan-connectivity-policy # **delete vnic** *vnic-name*	Deletes the specified vNIC from the LAN connectivity policy.
Step 4	UCS-A /org/lan-connectivity-policy # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to delete a vNIC named vnic3 from a LAN connectivity policy named LanConnect42 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope lan-connectivity-policy LanConnect42
UCS-A /org/lan-connectivity-policy # delete vnic vnic3
UCS-A /org/lan-connectivity-policy* # commit-buffer
UCS-A /org/lan-connectivity-policy #
```

Creating an iSCSI vNIC for a LAN Connectivity Policy

If you are continuing from Creating a LAN Connectivity Policy, on page 203, begin this procedure at Step 3.

Before you begin

The LAN connectivity policy must include an Ethernet vNIC that can be used as the overlay vNIC for the iSCSI device.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter / as the *org-name*.
Step 2	UCS-A /org # **scope lan-connectivity-policy** *policy-name*	Enters LAN connectivity policy mode for the specified LAN connectivity policy.

	Command or Action	Purpose
Step 3	UCS-A /org/lan-connectivity-policy # **create vnic-iscsi** *iscsi-vnic-name* .	Creates an iSCSI vNIC for the specified LAN connectivity policy. This name can be between 1 and 16 alphanumeric characters. You cannot use spaces or any special characters other than - (hyphen), _ (underscore), : (colon), and . (period), and you cannot change this name after the object is saved.
Step 4	(Optional) UCS-A /org/lan-connectivity-policy/vnic-iscsi # **set iscsi-adaptor-policy** *iscsi-adaptor-name*	Specifies the iSCSI adapter policy that you have created for this iSCSI vNIC.
Step 5	(Optional) UCS-A /org/lan-connectivity-policy/vnic-iscsi # **set auth-name** *authentication-profile-name*	Sets the authentication profile to be used by the iSCSI vNIC. The authentication profile must already exist for it to be set. For more information, see *Creating an Authentication Profile*.
Step 6	UCS-A /org/lan-connectivity-policy/vnic-iscsi # **set identity** { **dynamic-mac** {*dynamic-mac-address* | **derived** } | **mac-pool** *mac-pool-name* }	Specifies the MAC address for the iSCSI vNIC. **Note** The MAC address is set only for the Cisco UCS NIC M51KR-B Adapters.
Step 7	UCS-A /org/lan-connectivity-policy/vnic-iscsi # **set iscsi-identity** {**initiator-name** *initiator-name* | **initiator-pool-name** *iqn-pool-name*}	Specifies the name of the iSCSI initiator or the name of an IQN pool from which the iSCSI initiator name will be provided. The iSCSI initiator name can be up to 223 characters.
Step 8	UCS-A /org/lan-connectivity-policy/vnic-iscsi # **set overlay-vnic-name** *overlay-vnic-name*	Specifies the Ethernet vNIC that is used by the iSCSI device as the overlay vNIC. For more information, see *Configuring a vNIC for a Service Profile*.
Step 9	UCS-A /org/lan-connectivity-policy/vnic-iscsi # **create eth-if**	Creates an Ethernet interface for a VLAN assigned to the iSCSI vNIC.
Step 10	UCS-A /org/ex/vnic-iscsi/eth-if # **set vlanname** *vlan-name*	Specifies the VLAN name. The default VLAN is default. For the Cisco UCS M81KR Virtual Interface Card and the Cisco UCS VIC-1240 Virtual Interface Card, the VLAN that you specify must be the same as the native VLAN on the overlay vNIC. For the Cisco UCS M51KR-B Broadcom BCM57711 Adapter, the VLAN that you specify can be any VLAN assigned to the overlay vNIC.
Step 11	UCS-A /org/lan-connectivity-policy/vnic-iscsi # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to configure an iSCSI vNIC for a LAN connectivity policy named LanConnect42 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope lan-connectivity-policy LanConnect42
UCS-A /org/lan-connectivity-policy # create vnic-iscsi iSCSI1
UCS-A /org/lan-connectivity-policy/vnic-iscsi* # set iscsi-adaptor-policy iscsiboot
UCS-A /org/lan-connectivity-policy/vnic-iscsi* # set auth-name initauth
UCS-A /org/lan-connectivity-policy/vnic-iscsi* # set identity dynamic-mac derived
UCS-A /org/lan-connectivity-policy/vnic-iscsi* # set iscsi-identity initiator-name iSCSI1
UCS-A /org/lan-connectivity-policy/vnic-iscsi* # set overlay-vnic-name eth1
UCS-A /org/lan-connectivity-policy/vnic-iscsi* # create eth-if
UCS-A /org/lan-connectivity-policy/vnic-iscsi/eth-if* # set vlanname default
UCS-A /org/lan-connectivity-policy/vnic-iscsi/eth-if* # commit buffer
UCS-A /org/lan-connectivity-policy/vnic-iscsi/eth-if
```

What to do next

If desired, add another iSCI vNIC or a vNIC to the LAN connectivity policy. If not, include the policy in a service profile or service profile template.

Deleting an iSCSI vNIC from a LAN Connectivity Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*.
Step 2	UCS-A /org # **scope lan-connectivity-policy** *policy-name*	Enters LAN connectivity policy mode for the specified LAN connectivity policy.
Step 3	UCS-A /org/lan-connectivity-policy # **delete vnic-iscsi** *iscsi-vnic-name*	Deletes the specified iSCSI vNIC from the LAN connectivity policy.
Step 4	UCS-A /org/lan-connectivity-policy # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to delete an iSCSI vNIC named iscsivnic3 from a LAN connectivity policy named LanConnect42 and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # scope lan-connectivity-policy LanConnect42
UCS-A /org/lan-connectivity-policy # delete vnic-iscsi iscsivnic3
UCS-A /org/lan-connectivity-policy* # commit-buffer
UCS-A /org/lan-connectivity-policy #
```

Network Control Policy

This policy configures the network control settings for the Cisco UCS domain, including the following:

- Whether the Cisco Discovery Protocol (CDP) is enabled or disabled
- How the virtual interface (VIF) behaves if no uplink port is available in end-host mode
- The action that Cisco UCS Manager takes on the remote Ethernet interface, vEthernet interface , or vFibre Channel interface when the associated border port fails
- Whether the server can use different MAC addresses when sending packets to the fabric interconnect
- Whether MAC registration occurs on a per-VNIC basis or for all VLANs

Action on Uplink Fail

By default, the **Action on Uplink Fail** property in the network control policy is configured with a value of link-down. For adapters such as the Cisco UCS M81KR Virtual Interface Card, this default behavior directs Cisco UCS Manager to bring the vEthernet or vFibre Channel interface down if the associated border port fails. For Cisco UCS systems using a non-VM-FEX capable converged network adapter that supports both Ethernet and FCoE traffic, such as Cisco UCS CNA M72KR-Q and the Cisco UCS CNA M72KR-E, this default behavior directs Cisco UCS Manager to bring the remote Ethernet interface down if the associated border port fails. In this scenario, any vFibre Channel interfaces that are bound to the remote Ethernet interface are brought down as well.

Note If your implementation includes non-VM-FEX capable converged network adapters mentioned in this section and the adapter is expected to handle both Ethernet and FCoE traffic, we recommend that you configure the **Action on Uplink Fail** property with a value of warning. This configuration might result in an Ethernet teaming driver being unable to detect a link failure when the border port goes down.

MAC Registration Mode

MAC addresses are installed only on the native VLAN by default, which maximizes the VLAN port count in most implementations.

Note If a trunking driver is being run on the host and the interface is in promiscuous mode, we recommend that you set the MAC Registration Mode to All VLANs.

Configuring a Network Control Policy

MAC address-based port security for Emulex converged Network Adapters (N20-AE0102) is not supported. When MAC address-based port security is enabled, the fabric interconnect restricts traffic to packets that contain the MAC address that it first learns. This is either the source MAC address used in the FCoE Initialization Protocol packet, or the MAC address in an ethernet packet, whichever is sent first by the adaptor. This configuration can result in either FCoE or Ethernet packets being dropped.

Note Cisco UCS Manager Release 4.0(2) introduces support for **MAC Security** on Cisco UCS 6454 Fabric Interconnects.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*.
Step 2	UCS-A /org # **create nw-ctrl-policy** *policy-name*	Creates the specified network control policy, and enters organization network control policy mode.
Step 3	UCS-A /org/nw-ctrl-policy # {**disable** | **enable**} **cdp**	Disables or enables Cisco Discovery Protocol (CDP).
Step 4	UCS-A /org/nw-ctrl-policy # {**disable** | **enable**} **lldp transmit**	Disables or enables the transmission of LLDP packets on an interface.
Step 5	UCS-A /org/nw-ctrl-policy # {**disable** | **enable**} **lldp receive**	Disables or enables the reception of LLDP packets on an interface.
Step 6	UCS-A /org/nw-ctrl-policy # **set uplink-fail-action** {**link-down** | **warning**}	Specifies the action to be taken when no uplink port is available in end-host mode. Use the **link-down** keyword to change the operational state of a vNIC to down when uplink connectivity is lost on the fabric interconnect, and facilitate fabric failover for vNICs. Use the **warning** keyword to maintain server-to-server connectivity even when no uplink port is available, and disable fabric failover when uplink connectivity is lost on the fabric interconnect. The default uplink failure action is link-down.
Step 7	UCS-A /org/nw-ctrl-policy # **set mac-registration-mode**{**all-host-vlans** | **only-native-vlan**	Whether adapter-registered MAC addresses are added only to the native VLAN associated with the interface or added to all VLANs associated with the interface. This can be one of the following: • **Only Native Vlan**—MAC addresses are only added to the native VLAN. This option is the default, and it maximizes the port+VLAN count. • **All Host Vlans**—MAC addresses are added to all VLANs with which they are

	Command or Action	Purpose
		associated. Select this option if your VLANs are configured to use trunking but are *not* running in Promiscuous mode.
Step 8	UCS-A /org/nw-ctrl-policy # **create mac-security**	Enters organization network control policy MAC security mode.
Step 9	UCS-A /org/nw-ctrl-policy/mac-security # **set forged-transmit** {**allow** | **deny**}	Determine whether the server can use different MAC addresses when sending packets to the fabric interconnect. Entering **allow** means all server packets are accepted by the fabric interconnect, regardless of the MAC address associated with the packets. Entering **deny** means after the first packet has been sent to the fabric interconnect, all other packets must use the same MAC address or they will be silently rejected by the fabric interconnect. If you plan to install VMware ESX on the associated server, you must configure the MAC Security to **allow** for the network control policy applied to the default vNIC. If you do not configure MAC Security for **allow**, the ESX installation may fail because the MAC security permits only one MAC address while the installation process requires more than one MAC address.
Step 10	UCS-A /org/nw-ctrl-policy/mac-security # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to create a network control policy named ncp5, enable CDP, enable LLDP transmit and LLDP receive, set the uplink fail action to link-down, deny forged MAC addresses (enable MAC security), and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # create nw-ctrl-policy ncp5
UCS-A /org/nw-ctrl-policy* # enable cdp
UCS-A /org/nw-ctrl-policy* # enable lldp transmit
UCS-A /org/nw-ctrl-policy* # enable lldp receive
UCS-A /org/nw-ctrl-policy* # set uplink-fail-action link-down
UCS-A /org/nw-ctrl-policy* # create mac-security
UCS-A /org/nw-ctrl-policy/mac-security* # set forged-transmit deny
UCS-A /org/nw-ctrl-policy/mac-security* # commit-buffer
UCS-A /org/nw-ctrl-policy/mac-security #
```

The following example shows how to create a network control policy named ncp5, enable CDP, set the uplink fail action to link-down, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # create nw-ctrl-policy ncp5
```

```
UCS-A /org/nw-ctrl-policy* # enable cdp
UCS-A /org/nw-ctrl-policy* # set uplink-fail-action link-down
UCS-A /org/nw-ctrl-policy* # commit-buffer
UCS-A /org/nw-ctrl-policy #
```

Configuring Link Layer Discovery Protocol for Fabric Interconnect vEthernet Interfaces

Cisco UCS Manager allows you to enable and disable LLDP on a vEthernet interface. You can also retrieve information about these LAN uplink neighbors. This information is useful while learning the topology of the LAN connected to the UCS system and while diagnosing any network connectivity issues from the Fabric Interconnect (FI). The FI of a UCS system is connected to LAN uplink switches for LAN connectivity and to SAN uplink switches for storage connectivity. When using Cisco UCS with Cisco Application Centric Infrastructure (ACI), LAN uplinks of the FI are connected to ACI leaf nodes. Enabling LLDP on a vEthernet interface will help the Application Policy Infrastructure Controller (APIC) to identify the servers connected to the FI by using vCenter.

To permit the discovery of devices in a network, support for Link Layer Discovery Protocol (LLDP), a vendor-neutral device discovery protocol that is defined in the IEEE 802.1ab standard, is introduced. LLDP is a one-way protocol that allows network devices to advertise information about themselves to other devices on the network. LLDP transmits information about the capabilities and current status of a device and its interfaces. LLDP devices use the protocol to solicit information only from other LLDP devices.

You can enable or disable LLDP on a vEthernet interface based on the Network Control Policy (NCP) that is applied on the vNIC in the service profile.

Displaying Network Control Policy Details

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*.
Step 2	UCS-A /org # **scope nw-ctrl-policy** {**default** | *policy-name*}	Enters organization network control policy mode for the specified network control policy.
Step 3	UCS-A /org/nw-ctrl-policy # **show detail**	Displays details about the specified network control policy.

Example

The following example shows how to display the details of a network control policy named ncp5:

```
UCS-A# scope org /
UCS-A /org # scope nw-ctrl-policy ncp5
UCS-A /org/nw-ctrl-policy* # show detail

Network Control Policy:
```

```
    Name: ncp5
    CDP: Enabled
    LLDP Transmit: Enabled
    LLDP Receive: Enabled
    Uplink fail action: Link Down
    Adapter MAC Address Registration: Only Native Vlan
    Policy Owner: Local
    Description:

UCS-A /org/nw-ctrl-policy #
```

Deleting a Network Control Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org /**	Enters the root organization mode.
Step 2	UCS-A /org # **delete nwctrl-policy** *policy-name*	Deletes the specified network control policy.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example deletes the network control policy named ncp5 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # delete nwctrl-policy ncp5
UCS-A /org* # commit-buffer
UCS-A /org #
```

Creating a Multicast Policy

A multicast policy can be created only in the root organization and not in a sub-organization.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters organization mode for the specified organization.
Step 2	UCS-A /org # **create mcast-policy** *policy-name*	Creates a multicast policy with the specified policy name, and enters organization multicast policy mode.

	Command or Action	Purpose
Step 3	UCS-A /org/mcast-policy* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to create a multicast policy named policy1:

```
UCS-A# scope org /
UCS-A /org # create mcast-policy policy1
UCS-A /org/mcast-policy* # commit-buffer
UCS-A /org/mcast-policy #
```

Deleting a Multicast Policy

Note If you assigned a non-default (user-defined) multicast policy to a VLAN and then delete that multicast policy, the associated VLAN inherits the multicast policy settings from the default multicast policy until the deleted policy is re-created.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters organization mode for the specified organization.
Step 2	UCS-A /org # **delete mcast-policy** *policy-name*	Deletes a multicast policy with the specified policy name.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to delete a multicast policy named policy1:

```
UCS-A # scope org /
UCS-A /org # delete mcast-policy policy1
UCS-A /org* # commit-buffer
UCS-A /org #
```

Entering Multicast Policy Mode

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters organization mode for the specified organization.
Step 2	UCS-A /org # **scope mcast-policy** *policy-name*	Enters organization multicast policy mode.

Example

The following example shows how to create a multicast policy named policy1:

```
UCS-A# scope org /
UCS-A /org # scope mcast-policy policy1
UCS-A /org/mcast-policy #
```

Enter a Multicast Policy

You can enter an existing multicast policy using the **enter mcast-policy** *policy-name* command.

Before you begin

Create a multicast policy.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters organization mode for the specified organization.
Step 2	UCS-A /org # **enter mcast-policy** *policy-name*	Creates a new multicast policy with the specified policy name, and enters organization multicast policy mode.

Example

The following example shows how to create a multicast policy named policy1 and enter mcast-policy mode:

```
UCS-A# scope org /
UCS-A /org # enter mcast-policy policy1
UCS-A /org/mcast-policy #
```

Assigning a Global VLAN Multicast Policy

You can assign a multicast policy to a global VLAN in the Ethernet uplink fabric mode.

Before you begin

Create a VLAN.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope vlan default**	Enters Ethernet uplink VLAN mode.
Step 3	UCS-A /eth-uplink/vlan # **set mcastpolicy** *policy-name*	Assigns a multicast policy to a global VLAN.
Step 4	UCS-A /eth-uplink/vlan # **commit-buffer**	Commits the transaction to the system configuration.

Disassociating a Global VLAN Multicast Policy

You can disassociate a multicast policy from global VLANs in the Ethernet uplink fabric mode.

Note If you assigned a non-default (user-defined) multicast policy to a VLAN and then delete that multicast policy, the associated VLAN inherits the multicast policy settings from the default multicast policy until the deleted policy is re-created.

Before you begin

Create a Global VLAN and associate a multicast policy.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope vlan default**	Enters Ethernet uplink VLAN mode.
Step 3	UCS-A /eth-uplink/vlan # **set mcastpolicy ""**	Disassociates any multicast policy from the global VLAN. If you configure **set mcastpolicy ""** in a VLAN, the VLAN will inherit multicast settings from the default multicast policy.

	Command or Action	Purpose
Step 4	UCS-A /eth-uplink/vlan # **commit-buffer**	Commits the transaction to the system configuration.

Disassociating a VLAN Multicast Policy

You can disassociate a VLAN from any multicast policy in the Ethernet uplink fabric mode by entering an empty string ("") as the policy name.

Before you begin

Create a VLAN and associate a multicast policy to the VLAN.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	Required: UCS-A /eth-uplink # **scope fabric** {**a** | **b**}	Enters Ethernet uplink fabric mode for the specified fabric interconnect.
Step 3	UCS-A /eth-uplink/fabric # **scope vlan** *vlan-name*	Enters Ethernet uplink fabric VLAN mode.
Step 4	UCS-A /eth-uplink/fabric/vlan # **set mcastpolicy ""**	Disassociates any multicast policy for the VLAN. If you configure **set mcastpolicy ""** in a VLAN, the VLAN will inherit multicast settings from the default multicast policy.
Step 5	UCS-A /eth-uplink/fabric/vlan # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example disassociates any multicast policy from a VLAN named vlan1 and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric a
UCS-A /eth-uplink/fabric # scope vlan vlan1
UCS-A /eth-uplink/fabric/vlan # set mcastpolicy policy1
UCS-A /eth-uplink/fabric/vlan* # commit-buffer
UCS-A /eth-uplink/fabric/vlan #
```

Configuring SRIOV HPN Connection Policy

Single Root I/O Virtualization HPN Connection Policy

Beginning with the release 4.3(2b), Cisco UCS Manager provides Single Root I/O Virtualization High Performance Networking (SRIOV-HPN) Connection Policy support on Cisco UCS M5, M6 and M7 servers with UCS VIC 1400, 14000, and 15000 series adapters.

Single Root I/O Virtualization allows multiple VMs running a variety of guest operating systems to share a single PCIe network adapter within a host server. SR-IOV allows a VM to move data directly to and from the network adapter, bypassing the hypervisor for increased network throughput and lower server CPU burden.

You cannot enable the following when SRIOV-HPN is enabled:

- QinQ on the same vNIC
- VXLAN on the same vNIC
- Geneve offload on the same vNIC
- ENS on the same vNIC
- RoCE V2 on the same vNIC
- Netqueue on the same vNIC

Note

- CDN is supported on the host interface only and is not supported on the VM interface.
- Microsoft stand-alone NIC Teaming on SRIOV-HPN enabled vNICs is not supported.
- DPDK is supported on Linux VM.
- RSS is supported on the same vNIC.

Configuring SRIOV HPN Connection Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name*.
Step 2	UCS-A /org # **create sriov-hpn-conn-policy** *policy-name*	Specifies the name for the SRIOV HPN connection policy.
Step 3	UCS-A /org/sriov-hpn-conn-policy* # **set sriov-hpn-count***sriov hpn count*	Specifies the SRIOV HPN count for the SRIOV HPN connection policy.

	Command or Action	Purpose
Step 4	UCS-A /org/sriov-hpn-conn-policy* # **set transmit-queue-count***transmit queue count*	Specifies the transmit count for the SRIOV HPN connection policy.
Step 5	UCS-A /org/sriov-hpn-conn-policy* # **set receive-queue-count***receive queue count*	Specifies the receive queue count for the SRIOV HPN connection policy.
Step 6	UCS-A /org/sriov-hpn-conn-policy* # **set completion-queue-count***completion-queue count*	Specifies the completion queue count for the SRIOV HPN connection policy.
Step 7	UCS-A /org/sriov-hpn-conn-policy* # **set interrupt-queue-count***interrupt queue count*	Specifies the interrupt count for the SRIOV HPN connection policy.
Step 8	UCS-A /org/service-profile/vnic/sriov-hpn-conn-policy-ref* # commit-buffer	Commits the transaction to the system.

Example

The following example creates a SRIOV HPN connection policy *sriov-test*:

```
UCS-A# scope org
UCS-A /org # create sriov-hpn-conn-policy sriov-test
UCS-A /org/sriov-hpn-conn-policy* # set sriov-hpn-count 8
UCS-A /org/sriov-hpn-conn-policy* # set transmit-queue-count 1
UCS-A /org/sriov-hpn-conn-policy* # set receive-queue-count 4
UCS-A /org/sriov-hpn-conn-policy* # set completion-queue-count 5
UCS-A /org/sriov-hpn-conn-policy* # set interrupt-queue-count 8
UCS-A /org/sriov-hpn-conn-policy* # commit-buffer
UCS-A /org/sriov-hpn-conn-policy #
```

Assigning SRIOV-HPN Connection Policy to a vNIC

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* .
Step 2	UCS-A /org # **scope service-profile** *service-profile-name*	Enters the service profile mode.
Step 3	UCS-A /org/service-profile/ # **scope vnic***vnic-name*	Selects the appropriate vNIC.
Step 4	UCS-A /org/service-profile/vnic # **set adapter-policy SRIOV-HPN**	
Step 5	UCS-A /org/service-profile/vnic # **enter sriov-hpn-conn-policy-ref***SRIOV-HPN-Connection-Policy-name*	Assigns the selected SRIOV HPN connection policy to the vNIC.

	Command or Action	Purpose
Step 6	UCS-A /org/service-profile/vnic/sriov-hpn-conn-policy-ref* # **commit-buffer**	Commits the transaction to the system.

Example

The following assigns the SRIOV HPN Connection Policy sriov-test to a vNIC:

```
UCS-A# scope org
UCS-A /org # scope service-profile server 1/1
UCS-A /org/service-profile # scope vnic eth1
UCS-A /org/service-profile/vnic # set adapter-policy SRIOV-HPN
UCS-A /org/service-profile/vnic* # enter sriov-hpn-conn-policy-ref sriov-test
UCS-A /org/service-profile/vnic/sriov-hpn-conn-policy-ref* # commit-buffer
UCS-A /org/service-profile/vnic/sriov-hpn-conn-policy-ref # exit
```

Deleting SRIOV HPN Connection Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type **/** as the *org-name* .
Step 2	UCS-A /org # **scope service-profile** *service-profile-name*	Enters the service profile mode.
Step 3	UCS-A /org/service-profile/ # **scope vnic***vnic-name*	Selects the appropriate vNIC.
Step 4	UCS-A /org/service-profile/vnic # **show sriov-hpn-conn-policy-ref**	Displays the SRIOV HPN connection policy assigned to the vNIC.
Step 5	UCS-A /org/service-profile/vnic # **delete sriov-hpn-conn-policy-ref** *policy-name*	Deletes the specified SRIOV HPN connection policy.

Example

The following example deletes the SRIOV HPN connection policy named *sriov-test* and commits the transaction:

```
UCS-A# scope org
UCS-A /org # scope service-profile server 1/1
UCS-A /org/service-profile # scope vnic eth1
UCS-A /org/service-profile/vnic # show sriov-hpn-conn-policy-ref

SRIOV HPN Connection Policy Reference:
SRIOV HPN Connection Policy Name
--------------------------------
```

```
sriov-test
UCS-A /org/service-profile/vnic # delete sriov-hpn-conn-policy-ref sriov-test
UCS-A /org/service-profile/vnic # exit
```

GENEVE Offload

Cisco UCS Manager now supports Generic Network Virtualization Encapsulation (GENEVE) Offload on the ESXi platform, which allows essentially any information to be encoded in a packet and passed between tunnel endpoints. GENEVE provides the overlay capability to create isolated, multi-tenant broadcast domains across data center fabrics on UCS VIC 1400 Series, VIC 14000 Series, and VIC 15000 Series adapters. Using the GENEVE protocol allows you to create logical networks that span physical network boundaries.

GENEVE offload is present in all Ethernet adapter policies and is disabled by default.

When GENEVE offload is enabled, Cisco recommends the following values:

- Transmit Queues: 1
- TX Ring Size: 4096
- Receive Queues: 8
- RX Ring Size: 4096
- Completion Queues: 16
- Interrupts: 32

Feature Compatibility

- **UCS Manager Release 6.0(2b) and Later**: Supports Receive Side Scaling (RSS) with ENS.
- **UCS Manager Release 4.3(5a) and Later**: Supports IPv6 overlay offload functionality (compatible with ESXi 8.0 U1 and later)
- **Unsupported Features**: Azure Stack QoS and RoCEv2 are not supported when GENEVE offload is enabled

Limitations with GENEVE offload:

- GENEVE offload is not supported on Cisco UCS VIC 1300 and 1200 Series adapters.
- GENEVE offload-enabled interfaces do not support usNIC, Netflow, advanced filters, NetQueue, or aRFS.
- External outer IPV6 is not supported with GENEVE offload.
- Cisco recommends that you remove the GENEVE offload configuration before downgrading to any non-supported release versions.

Enhanced Network Stack support with Geneve Offload

Enhanced Network Stack (ENS) also known as Enhanced Data Path Mode on ESXi, utilizes Data Plane Development Kit (DPDK)-like techniques for higher throughput and employs polling to achieve high packet rates throug polling.

For more information, see Enhanced Network Stack (ENS) Support, on page 226.

Capabilities of nenic-ens Driver:

- Supports TSO for both IPv4 and IPv6.
- Supports Geneve offload with IPv4 or IPv6 outer headers.
- Supports Tx/Rx checksum offload for IPv4 or IPv6 inner/outer packets.
- Supports NetQueue (recognized as Geneve filter in ESXi).
- RSS is supported on ESXi 8.0 and 9.0. RSS allows for better traffic distribution across CPU cores than NetQueue.

Recommended Ethernet Adapter Policy Settings:

For ENS for releases prior to 4.3(4a):

- Single Tx/Rx queue
- Tx/Rx ring size of 4096
- Enable GENEVE offload

For releases 4.3(4a) or later:

- Enable **Multi Tx/Rx queues** (NetQueue allows the use of multiple transmit (Tx) and receive (Rx) queues for network traffic and helps in distributing the load more evenly across multiple CPU cores.)
- Enable **Advanced Filter**.
- Enable **GENEVE offload**.
- Enable **VMQ Connection Policy** with 2 or more VMQ queues, applied to vNICs for VM data path.

Note For detailed information on implementing GENEVE offload configuration, see the VMware NSX-T.

Enhanced Network Stack (ENS) Support

The Enhanced Network Stack (ENS), also referred to as **Enhanced Data Path Mode** in VMware environments, is designed for high-performance networking. It utilizes **Data Plane Development Kit (DPDK)**-like techniques, such as polling and optimized memory management, to achieve significantly higher throughput and packet rates compared to the standard network stack.

Driver Selection

The ESXi operating system selects the appropriate driver based on the configuration of the NSX-T virtual switch (vSwitch):

- **Standard Mode:** The system uses the `nenic` driver.
- **ENS Mode:** The system uses the `nenic-ens` driver for uplink ports.

ENS Datapath Modes

Beginning with ESXi 8.0 and 9.0, the `nenic-ens` driver supports two distinct datapath modes:

- **Enhanced Datapath - Standard:** Operates in **Interrupt Mode**, which balances performance with CPU efficiency.
- **Enhanced Datapath - Performance:** Operates in **Polling Mode**, utilizing continuous CPU cycles to process packets at maximum speed.

Receive Side Scaling (RSS) Support with ENS

Starting with Cisco UCS Manager Release 4.3(5a), the `nenic-ens` driver supports **Receive Side Scaling (RSS)** on ESXi 8.0 and ESXi 9.0.

RSS allows the system to distribute received network traffic across multiple CPU cores more efficiently than NetQueue. This ensures that Cisco VIC adapters maintain high performance and competitive traffic distribution capabilities in virtualized environments.

Prerequisites for RSS with ENS:

- **GENEVE Offload** Must be enabled in the Ethernet adapter policy. For more information, see
- **Hardware** Requires Cisco UCS VIC 15000 Series adapters.
- **Firmware** Requires Cisco VIC firmware version 5.2(2.29) or later.

Feature Capabilities and Compatibility

The `nenic-ens` driver provides the following capabilities:

- **Checksum Offload** Supports Tx/Rx checksum offload for both inner and outer IPv4/v6 packets.
- **TSO** Supports TCP Segmentation Offload (TSO) for IPv4 and IPv6.
- **NetQueue** Supports NetQueue (also known as Geneve filter) for traffic distribution.
- **RSS** Supports Receive Side Scaling for improved load balancing on supported ESXi versions.

Configuring Ethernet Adapter Policies

Configuring an Ethernet Adapter Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* .
Step 2	UCS-A /org # **create eth-policy** *policy-name*	Creates the specified Ethernet adapter policy and enters organization Ethernet policy mode.

	Command or Action	Purpose
Step 3	(Optional) UCS-A /org/eth-policy # **set arfs accelaratedrfs** {**enabled** | **disabled**}	Configures Accelerated RFS.
Step 4	(Optional) UCS-A /org/eth-policy # **set comp-queue count** *count*	Configures the Ethernet completion queue.
Step 5	(Optional) UCS-A /org/eth-policy # **set descr** *description*	Provides a description for the policy. **Note** If your description includes spaces, special characters, or punctuation, you must begin and end your description with quotation marks. The quotation marks will not appear in the description field of any **show** command output.
Step 6	(Optional) UCS-A /org/eth-policy # **set failover timeout** *timeout-sec*	Configures the Ethernet failover.
Step 7	(Optional) UCS-A /org/eth-policy # **set interrupt** {**coalescing-time** *sec* | **coalescing-type** {**idle** | **min**} | **count** *count* | **mode** {**intx** | **msi** | **msi-x**}}	Configures the Ethernet interrupt.
Step 8	(Optional) UCS-A /org/eth-policy # **set nvgre adminstate** {**disabled** | **enabled**}	Configures NVGRE.
Step 9	(Optional) UCS-A /org/eth-policy # **set offload** {**large-receive** | **tcp-rx-checksum** | **tcp-segment** | **tcp-tx-checksum**} {**disabled** | **enabled**}	Configures the Ethernet offload.
Step 10	(Optional) UCS-A /org/eth-policy # **set policy-owner** {**local** | **pending**}	Specifies the owner for the Ethernet adapter policy.
Step 11	(Optional) UCS-A /org/eth-policy # **set recv-queue** {**count** *count* | **ring-size** *size-num*}	Configures the Ethernet receive queue.
Step 12	(Optional) UCS-A /org/eth-policy # **set rss receivesidescaling** {**disabled** | **enabled**}	Configures the RSS. RSS is supported with the Enhanced Network Stack (ENS) on ESXi 8.0 and 9.0.
Step 13	(Optional) UCS-A /org/eth-policy # **set trans-queue** {**count** *count* | **ring-size** *size-num*}	Configures the Ethernet transmit queue.
Step 14	(Optional) UCS-A /org/eth-policy # **set vxlan adminstate** {**disabled** | **enabled**}	Configures VXLAN.
Step 15	(Optional) UCS-A /org/eth-policy # **set geneve adminstate** {**disabled** | **enabled**}	Enables or disables Generic Network Virtualization Encapsulation (GENEVE) offload. **Note**

	Command or Action	Purpose
		This must be enabled to support **Receive Side Scaling (RSS)** with the **Enhanced Network Stack (ENS)** on ESXi 8.0 and 9.0.
Step 16	UCS-A /org/eth-policy # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example configures an Ethernet adapter policy, and commits the transaction:

```
UCS-A# scope org
UCS-A /org* # create eth-policy EthPolicy19
UCS-A /org/eth-policy* # set comp-queue count 16
UCS-A /org/eth-policy* # set descr "This is an Ethernet adapter policy example."
UCS-A /org/eth-policy* # set failover timeout 300
UCS-A /org/eth-policy* # set interrupt count 64
UCS-A /org/eth-policy* # set offload large-receive disabled
UCS-A /org/eth-policy* # set recv-queue count 32
UCS-A /org/eth-policy* # set rss receivesidescaling enabled
UCS-A /org/eth-policy* # set trans-queue
UCS-A /org/eth-policy* # commit-buffer
UCS-A /org/eth-policy #
```

This example shows how to configure an Ethernet adapter policy to support RSS with the Enhanced Network Stack (ENS) on ESXi 8.0 and 9.0 using a Cisco UCS VIC 15000 Series adapter.

Deleting an Ethernet Adapter Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type / as the *org-name* .
Step 2	UCS-A /org # **delete eth-policy** *policy-name*	Deletes the specified Ethernet adapter policy.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example deletes the Ethernet adapter policy named EthPolicy19 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # delete eth-policy EthPolicy19
```

```
UCS-A /org* # commit-buffer
UCS-A /org #
```

Configuring the Default vNIC Behavior Policy

Default vNIC Behavior Policy

Default vNIC behavior policy allows you to configure how vNICs are created for a service profile. You can choose to create vNICS manually, or you can create them automatically.

You can configure the default vNIC behavior policy to define how vNICs are created. This can be one of the following:

- **None**—Cisco UCS ManagerCisco UCS Central does not create default vNICs for a service profile. All vNICs must be explicitly created.
- **HW Inherit**—If a service profile requires vNICs and none have been explicitly defined, Cisco UCS ManagerCisco UCS Central creates the required vNICs based on the adapter installed in the server associated with the service profile.

Note If you do not specify a default behavior policy for vNICs, **HW Inherit** is used by default.

Configuring a Default vNIC Behavior Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org /**	Enters the root organization mode.
Step 2	UCS-A/org # **scope vnic-beh-policy**	Enters default vNIC behavior policy mode.
Step 3	UCS-A/org/vnic-beh-policy # **set action** {**hw-inherit** [**template_name** *name*] | **none**}	Specifies the default vNIC behavior policy. This can be one of the following: • **hw-inherit**—If a service profile requires vNICs and none have been explicitly defined, Cisco UCS Manager creates the required vNICs based on the adapter installed in the server associated with the service profile. If you specify **hw-inherit**, you can also specify a vNIC template to create the vNICs.

	Command or Action	Purpose
		• **none**—Cisco UCS Manager does not create default vNICs for a service profile. All vNICs must be explicitly created.
Step 4	UCS-A/org/vnic-beh-policy # **commit-buffer**	Commits the transaction to the system configuration.

Example

This example shows how to set the default vNIC behavior policy to **hw-inherit**:

```
UCS-A # scope org /
UCS-A/org # scope vnic-beh-policy
UCS-A/org/vnic-beh-policy # set action hw-inherit
UCS-A/org/vnic-beh-policy* # commit-buffer
UCS-A/org/vnic-beh-policy #
```

Configuring a Network Control Policy

MAC address-based port security for Emulex converged Network Adapters (N20-AE0102) is not supported. When MAC address-based port security is enabled, the fabric interconnect restricts traffic to packets that contain the MAC address that it first learns. This is either the source MAC address used in the FCoE Initialization Protocol packet, or the MAC address in an ethernet packet, whichever is sent first by the adaptor. This configuration can result in either FCoE or Ethernet packets being dropped.

Note Cisco UCS Manager Release 4.0(2) introduces support for **MAC Security** on Cisco UCS 6454 Fabric Interconnects.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters the organization mode for the specified organization. To enter the root organization mode, enter **/** as the *org-name*.
Step 2	UCS-A /org # **create nw-ctrl-policy** *policy-name*	Creates the specified network control policy, and enters organization network control policy mode.
Step 3	UCS-A /org/nw-ctrl-policy # {**disable** | **enable**} **cdp**	Disables or enables Cisco Discovery Protocol (CDP).
Step 4	UCS-A /org/nw-ctrl-policy # {**disable** | **enable**} **lldp transmit**	Disables or enables the transmission of LLDP packets on an interface.

	Command or Action	Purpose
Step 5	UCS-A /org/nw-ctrl-policy # {**disable** | **enable**} **lldp receive**	Disables or enables the reception of LLDP packets on an interface.
Step 6	UCS-A /org/nw-ctrl-policy # **set uplink-fail-action** {**link-down** | **warning**}	Specifies the action to be taken when no uplink port is available in end-host mode. Use the **link-down** keyword to change the operational state of a vNIC to down when uplink connectivity is lost on the fabric interconnect, and facilitate fabric failover for vNICs. Use the **warning** keyword to maintain server-to-server connectivity even when no uplink port is available, and disable fabric failover when uplink connectivity is lost on the fabric interconnect. The default uplink failure action is link-down.
Step 7	UCS-A /org/nw-ctrl-policy # **set mac-registration-mode**{**all-host-vlans** | **only-native-vlan**	Whether adapter-registered MAC addresses are added only to the native VLAN associated with the interface or added to all VLANs associated with the interface. This can be one of the following: • **Only Native Vlan**—MAC addresses are only added to the native VLAN. This option is the default, and it maximizes the port+VLAN count. • **All Host Vlans**—MAC addresses are added to all VLANs with which they are associated. Select this option if your VLANs are configured to use trunking but are *not* running in Promiscuous mode.
Step 8	UCS-A /org/nw-ctrl-policy # **create mac-security**	Enters organization network control policy MAC security mode.
Step 9	UCS-A /org/nw-ctrl-policy/mac-security # **set forged-transmit** {**allow** | **deny**}	Determine whether the server can use different MAC addresses when sending packets to the fabric interconnect. Entering **allow** means all server packets are accepted by the fabric interconnect, regardless of the MAC address associated with the packets. Entering **deny** means after the first packet has been sent to the fabric interconnect, all other packets must use the same MAC address or they will be silently rejected by the fabric interconnect. If you plan to install VMware ESX on the associated server, you must configure the MAC Security to **allow** for the network control policy applied to the default vNIC. If you do

	Command or Action	Purpose
		not configure MAC Security for **allow**, the ESX installation may fail because the MAC security permits only one MAC address while the installation process requires more than one MAC address.
Step 10	UCS-A /org/nw-ctrl-policy/mac-security # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to create a network control policy named ncp5, enable CDP, enable LLDP transmit and LLDP receive, set the uplink fail action to link-down, deny forged MAC addresses (enable MAC security), and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # create nw-ctrl-policy ncp5
UCS-A /org/nw-ctrl-policy* # enable cdp
UCS-A /org/nw-ctrl-policy* # enable lldp transmit
UCS-A /org/nw-ctrl-policy* # enable lldp receive
UCS-A /org/nw-ctrl-policy* # set uplink-fail-action link-down
UCS-A /org/nw-ctrl-policy* # create mac-security
UCS-A /org/nw-ctrl-policy/mac-security* # set forged-transmit deny
UCS-A /org/nw-ctrl-policy/mac-security* # commit-buffer
UCS-A /org/nw-ctrl-policy/mac-security #
```

The following example shows how to create a network control policy named ncp5, enable CDP, set the uplink fail action to link-down, and commit the transaction:

```
UCS-A# scope org /
UCS-A /org # create nw-ctrl-policy ncp5
UCS-A /org/nw-ctrl-policy* # enable cdp
UCS-A /org/nw-ctrl-policy* # set uplink-fail-action link-down
UCS-A /org/nw-ctrl-policy* # commit-buffer
UCS-A /org/nw-ctrl-policy #
```

Deleting a Network Control Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org /**	Enters the root organization mode.
Step 2	UCS-A /org # **delete nwctrl-policy** *policy-name*	Deletes the specified network control policy.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example deletes the network control policy named ncp5 and commits the transaction:

```
UCS-A# scope org /
UCS-A /org # delete nwctrl-policy ncp5
UCS-A /org* # commit-buffer
UCS-A /org #
```

Configuring Multicast Policies

Multicast Policy

This policy is used to configure Internet Group Management Protocol (IGMP) snooping, IGMP querier, and IGMP source IP proxy. IGMP Snooping dynamically determines hosts in a VLAN that should be included in particular multicast transmissions. You can create, modify, and delete a multicast policy that can be associated to one or more VLANs. When a multicast policy is modified, all VLANs associated with that multicast policy are re-processed to apply the changes.

By default, IGMP snooping is enabled and IGMP querier is disabled. When IGMP snooping is enabled, the fabric interconnects send the IGMP queries only to the hosts. They do not send IGMP queries to the upstream network. To send IGMP queries to the upstream, do one of the following:

- Configure IGMP querier on the upstream fabric interconnect with IGMP snooping enabled
- Disable IGMP snooping on the upstream fabric interconnect
- Change the fabric interconnects to switch mode

By default, IGMP Source IP Proxy state is enabled. When IGMP Source IP Proxy is enabled, the fabric interconnect acts as a proxy for its hosts and manages the membership of hosts and routing devices in multicast groups. IP hosts use IGMP to report their multicast group memberships to any immediately neighboring multicast routing devices. When IGMP source IP proxy is disabled, the fabric interconnect will forward the IGMP messages from the hosts towards the upstream router or switch without any change.

The following limitations and guidelines apply to multicast policies:

- Only the default multicast policy is allowed for a global VLAN.
- We highly recommend you use the same IGMP snooping state on the fabric interconnects and the associated LAN switches. For example, if IGMP snooping is disabled on the fabric interconnects, it should be disabled on any associated LAN switches as well.
- The option to enable or disable IGMP source IP proxy is supported on the following fabric interconnects:
  - Cisco UCS 6600 Series Fabric Interconnect
  - Cisco UCS Fabric Interconnects 9108 100G
  - Cisco UCS 6500 Series Fabric Interconnects
  - Cisco UCS 6400 Series Fabric Interconnects

Creating a Multicast Policy

A multicast policy can be created only in the root organization and not in a sub-organization.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters organization mode for the specified organization.
Step 2	UCS-A /org # **create mcast-policy** *policy-name*	Creates a multicast policy with the specified policy name, and enters organization multicast policy mode.
Step 3	UCS-A /org/mcast-policy* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to create a multicast policy named policy1:

```
UCS-A# scope org /
UCS-A /org # create mcast-policy policy1
UCS-A /org/mcast-policy* # commit-buffer
UCS-A /org/mcast-policy #
```

Configuring IGMP Parameters

You can configure the following parameters for a multicast policy:

1. Enable or disable IGMP snooping. The default state is enabled.
2. Set the IGMP snooping querier state and IPv4 address. The default state is disabled.
3. Set the IGMP source IP proxy state. The default state is enabled.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters organization mode for the specified organization.
Step 2	UCS-A /org # **create mcast-policy** *policy-name*	Creates a new multicast policy with the specified policy name, and enters organization multicast policy mode.
Step 3	UCS-A /org/mcast-policy* # **set querier**{**enabled** | **disabled**}	Enables or disables IGMP snooping querier. By default, IGMP snooping querier is disabled for a multicast policy.

	Command or Action	Purpose
Step 4	UCS-A /org/mcast-policy* # **set querierip** *IGMP snooping querier IPv4 address*	Specifies the IPv4 address for the IGMP snooping querier.
Step 5	UCS-A /org/mcast-policy* # **set snooping**{**enabled** | **disabled**}	Enables or disables IGMP snooping. By default, IGMP snooping is enabled for a multicast policy.
Step 6	UCS-A /org/mcast-policy* # **set source-ip-proxy**{**enabled** | **disabled**}	Enables or disables IGMP source IP proxy. By default, IGMP source IP proxy state is enabled for a multicast policy. **Note** IGMP source IP proxy is supported on Cisco UCS 6400 Series, Series Fabric Interconnects.
Step 7	UCS-A /org/mcast-policy* # **commit-buffer**	Commits the transaction to the system configuration. **Note** Follow these guidelines if you choose to set IGMP Snooping querier IP addresses for a multicast policy: **a.** In the Ethernet Switch-Mode configuration, you must set the querier IP addresses for each FI in the domain. **b.** In the Ethernet End-Host mode, you can set the querier IP address just for FI A, and optionally for FI B as well. If an IP address is not set explicitly for FI-B, it uses the same address set for FI A.

Example

The following example shows how to create and enter a multicast policy named policy1:

```
UCS-A# scope org /
UCS-A /org # create mcast-policy policy1
UCS-A /org/mcast-policy* # set querier enabled
UCS-A /org/mcast-policy* # set querierip 1.2.3.4
UCS-A /org/mcast-policy* # set snooping enabled
UCS-A /org/mcast-policy* # set source-ip-proxy enabled
UCS-A /org/mcast-policy* # commit-buffer
UCS-A /org/mcast-policy #
```

Modifying Multicast Policy Parameters

You can modify an existing multicast policy to change the state of IGMP snooping, IGMP snooping querier, or IGMP source IP proxy state. When a multicast policy is modified, all VLANs associated with that multicast policy are re-processed to apply the changes.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters organization mode for the specified organization.
Step 2	UCS-A /org # **scope mcast-policy** *policy-name*	Enters organization multicast policy mode.
Step 3	UCS-A /org/mcast-policy* # **set querier**{**enabled** | **disabled**}	Enables or disables IGMP snooping querier. By default, IGMP snooping querier is disabled for a multicast policy.
Step 4	UCS-A /org/mcast-policy* # **set querierip** *IGMP snooping querier IPv4 address*	Specifies the IPv4 address for the IGMP snooping querier.
Step 5	UCS-A /org/mcast-policy* # **set snooping**{**enabled** | **disabled**}	Enables or disables IGMP snooping. By default, IGMP snooping is enabled for a multicast policy.
Step 6	UCS-A /org/mcast-policy* # **set-source-ip-proxy**{**enabled** | **disabled**}	Enables or disables IGMP source IP proxy. By default, IGMP source IP proxy state is enabled for a multicast policy.
Step 7	UCS-A /org/mcast-policy* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to create a multicast policy named policy1:

```
UCS-A# scope org /
UCS-A /org # scope mcast-policy policy1
UCS-A /org/mcast-policy* # set querier enabled
UCS-A /org/mcast-policy* # set querierip 1.2.3.4
UCS-A /org/mcast-policy* # set snooping enabled
UCS-A /org/mcast-policy* # set source-ip-proxy enabled
UCS-A /org/mcast-policy* # commit-buffer
UCS-A /org/mcast-policy #
```

Assigning a VLAN Multicast Policy

You can set a multicast policy for a VLAN in the Ethernet uplink fabric mode. You cannot set a multicast policy for an isolated VLAN.

Before you begin

Create a VLAN.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	Required: UCS-A /eth-uplink # **scope fabric** {**a** | **b**}	Enters Ethernet uplink fabric mode for the specified fabric interconnect.
Step 3	UCS-A /eth-uplink/fabric # **scope vlan** *vlan-name*	Enters Ethernet uplink fabric VLAN mode.
Step 4	UCS-A /eth-uplink/fabric/vlan # **set mcastpolicy** *policy-name*	Assigns a multicast policy for the VLAN.
Step 5	UCS-A /eth-uplink/fabric/vlan # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example sets a named VLAN accessible to one fabric interconnect and commits the transaction:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric a
UCS-A /eth-uplink/fabric # scope vlan vlan1
UCS-A /eth-uplink/fabric/vlan # set mcastpolicy policy1
UCS-A /eth-uplink/fabric/vlan* # commit-buffer
UCS-A /eth-uplink/fabric/vlan #
```

Deleting a Multicast Policy

Note If you assigned a non-default (user-defined) multicast policy to a VLAN and then delete that multicast policy, the associated VLAN inherits the multicast policy settings from the default multicast policy until the deleted policy is re-created.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters organization mode for the specified organization.
Step 2	UCS-A /org # **delete mcast-policy** *policy-name*	Deletes a multicast policy with the specified policy name.

	Command or Action	Purpose
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to delete a multicast policy named policy1:

```
UCS-A # scope org /
UCS-A /org # delete mcast-policy policy1
UCS-A /org* # commit-buffer
UCS-A /org #
```

LACP Policy

Link Aggregation combines multiple network connections in parallel to increase throughput and to provide redundancy. Link aggregation control protocol (LACP) provides additional benefits for these link aggregation groups. Cisco UCS Manager enables you to configure LACP properties using LACP policy.

You can configure the following for a lacp policy:

- **Suspended-individual**: If you do not configure the ports on an upstream switch for lacp, the fabric interconnects treat all ports as uplink Ethernet ports to forward packets. You can place the lacp port in suspended state to avoid loops. When you set suspend-individual on a port-channel with lacp, if a port that is part of the port-channel does not receive PDUs from the peer port, it will go into suspended state.
- **Timer values**: You can configure rate-fast or rate-normal. In rate-fast configuration, the port is expected to receive 1 PDU every 1 second from the peer port. The time out for this is 3 seconds. In rate-normal configuration, the port is expected to receive 1 PDU every 30 seconds. The timeout for this is 90 seconds.

System creates a default lacp policy at system start up. You can modify this policy or create new. You can also apply one lacp policy to multiple port-channels.

Creating a LACP Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters the root organization mode.
Step 2	UCS-A /org # **create lacppolicy***policy nam*.	Creates the specified lacp policy.
Step 3	UCS-A /org # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example creates the lacp policy and commits the transaction:

```
UCS-A # scope org
UCS-A /org # create lacppolicy lacp1
UCS-A /org* # commit-buffer
UCS-A /org #
```

Editing a LACP Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org**	Enters the root organization mode.
Step 2	UCS-A /org # **scope lacppolicy** *policy-name* .	Enters the specified lacp policy.
Step 3	UCS-A /org/lacp policy/ policy-name # **set suspend-individual** *true* .	Sets suspend individual for the policy.
Step 4	UCS-A /org/lacp policy/ policy-name # **set lacp-rate** *fast* .	Sets LACP rate for the policy.
Step 5	UCS-A /org/lacp policy/ policy-name # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example modifies the lacp policy and commits transaction:

```
UCS-A# scope org
UCS-A/org # scope lacppolicy policy-name
UCS-A /org/lacp policy policy-name #  set suspend-individual true
UCS-A/prg/policy policy-name # set lacp-rate fast
UCS-A /org* # commit-buffer
UCS-A /org #
```

Assigning LACP Policy to Port-Channels

Default lacp policy is assigned to port channels by default. You can assign a different lacp policy to the port channel. If the assigned policy does not exist, system generates a fault. You can create the same policy to clear the fault.

Note You can assign lacp policy to port-channels, FCoE port-channels, and ethernet storage port-channels. This procedures describes assigning the lacp policy to port-channels.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric**	Enters the fabric mode.
Step 3	UCS-A /eth-uplink/fabric # **scope port-channel**	Enters the port-channel mode.
Step 4	UCS-A /eth-uplink/fabric/port-channel # **set lacp-policy-name***policy-name*	Specifies the lacp policy for this port-channel.
Step 5	UCS-A /eth-uplink/ fabric/port-channel **commit-buffer**	Commits the transaction to the system.

Example

The following example shows assigning a lacp policy to a port-channel:

```
UCS-A# scope eth-uplink
UCS-A UCS-A/eth-uplink # scope fabric
UCS-A UCS-A/eth-uplink/facric # scope port-channel
UCS-A UCS-A/eth-uplink/port-channel # set lacp-policy-name
UCS-A UCS-A/eth-uplink/port-channel* # commit-buffer
UCS-A UCS-A/eth-uplink/port-channel #
```

Configuring UDLD Link Policies

Understanding UDLD

UniDirectional Link Detection (UDLD) is a Layer 2 protocol that enables devices connected through fiber-optic or twisted-pair Ethernet cables to monitor the physical configuration of the cables and detect when a unidirectional link exists. All connected devices must support UDLD for the protocol to successfully identify and disable unidirectional links. When UDLD detects a unidirectional link, it marks the link as unidirectional. Unidirectional links can cause a variety of problems, including spanning-tree topology loops.

UDLD works with the Layer 1 mechanisms to determine the physical status of a link. At Layer 1, autonegotiation takes care of physical signaling and fault detection. UDLD performs tasks that autonegotiation cannot perform, such as detecting the identities of neighbors and shutting down misconnected interfaces. When you enable both autonegotiation and UDLD, the Layer 1 and Layer 2 detections work together to prevent physical and logical unidirectional connections and the malfunctioning of other protocols.

A unidirectional link occurs whenever traffic sent by a local device is received by its neighbor but traffic from the neighbor is not received by the local device.

Modes of Operation

UDLD supports two modes of operation: normal (the default) and aggressive. In normal mode, UDLD can detect unidirectional links due to misconnected interfaces on fiber-optic connections. In aggressive mode,

UDLD can also detect unidirectional links due to one-way traffic on fiber-optic and twisted-pair links and to misconnected interfaces on fiber-optic links.

In normal mode, UDLD detects a unidirectional link when fiber strands in a fiber-optic interface are misconnected and the Layer 1 mechanisms do not detect this misconnection. If the interfaces are connected correctly but the traffic is one way, UDLD does not detect the unidirectional link because the Layer 1 mechanism, which is supposed to detect this condition, does not do so. In case, the logical link is considered undetermined, and UDLD does not disable the interface. When UDLD is in normal mode, if one of the fiber strands in a pair is disconnected and autonegotiation is active, the link does not stay up because the Layer 1 mechanisms did not detect a physical problem with the link. In this case, UDLD does not take any action, and the logical link is considered undetermined.

UDLD aggressive mode is disabled by default. Configure UDLD aggressive mode only on point-to-point links between network devices that support UDLD aggressive mode. With UDLD aggressive mode enabled, when a port on a bidirectional link that has a UDLD neighbor relationship established stops receiving UDLD packets, UDLD tries to reestablish the connection with the neighbor and administratively shuts down the affected port. UDLD in aggressive mode can also detect a unidirectional link on a point-to-point link on which no failure between the two devices is allowed. It can also detect a unidirectional link when one of the following problems exists:

- On fiber-optic or twisted-pair links, one of the interfaces cannot send or receive traffic.
- On fiber-optic or twisted-pair links, one of the interfaces is down while the other is up.
- One of the fiber strands in the cable is disconnected.

Methods to Detect Unidirectional Links

UDLD operates by using two mechanisms:

- Neighbor database maintenance

  UDLD learns about other UDLD-capable neighbors by periodically sending a hello packet (also called an advertisement or probe) on every active interface to keep each device informed about its neighbors. When the switch receives a hello message, it caches the information until the age time (hold time or time-to-live) expires. If the switch receives a new hello message before an older cache entry ages, the switch replaces the older entry with the new one.

  UDLD clears all existing cache entries for the interfaces affected by the configuration change whenever an interface is disabled and UDLD is running, whenever UDLD is disabled on an interface, or whenever the switch is reset. UDLD sends at least one message to inform the neighbors to flush the part of their caches affected by the status change. The message is intended to keep the caches synchronized.

- Event-driven detection and echoing

  UDLD relies on echoing as its detection mechanism. Whenever a UDLD device learns about a new neighbor or receives a resynchronization request from an out-of-sync neighbor, it restarts the detection window on its side of the connection and sends echo messages in reply. Because this behavior is the same on all UDLD neighbors, the sender of the echoes expects to receive an echo in reply.

  If the detection window ends and no valid reply message is received, the link might shut down, depending on the UDLD mode. When UDLD is in normal mode, the link might be considered undetermined and might not be shut down. When UDLD is in aggressive mode, the link is considered unidirectional, and the interface is shut down.

If UDLD in normal mode is in the advertisement or in the detection phase and all the neighbor cache entries are aged out, UDLD restarts the link-up sequence to resynchronize with any potentially out-of-sync neighbors.

If you enable aggressive mode when all the neighbors of a port have aged out either in the advertisement or in the detection phase, UDLD restarts the link-up sequence to resynchronize with any potentially out-of-sync neighbor. UDLD shuts down the port if, after the fast train of messages, the link state is still undetermined.

UDLD Configuration Guidelines

The following guidelines and recommendations apply when you configure UDLD:

- A UDLD-capable interface also cannot detect a unidirectional link if it is connected to a UDLD-incapable port of another switch.
- When configuring the mode (normal or aggressive), make sure that the same mode is configured on both sides of the link.
- UDLD should be enabled only on interfaces that are connected to UDLD capable devices. The following interface types are supported:
  - Ethernet uplink
  - FCoE uplink
  - Ethernet uplink port channel member
  - FCoE uplink port channel member

Configuring a UDLD Link Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org /**	Enters the root organization mode.
Step 2	UCS-A /org # **create udld-link-policy** *link-policy-name*	Creates a UDLD link policy with the specified name, and enters UDLD link policy mode.
Step 3	UCS-A /org/udld-link-policy # **commit-buffer**	Commits the transaction to the system configuration.
Step 4	UCS-A /org/udld-link-policy # **exit**	Returns to the previous mode.
Step 5	UCS-A /org # **scope udld-link-policy** *link-policy-name*	Enters UDLD link policy mode for the specified UDLD link policy.
Step 6	UCS-A /org/udld-link-policy # **set mode** {**aggressive** | **normal**}	Specifies the mode for the UDLD link policy.
Step 7	UCS-A /org/udld-link-policy # **set admin-state** {**disabled** | **enabled**}	Disables or enables UDLD on the interface.

	Command or Action	Purpose
Step 8	UCS-A /org/udld-link-policy # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to create a link profile called UDLDPol1, sets the mode to aggressive, and enables UDLD on the interface.

```
UCS-A# scope org /
UCS-A /chassis/org # create udld-link-policy UDLDPol1
UCS-A /chassis/org/udld-link-policy* # commit-buffer
UCS-A /chassis/org/udld-link-policy # exit
UCS-A /chassis/org # scope udld-link-policy UDLDPol1
UCS-A /chassis/org/udld-link-policy # set mode aggressive
UCS-A /chassis/org/udld-link-policy* # set admin-state enabled
UCS-A /chassis/org/udld-link-policy* # commit-buffer
UCS-A /chassis/org/udld-link-policy #
```

Modifying the UDLD System Settings

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** /	Enters the root organization mode.
Step 2	UCS-A /org # **show udld-policy**	Displays the current UDLD system settings.
Step 3	UCS-A /org # **scope udld-policy default**	Enters UDLD policy mode for the global UDLD policy.
Step 4	UCS-A /org/udld-policy # **set message-interval** *seconds*	Specifies the time interval (in seconds) between UDLD probe messages on ports that are in advertisement mode. Enter an integer between 7 and 60. The default is 15 seconds.
Step 5	UCS-A /org/udld-policy # **set recovery-action** [**reset** | **none**]	Specifies the action to be taken on any ports that are disabled when UDLD aggressive mode is enabled. The default is none.
Step 6	UCS-A /org/udld-policy # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to update the default UDLD system settings for a 30 second time interval.

```
UCS-A# scope org /
UCS-A /chassis/org # show udld-policy

UDLD system settings:
    Name       Message interval (sec) Recovery action
    ---------- ---------------------- ---------------
    default    15                     None

UCS-A /chassis/org # scope udld-policy default
UCS-A /chassis/org/udld-policy # set message-interval 30
UCS-A /chassis/org/udld-policy* # commit-buffer
UCS-A /chassis/org/udld-policy #
```

Configuring a Link Profile

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** /	Enters the root organization mode.
Step 2	UCS-A /org # **create eth-link-profile** *link-profile-name*	Creates a link profile with the specified name, and enters link profile mode.
Step 3	UCS-A /org/eth-link-profile # **commit-buffer**	Commits the transaction to the system configuration.
Step 4	UCS-A /org/eth-link-profile # **exit**	Returns to the previous mode.
Step 5	UCS-A /org # **scope eth-link-profile** *link-profile-name*	Enters link profile mode for the specified link profile.
Step 6	UCS-A /org/eth-link-profile # **set udld-link-policy** *link-policy-name*	Assigns the specified UDLD link policy to the link profile.
Step 7	UCS-A /org/eth-link-profile # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to create a link profile called LinkProfile1 and assign the default UDLD link policy.

```
UCS-A# scope org /
UCS-A /chassis/org # create eth-link-profile LinkProfile1
UCS-A /chassis/org/eth-link-profile* # commit-buffer
UCS-A /chassis/org/eth-link-profile # exit
UCS-A /chassis/org # scope eth-link-profile LinkProfile1
UCS-A /chassis/org/eth-link-profile # set udld-link-policy default
UCS-A /chassis/org/eth-link-profile* # commit-buffer
```

Assigning a Link Profile to a Port Channel Ethernet Interface

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric** {**a** | **b**}	Enters Ethernet uplink fabric mode for the specified fabric.
Step 3	UCS-A /eth-uplink/fabric # **scope port-channel** *port-chan-id*	Enters Ethernet uplink fabric port channel mode for the specified port channel.
Step 4	UCS-A /eth-uplink/fabric/port-channel # **scope member-port** *slot-id port-id*	Enters Ethernet server fabric, fabric port channel mode for the specified member port.
Step 5	UCS-A /eth-uplink/fabric/port-channel/member-port # **set eth-link-profile** *link-profile-name*	Assigns the specified link profile.
Step 6	UCS-A /eth-uplink/fabric/port-channel/member-port # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to assign link profile LinkProfile1 to a port channel Ethernet interface:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric a
UCS-A /eth-uplink/fabric # scope port-channel 88
UCS-A /eth-uplink/fabric/port-channel # scope member-port 1 31
UCS-A /eth-uplink/fabric/port-channel/member-port # set eth-link-profile LinkProfile1
UCS-A /eth-uplink/fabric/port-channel/member-port* # commit-buffer
UCS-A /eth-uplink/fabric/port-channel/member-port #
```

Assigning a Link Profile to a Port Channel FCoE Interface

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters Fibre Channel uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric** {**a** | **b**}	Enters Fibre Channel uplink fabric mode for the specified fabric.
Step 3	UCS-A /fc-uplink/fabric # **scope fcoe-port-channel** *port-chan-id*	Enters Fibre Channel uplink fabric port channel mode for the specified port channel.

	Command or Action	Purpose
Step 4	UCS-A /fc-uplink/fabric/fcoe-port-channel # **scope fcoe-member-port** *slot-id port-id*	Enters Fibre Channel server fabric, fabric port channel mode for the specified member port.
Step 5	UCS-A /fc-uplink/fabric/fcoe-port-channel/fcoe-member-port # **set eth-link-profile** *link-profile-name*	Assigns the specified link profile.
Step 6	UCS-A /fc-uplink/fabric/fcoe-port-channel/fcoe-member-port # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to assign link profile LinkProfile1 to a port channel FCoE interface:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # scope fcoe-port-channel 192
UCS-A /fc-uplink/fabric/fcoe-port-channel # scope fcoe-member-port 1 20
UCS-A /fc-uplink/fabric/fcoe-port-channel/fcoe-member-port # set eth-link-profile LinkProfile1
UCS-A /fc-uplink/fabric/fcoe-port-channel/fcoe-member-port* # commit-buffer
UCS-A /fc-uplink/fabric/fcoe-port-channel/fcoe-member-port #
```

Assigning a Link Profile to an Uplink Ethernet Interface

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric** {**a** | **b**}	Enters Ethernet uplink fabric mode for the specified fabric.
Step 3	UCS-A /eth-uplink/fabric # **scope interface** *slot-num port num*	Enters the interface command mode for the specified uplink port.
Step 4	UCS-A /eth-uplink/fabric/interface # **set eth-link-profile** *link-profile-name*	Assigns the specified link profile.
Step 5	UCS-A /eth-uplink/fabric/interface # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to assign link profile LinkProfile1 to an uplink Ethernet interface:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric a
UCS-A /eth-uplink/fabric # scope interface 2 2
UCS-A /eth-uplink/fabric/interface # set eth-link-profile LinkProfile1
```

```
UCS-A /eth-uplink/fabric/interface* # commit-buffer
UCS-A /eth-uplink/fabric/interface #
```

Assigning a Link Profile to an Uplink FCoE Interface

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope fc-uplink**	Enters Fibre Channel uplink mode.
Step 2	UCS-A /fc-uplink # **scope fabric** {**a** | **b**}	Enters Fibre Channel uplink fabric mode for the specified fabric.
Step 3	UCS-A /fc-uplink/fabric # **scope fcoeinterface** *slot-num port num*	Enters the Fibre Channel interface command mode for the specified uplink port.
Step 4	UCS-A /fc-uplink/fabric/fcoeinterface # **set eth-link-profile** *link-profile-name*	Assigns the specified link profile.
Step 5	UCS-A /fc-uplink/fabric/fcoeinterface # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to assign link profile LinkProfile1 to an uplink FCoE interface:

```
UCS-A# scope fc-uplink
UCS-A /fc-uplink # scope fabric a
UCS-A /fc-uplink/fabric # scope fcoeinterface 2 2
UCS-A /fc-uplink/fabric/fcoeinterface # set eth-link-profile LinkProfile1
UCS-A /fc-uplink/fabric/fcoeinterface* # commit-buffer
UCS-A /fc-uplink/fabric/fcoeinterface #
```

VMQ Connection Policy

Cisco UCS Manager enables you to configure VMQ connection policy for a vNIC. VMQ provides improved network performance to the entire management operating system. Configuring a VMQ vNIC connection policy involves the following:

- Create a VMQ connection policy
- Create a static vNIC in a service profile
- Apply the VMQ connection policy to the vNIC

If you want to configure the VMQ vNIC on a service profile for a server, at least one adapter in the server must support VMQ. Make sure the servers have at least one the following adapters installed:

- UCS-VIC-M82-8P
- UCSB-MLOM-40G-01

- UCSC-PCIE-CSC-02

The following are the supported Operating Systems for VMQ:

- Windows 2012
- Windows 2012R2

You can apply only any one of the vNIC connection policies on a service profile at any one time. Make sure to select one of the three options such as Dynamic, usNIC or VMQ connection policy for the vNIC. When a VMQ vNIC is configured on service profile, make sure you have the following settings:

- Select SRIOV in the BIOS policy.
- Select Windows in the Adapter policy.

Creating a VMQ Connection Policy

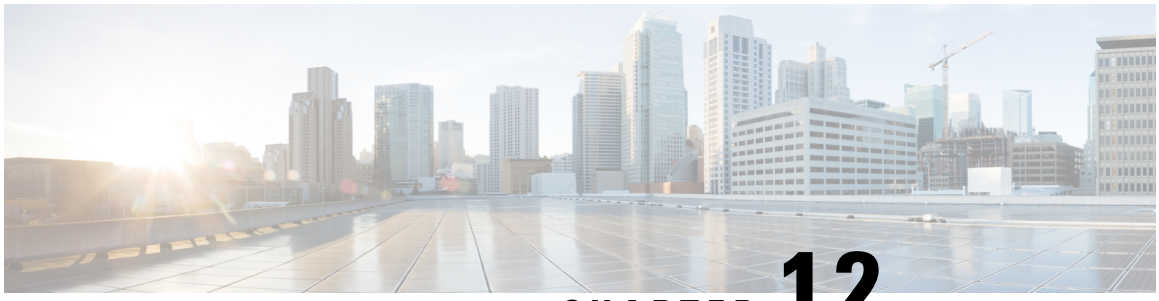
Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope org** *org-name*	Enters organization mode for the specified organization. To enter the root organization mode, type **/** as the *org-name*.
Step 2	UCS-A /org # **create vmq-conn-policy** *policy-name*	Specifies the name for this VMQ connection policy.
Step 3	UCS-A /org/vmq-conn-policy* # **set queue-count***queue count*	Specifies the queue count for the VMQ connection policy.
Step 4	UCS-A /org/vmq-conn-policy* # **set interrupt-count***interrupt count*	Specifies the interrupt count for the VMQ connection policy.
Step 5	UCS-A /org/vmq-conn-policy* # **commit-buffer**	Commits the transaction to the system.

Example

The following example creates a VMQ connection policy:

```
UCS-A# scope org
UCS-A /org # create vmq-conn-policy policy name
UCS-A /org/vmq-conn-policy* # set queue-count queue count (number)
UCS-A /org/vmq-conn-policy* # set interrupt-count queue count (number)
UCS-A /org/vmq-conn-policy* # commit-buffer
UCS-A /org/vmq-conn-policy #
```



CHAPTER 12

Configuring MACsec

- About MACsec, on page 251
- Guidelines and Limitations for MACsec, on page 252
- Enabling MACsec Configuration, on page 255
- Disabling MACsec Configuration, on page 255
- Creating a MACsec Policy, on page 256
- Viewing MACsec Policy, on page 258
- Deleting a MACsec Policy, on page 258
- Creating a MACsec Keychain, on page 259
- Viewing a MACsec Keychain, on page 259
- Deleting a MACsec Keychain, on page 260
- Creating a MACsec Key, on page 260
- Viewing MACsec Keys, on page 262
- Deleting a MACsec Key, on page 263
- Creating a LifeTime, on page 263
- Viewing a LifeTime, on page 264
- Deleting a LifeTime, on page 265
- Creating a MACsec Interface Configuration, on page 265
- Viewing MACsec Interface Configuration, on page 267
- Deleting a MACsec Interface Configuration, on page 267
- Configuring MACsec on an Uplink Interface, on page 268
- Viewing MACsec on an Uplink Interface, on page 268
- Deleting MACsec on an Uplink Interface, on page 269
- Configuring MACsec on an Uplink Port Channel Member Interface, on page 270
- Viewing MACsec on an Uplink Port Channel Member Interface, on page 271
- Deleting MACsec on an Uplink Port Channel Member Interface, on page 271
- Configurable EAPOL Destination and Ethernet Type, on page 272
- Displaying MACsec Sessions, on page 274
- Displaying MACsec Statistics, on page 275

About MACsec

MACsec is an IEEE 802.1AE standards-based Layer 2 hop-by-hop encryption that provides data confidentiality and integrity for media access independent protocols.

MACsec 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.

MACsec encrypts the entire data except for the Source and Destination MAC addresses of an Ethernet packet. It offers the following capabilities:

- Provides line rate encryption.
- Ensures data confidentiality by providing strong encryption at Layer 2.
- Provides integrity checking to help ensure that data cannot be modified in transit.
- Key Lifetime and Hitless Key Rollover, on page 252
- Fallback Key, on page 252

Key Lifetime and Hitless Key Rollover

A MACsec keychain can have multiple pre-shared keys (PSKs), each configured with a key ID and an optional lifetime. A key lifetime specifies at which time the key activates and expires. In the absence of a lifetime configuration, the default lifetime is unlimited. When a lifetime is configured, MKA rolls over to the next configured pre-shared key in the keychain after the lifetime is expired. The time zone of the key can be local or UTC. The default time zone is UTC.

To configure a MACsec keychain, see Creating a MACsec Keychain, on page 259

A key can roll over to a second key within the same keychain by configuring the second key (in the keychain) and configuring 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, the key rolls over without traffic interruption).

Note The lifetime of the keys are overlapped to achieve hitless key rollover.

Fallback Key

A MACsec session can fail due to a key/key ID (CKN) mismatch or a finite key duration between the Fabric Interconnect and the peer. If a MACsec session fails, a fallback session can take over if a fallback key is configured. A fallback session prevents downtime due to primary session failure and allows a user time to fix the key issue causing the failure. A fallback key also provides a backup session if the primary session fails to start. This feature is optional.

For more information, see Creating a MACsec Keychain.

Guidelines and Limitations for MACsec

MACsec functionality supports the following:

- Ethernet Uplink interfaces
- Ethernet Port-channel member link interfaces

- MKA is the only supported key exchange protocol for MACsec.

Note The Security Association Protocol (SAP) is not supported.

MACsec functionality does not support the following:

- Unified uplink
- FCoE uplinks
- Server, Storage, and Appliance ports
- QSA
- Link-level flow control (LLFC) and priority flow control (PFC)
- Multiple MACsec peers (different SCI values) for the same interface
- 1G port or any port on a MAC block that has 1G ports on it.

Note MACsec configuration is supported only on end host mode.

MACsec Support with Cisco UCS Fabric Interconnects

Supported on the following Fabric Interconnects:

- Cisco UCS 6652 Fabric Interconnect (UCS Manager 6.0(2b) release)
- Cisco UCS 6664 Fabric Interconnect and Cisco UCS Fabric Interconnect 9108 100G (Cisco UCS X-Series Direct) Fabric Interconnect (UCS Manager 6.0(1b) release)

Not Supported on Cisco UCS 6300 Series and Cisco UCS 6324 Fabric Interconnects.

Keychain Limitations

- You cannot overwrite the Key Hex String when the MACsec Keychain is applied on the interface. Instead, you must delete the old key and create the new key or a new keychain.
- For a given keychain, key activation time must overlap to avoid any period of time when no key is activated. If a time period occurs during which no key is activated, session negotiation fails and traffic drops can occur. The key with the latest start time among the currently active keys takes precedence for a MACsec key rollover.
- A MACSec session cannot be established if the CKN (Key ID) or CAK (Key Hex String) is set to all zeros.

Fallback Limitations

- If a MACsec session is secured on an old primary key, it does not go to a fallback session in case of mismatched latest active primary key. So the session remains secured on the old primary key and shows

as rekeying on the old CA (Connectivity Association) under status. And the MACsec session on the new key on primary PSK will be in the Init state.

- Use only one key with infinite lifetime in the fallback key chain. Multiple keys are not supported.
- The key ID (CKN) used in the fallback key chain must not match with any of the key IDs (CKNs) used in the primary key chain of the same switch interface and peer upstream switch interface.
- Once configured, fallback configuration on an interface cannot be removed, unless the complete MACsec configuration on the interface is removed.

MACsec Policy Limitations

- BPDU packets can be transmitted before a MACsec session becomes secure.
- We recommend you to apply the same security policy **Should Secure-Should Secure** or **Must Secure-Must Secure** on the fabric interconnect and the peer switch interface.
- While making changes to the MACSec policy parameters, do not change the **Key Server Priority** along with other parameters if the policy is already applied to any of the uplinks.

Note Configuring MACsec with security-policy as **must-secure** on an Uplink Interface brings down the port, and the traffic drops until the MACsec session is secured.

Layer 2 Tunneling Protocol (L2TP) Restrictions

MACsec is not supported on ports that are configured for dot1q tunneling or L2TP.

MACsec EAPOL Limitations

- For enabling EAPOL (Extensible Authentication Protocol over LAN) configuration, the range of Ethernet type between 0 to 0x599 is invalid.
- While configuring EAPOL packets, the following combinations must not be used:
  - MAC Address 0100.0ccd.cdd0 with any ethertype
  - Any MAC Address with Ether types: 0xfff0, 0x800, 0x86dd
  - The default destination MAC address, 0180.c200.0003 with the default Ethernet type, 0x888e
  - Different EAPOL DMAC addresses and Ethertype on both MACsec peers. The MACsec session works only if the MACsec peer is sending MKAPDUs with the DMAC and Ethertype configured locally.
  - Within the same slice of the forwarding engine, EAPOL ethertype and dot1q ethertype cannot have the same value.
  - More than one custom EAPOL is not supported.
  - You cannot modify a custom EAPOL configuration if applied on any interface.

Statistics Limitations

- Statistics are cumulative.
- Few CRC errors may occur during the transition between MACsec and non-MACsec mode (regular port shut/no shut).
- The IEEE8021-SECY-MIB OIDs secyRxSAStatsOKPkts, secyTxSAStatsProtectedPkts, and secyTxSAStatsEncryptedPkts can carry only up to 32 bits of counter values, but the traffic may exceed 32 bits.

Enabling MACsec Configuration

Before you can access the MACsec commands, you must enable MACsec.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope macsec**	Enters the MACsec mode.
Step 2	UCS-A /macsec # **enable**	Enable MACsec.
Step 3	UCS-A /macsec* # **commit-buffer**	Commits the transaction to the system configuration.
Step 4	UCS-A /macsec # **show**	Displays the MACsec configuration.

Example

The following example enables a MACsec configuration:

```
UCS-A# scope macsec
UCS-A /macsec# enable
UCS-A /macsec* # commit-buffer
UCS-A /macsec# show

MACsec Feature:
Admin State
-----------
Enabled
UCS-A /macsec
```

Disabling MACsec Configuration

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope macsec**	Enters the MACsec mode.

	Command or Action	Purpose
Step 2	UCS-A /macsec # **disable**	Disables MACsec.
Step 3	UCS-A /macsec* # **commit-buffer**	Commits the transaction to the system configuration.
Step 4	UCS-A /macsec # **show**	Displays the MACsec configuration.

Example

The following example disables the MACsec encryption and commits the transaction:

```
UCS-A# scope macsec
UCS-A /macsec # disable
UCS-A /macsec* # commit-buffer
UCS-A /macsec# show

MACsec Feature:
Admin State
-----------
Disabled
UCS-A /macsec
```

Creating a MACsec Policy

You can create multiple MACsec policies with different parameters. However, only one policy can be active on an interface.

Before you begin

Ensure that MACsec is enabled.

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope macsec**	Enters the MACsec mode.
Step 2	UCS-A /macsec # **create macsec-policy** *<name>*	Creates a MACsec policy.
Step 3	UCS-A /macsec/macsec-policy* # **set cipher-suite** { **gcm-aes-xpn-256** | **gcm-aes-xpn-128** | **gcm-aes-256** | **gcm-aes-128** }	Configure the cipher suite to be used for MACsec encryption. Configures one of the following ciphers: GCM-AES-128, GCM-AES-256, GCM-AES-XPN-128, or GCM-AES-XPN-256.
Step 4	UCS-A /macsec/macsec-policy* # **set key-server-priority** *<0-255>*	Enter the key server priority. You can enter a value between 0-255. Lower the value, higher the preference to be selected as the key server.

	Command or Action	Purpose
		Configures the key server priority to break the tie between peers during a key exchange. The range is from 0 (highest) and 255 (lowest), and the default value is 16.
Step 5	UCS-A /macsec/macsec-policy* # **set security-policy** { **should-secure** | **must-secure** }	Configures one of the following security policies to define the handling of data and control packets: • **must-secure**—Packets that do not carry MACsec headers are dropped. • **should-secure**—Packets that do not carry MACsec headers are permitted. This is the default value.
Step 6	UCS-A /macsec/macsec-policy* # **set replay-window-size** *<0-596000000>*	Configures the replay protection window such that the secured interface does not accept any packet that is less than the configured window size. The range is from 0 to 596000000.
Step 7	UCS-A /macsec/macsec-policy* # **set sak-expiry-time** *<60-2592000>*	Configures the time in seconds to force an SAK rekey. This command can be used to change the session key to a predictable time interval. The default is 0.
Step 8	UCS-A /macsec/macsec-policy* # **set confidentiality-offset** { **conf-offset-0** | **conf-offset-30** | **conf-offset-50** }	Configures one of the following confidentiality offsets in the Layer 2 frame, where encryption begins: CONF-OFFSET-0, CONF-OFFSET-30, or CONF-OFFSET-50.
Step 9	UCS-A /macsec/macsec-policy* # **set include-icv-indicator** { **yes** | **no** }	Configure the ICV for the frame arriving on the port.
Step 10	UCS-A /macsec/macsec-policy* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to enable a MACsec policy:

```
UCS-A # scope macsec
UCS-A /macsec # create macsec-policy macsec_policy
UCS-A /macsec/macsec-policy* # set cipher-suite gcm-aes-xpn-256
UCS-A /macsec/macsec-policy* # set key-server-priority 16
UCS-A /macsec/macsec-policy* # set security-policy should-secure
UCS-A /macsec/macsec-policy* # set replay-window-size 0
UCS-A /macsec/macsec-policy* # set sak-expiry-time 60
UCS-A /macsec/macsec-policy* # set confidentiality-offset conf-offset-0
UCS-A /macsec/macsec-policy* # set include-icv-indicator yes
UCS-A /macsec/macsec-policy* # commit-buffer
UCS-A /macsec/macsec-policy #
```

Viewing MACsec Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope macsec**	Enters the MACsec mode.
Step 2	UCS-A /macsec # **show macsec-policy**	Displays the MACsec policy details.

Example

The following example shows how to view a MACsec policy:

```
UCS-A # scope macsec
UCS-A /macsec # show macsec-policy

MACsec Policy:
    MACsec Policy Name Cipher Suite    Key Server Priority Security Policy Repla
y Window Size SAK Expiry Time Confidentiality Offset Include ICV Indicator
    ------------------ --------------- ------------------- --------------- -----
------------ --------------- ---------------------- ---------------------
    default            GCM AES XPN 256 16                  Should Secure   14880
9600          0               Conf Offset 0          No
    test1              GCM AES XPN 256 16                  Should Secure   14880
9600          61              Conf Offset 0          No

UCS-A /macsec* #
```

Deleting a MACsec Policy

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope macsec**	Enters the MACsec mode.
Step 2	UCS-A /macsec # **delete macsec-policy** *<name>*	Deletes a MACsec policy.
Step 3	UCS-A /macsec # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to delete a MACsec policy:

```
UCS-A # scope macsec
UCS-A /macsec # delete macsec-policy macsec_policy
```

```
UCS-A /macsec* # commit-buffer
UCS-A /macsec #
```

Creating a MACsec Keychain

- Only MACsec keychains result in converged MKA sessions.
- You can create a MACsec keychain and keys on the device.

Before you begin

Ensure that MACsec is enabled.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope macsec**	Enters the MACsec mode.
Step 2	UCS-A /macsec # **create macsec-keychain** *<name>*	Creates a MACsec keychain to hold a set of MACsec keys and enters MACsec keychain configuration mode.
Step 3	UCS-A /macsec* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to create a MACsec Keychain, and commits the transaction:

```
UCS-A# scope macsec
UCS-A /macsec # create macsec-keychain kc
UCS-A /macsec* # commit-buffer
UCS-A /macsec #
```

Viewing a MACsec Keychain

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope macsec**	Enters the MACsec mode.
Step 2	UCS-A /macsec # **show macsec-keychain**	Displays the MACsec keychain details.

Example

The following example shows how to view a MACsec keychain:

```
UCS-A# scope macsec
UCS-A /macsec #  show macsec-keychain

Keychain:
    Keychain Name
    -------------
    test-kc-1
    test-kc-2
    test1

UCS-A /macsec #
```

Deleting a MACsec Keychain

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope macsec**	Enters the MACsec mode.
Step 2	UCS-A /macsec # **delete macsec-keychain** *<name>*	Deletes the MACsec Keychain.
Step 3	UCS-A /macsec* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to delete a MACsec keychain:

```
UCS-A# scope macsec
UCS-A /macsec # delete macsec-keychain kc
UCS-A /macsec* # commit-buffer
UCS-A /macsec #
```

Creating a MACsec Key

You can create a MACsec key on the device.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope macsec**	Enters the MACsec mode.

	Command or Action	Purpose
Step 2	UCS-A /macsec # **create macsec-keychain** <*name*>	Creates a MACsec keychain to hold a set of MACsec keys and enters MACsec keychain configuration mode.
Step 3	UCS-A /macsec/macsec-keychain* # **create macsec-key** <*id*>	Creates a MACsec key and enters MACsec key configuration mode. The range is from 1 to 32 octets, and the maximum size is 64. A total of 64 Key ids can be configured per MACsec Keychain. The key must consist of an even number of characters. **Note** The key must consist of an even number of characters.
Step 4	UCS-A /macsec/macsec-keychain* # **set key-hex-string** <*key*>	Set the key between 32 and 144 hexadecimal characters. The key length is based on the encryption type and cryptographic algorithm. Type 0 (Unencrypted Key) • AES_128_CMAC: 32 hexadecimal characters • AES_256_CMAC: 64 hexadecimal characters Type 7 • AES_128_CMAC: 66 hexadecimal characters • AES_256_CMAC: 130 hexadecimal characters Type 6 • AES_128_CMAC: 100 hexadecimal characters • AES_256_CMAC: 144 hexadecimal characters
Step 5	UCS-A /macsec/macsec-keychain* # **set encrypt-type** { **type-0** | **type-7** } | **type-6** }	The encrypt type includes the following: • Type 0—Set the encrypt type as type 0 to configure key-hex-string as an unencrypted string. • Type 7—Set the encrypt type as type 7 to configure key-hex-string as an encrypted string. • Type 6—Set the encrypt type as type 6 to configure key-hex-string as an AES

	Command or Action	Purpose
		encrypted string. The type 6 encryption utilizes the Advanced Encryption Standard (AES) for an enhanced security. For more information, see the *Creating an AES Encryption* section in Cisco UCS Manager Administration Management Guide 4.3.
Step 6	UCS-A /macsec/macsec-keychain* # **set cryptographic-algorithm** { **aes-128-cmac** | **aes-256-cmac** }	Set cryptographic authentication algorithm with 128-bit or 256-bit encryption.
Step 7	UCS-A /macsec/macsec-keychain* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to create a MACsec key:

```
UCS-A# scope macsec
UCS-A /macsec # create macsec-keychain kc
UCS-A /macsec/macsec-keychain* # create macsec-key 10
UCS-A /macsec/macsec-keychain/macsec-key* # set key
abcdef0123456789abcdef0123456789abcdef0123456789abcdef0123456789
UCS-A /macsec/macsec-keychain/macsec-key* # set encrypt-type type-0
UCS-A /macsec/macsec-keychain/macsec-key* # set cryptographic-algorithm aes-256-cmac
UCS-A /macsec/macsec-keychain/macsec-key* # commit-buffer
UCS-A /macsec/macsec-keychain/macsec-key #
```

Viewing MACsec Keys

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope macsec**	Enters the MACsec mode.
Step 2	UCS-A /macsec # **scope macsec-keychain** *<name>*	Enters the MACsec keychain configuration mode.
Step 3	UCS-A /macsec/macsec-keychain* # **show macsec-key**	Displays the MACsec key configuration details.

Example

The following example shows how to view a MACsec key:

```
UCS-A# scope macsec
UCS-A /macsec # scope macsec-keychain kc
UCS-A /macsec/macsec-keychain* # show macsec-key
```

```
MACsec Key:
    Key ID     Key Hex String Encryption Type Cryptographic Algorithm
    ---------- -------------- --------------- ----------------------
    11         ****           Type 0          AES 256 CMAC
```

Deleting a MACsec Key

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope macsec**	Enters the MACsec mode.
Step 2	UCS-A /macsec # **scope macsec-keychain** *<name>*	Enters the MACsec keychain configuration mode.
Step 3	UCS-A /macsec/macsec-keychain # **delete macsec-key** *<id>*	Deletes a MACsec Key.
Step 4	UCS-A /macsec/macsec-keychain* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to delete a MACsec Key:

```
UCS-A# scope macsec
UCS-A /macsec # scope macsec-keychain kc
UCS-A /macsec/macsec-keychain # delete macsec-key 10
UCS-A /macsec/macsec-keychain/macsec-key* # commit-buffer
UCS-A /macsec/macsec-keychain/macsec-key #
```

Creating a LifeTime

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope macsec**	Enters the MACsec mode.
Step 2	UCS-A /macsec # **scope macsec-keychain** *<name>*	Enters the MACsec Keychain configuration mode.
Step 3	UCS-A /macsec/macsec-keychain # **scope macsec-key** *<id>*	Enters the MACsec Key ID.
Step 4	UCS-A /macsec/macsec-keychain/macsec-key # **create life-time**	Creates a MACsec Key Lifetime.

	Command or Action	Purpose
Step 5	UCS-A /macsec/macsec-keychain/macsec-key* # **set start-date-time***jan 1 2024 0 0 0*	The start-time argument is the time of day and date that the key becomes active.
Step 6	UCS-A /macsec/macsec-keychain/macsec-key* # **set end-date-time** *jan 2 2024 0 0 0*	The end-time argument is the time of day and date that the key becomes active.
Step 7	UCS-A /macsec/macsec-keychain/macsec-key* # **set duration** *<0-2147483646>*	The duration argument is the length of the lifetime in seconds. The maximum length is 2147483646 seconds (approximately 68 years).
Step 8	UCS-A /macsec/macsec-keychain/macsec-key* # **set timezone** { **local** | **UTC** }	The time zone of the key can be local or UTC. The default time zone is UTC.
Step 9	UCS-A /macsec/macsec-keychain/macsec-key* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to create a Lifetime:

```
UCS-A# scope macsec
UCS-A /macsec # scope macsec-keychain kc
UCS-A /macsec/macsec-keychain* # scope macsec-key 10
UCS-A /macsec/macsec-keychain/macsec-key* # create life-time
UCS-A /macsec/macsec-keychain/macsec-key/life-time* # set start-date-time jan 1 2024 0 0 0
UCS-A /macsec/macsec-keychain/macsec-key/life-time* # set end-date-time jan 2 2024 0 0 0
UCS-A /macsec/macsec-keychain/macsec-key/life-time* # set timezone local
UCS-A /macsec/macsec-keychain/macsec-key/life-time* # commit-buffer
UCS-A /macsec/macsec-keychain/macsec-key/life-time #
```

Viewing a LifeTime

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope macsec**	Enters the MACsec mode.
Step 2	UCS-A /macsec # **scope macsec-keychain** *<name>*	Enters the MACsec keychain configuration mode.
Step 3	UCS-A /macsec/macsec-keychain # **scope macsec-key** *<id>*	Enters the MACsec key configuration mode.
Step 4	UCS-A /macsec/macsec-keychain/macsec-key # **show life-time**	Displays the Lifetime details.

Example

The following example shows how to view a Lifetime:

```
UCS-A# scope macsec
UCS-A /macsec # scope macsec-keychain kc
UCS-A /macsec/macsec-keychain # scope macsec-key 11
UCS-A /macsec/macsec-keychain/macsec-key # show life-time

Life Time:
    Start Date Time         End Date Time           Timezone Duration(sec)
    ----------------------- ----------------------- -------- -------------
    2024-04-08T16:55:38.000 2024-04-08T16:55:38.000 Local    0
UCS-A /macsec/macsec-keychain/macsec-key #
```

Deleting a LifeTime

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope macsec**	Enters the MACsec mode.
Step 2	UCS-A /macsec # **scope macsec-keychain** *<name>*	Enters the MACsec keychain configuration mode.
Step 3	UCS-A /macsec/macsec-keychain # **scope macsec-key** *<id>*	Enters the MACsec key configuration mode.
Step 4	UCS-A /macsec/macsec-keychain/macsec-key # **delete life-time**	Deletes the Lifetime.
Step 5	UCS-A /macsec/macsec-keychain/macsec-key* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to delete a Lifetime:

```
UCS-A# scope macsec
UCS-A /macsec # scope macsec-keychain kc
UCS-A /macsec/macsec-keychain # scope macsec-key 10
UCS-A /macsec/macsec-keychain/macsec-key # delete life-time
UCS-A /macsec/macsec-keychain/macsec-key* # commit-buffer
UCS-A /macsec/macsec-keychain/macsec-key #
```

Creating a MACsec Interface Configuration

You can create multiple MACsec policies with different parameters. However, only one policy can be active on an interface.

Before you begin

Ensure that MACsec is enabled.

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope macsec**	Enters the MACsec mode.
Step 2	UCS-A /macsec# **create macsec-interface-config** <*name*>	Create a MACsec interface configuration.
Step 3	UCS-A /macsec/macsec-interface-config* # **set key-chain-name** <*macsec-keychain-name*>	Sets the MACsec keychain name for the specified MACsec policy.
Step 4	UCS-A /macsec/macsec-interface-config* # **set policy-name** <*macsec-policy*>	Sets the MACsec policy name for the specified MACsec policy.
Step 5	UCS-A /macsec/macsec-interface-config* # **set fallback-keychain-name** <*macsec-keychain-name*>	Applies the MACsec configuration on a physical interface with a fallback keychain. It is optional to configure a fallback PSK. If a fallback keychain is configured, the fallback keychain along with the primary keychain ensures that the session remains active even if the primary keychain is mismatched, or there is no active key for the primary keychain.
Step 6	UCS-A /macsec/macsec-interface-config* # **set eapol-name** <*eapol-name*>	Applies the MACsec configuration on a physical interface with an EAPOL configuration. For more information on MACsec EAPOL, see Configurable EAPOL Destination and Ethernet Type.
Step 7	UCS-A /macsec/macsec-interface-config* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example creates a MACsec interface configuration:

```
UCS-A scope macsec
UCS-A /macsec # create macsec-interface-config macsec_ifconfig
UCS-A /macsec/macsec-interface-config* # set key-chain-name kc
UCS-A /macsec/macsec-interface-config* # set policy-name macsec-policy
UCS-A /macsec/macsec-interface-config* # set fallback-keychain-name fb_kc
UCS-A /macsec/macsec-interface-config* # commit-buffer
UCS-A /macsec/macsec-interface-config #
```

Viewing MACsec Interface Configuration

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope macsec**	Enters the MACsec mode.
Step 2	UCS-A /macsec# **show macsec-interface-config**	Displays the MACsec interface configuration details.

Example

The following example shows how to view a MACsec interface configuration:

```
UCS-A# scope macsec
UCS-A /macsec # show macsec-interface-config

Interface Configuration:
Interface Configuration Name Interface Keychain Name Interface Policy Name Fallback Keychain
 Name EAPOL Name
---------------------------- ----------------------- ---------------------
---------------------- ----------
cus-eapol-m-t0 keychain-type0-aes128 mp-must fallback-type0-aes128 custom
cus-eapol-s-t7 keychain-type7-aes256 mp-should fallback-type7-aes256 custom
custom-eapol keychain-type0-aes256 mp-must fallback-type0-aes256 custom
dummy-config dummy-key default default
mic-m-t0-aes128 keychain-type0-aes128 mp-must fallback-type0-aes128 default
mic-m-t0-aes256 keychain-type0-aes256 mp-must fallback-type0-aes256 default
```

Deleting a MACsec Interface Configuration

Procedure

	Command or Action	Purpose
Step 1	UCS-A # **scope macsec**	Enters the MACsec mode.
Step 2	UCS-A /macsec# **delete macsec-interface-config** *<name>*	Deletes a MACsec interface configuration mode.
Step 3	UCS-A /macsec* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to delete a MACsec interface configuration:

```
UCS-A scope macsec
UCS-A /macsec # delete macsec-interface-config macsec_ifconfig
UCS-A /macsec* # commit-buffer
UCS-A /macsec #
```

Configuring MACsec on an Uplink Interface

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters ethernet uplink mode.
Step 2	UCS-A# /eth-uplink/fabric # **scope fabric** {**a** | **b**}	Enters ethernet uplink fabric interconnect mode for the specified fabric interconnect (A or B).
Step 3	UCS-A# /eth-uplink/fabric # **scope interface** *<slot id>* *<port id>*	Specifies the interface that you are configuring.
Step 4	UCS-A# /eth-uplink/fabric/interface # **set macsec-intf-config-name** *<name>*	Sets the MACsec interface configuration name.
Step 5	UCS-A# /eth-uplink/fabric/interface* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to configure MACsec on an uplink interface:

```
UCS-A# scope eth-uplink
UCS-A# /eth-uplink/fabric # scope fabric a
UCS-A# /eth-uplink/fabric # scope interface 1 1
UCS-A# /eth-uplink/fabric/interface # set macsec-intf-config-name macsec_ifconfig
UCS-A# /eth-uplink/fabric/interface* # commit-buffer
UCS-A# /eth-uplink/fabric/interface #
```

Viewing MACsec on an Uplink Interface

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters ethernet uplink mode.
Step 2	UCS-A# /eth-uplink # **scope fabric** {**a** | **b**}	Enters ethernet uplink fabric interconnect mode for the specified fabric interconnect (A or B).

	Command or Action	Purpose
Step 3	UCS-A# /eth-uplink/fabric# **scope interface** *<name>*	Specifies the interface that you are configuring.
Step 4	UCS-A# /eth-uplink/fabric# **show interface** *<slot-id>* *<port-id>* **detail**	

Example

The following example show how to view MACsec on an uplink interface:

```
UCS-A# scope eth-uplink
UCS-A# /eth-uplink # scope fabric a
UCS-A# /eth-uplink/fabric # scope interface 1 1
UCS-A# /eth-uplink/fabric/interface # show interface detail
Interfaces:
    Slot Id: 1
    Port Id: 2
    User Label:
    Admin State: Enabled
    Oper State: Sfp Not Present
    State Reason: xcvr-absent
    flow control policy: default
    Speed: Auto
    Oper Speed: Auto
    Lic State: License Ok
    Grace Period: 0
    Ethernet Link Profile name: default
    Oper Ethernet Link Profile name: fabric/lan/eth-link-prof-default
    Udld Oper State: Unknown
    MACsec Interface Config name: test-mic
    Licensing Message: Perpetual software license is installed. All ports on this Fabric
Interconnect are licensed
```

Deleting MACsec on an Uplink Interface

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters ethernet uplink mode.
Step 2	UCS-A# /eth-uplink # **scope fabric** {**a** | **b**}	Enters ethernet uplink fabric interconnect mode for the specified fabric interconnect (A or B).
Step 3	UCS-A# /eth-uplink/fabric# **scope interface** *<slot-id>* *<port-id>*	Enters the interface configuration mode.
Step 4	UCS-A# /eth-uplink/fabric/interface # **set macsec-intf-config-name""**	Deletes the MACsec interface configuration name.
Step 5	UCS-A# /eth-uplink/fabric/interface* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to delete a MACsec on an uplink interface:

```
UCS-A# scope eth-uplink
UCS-A# /eth-uplink/fabric # scope fabric a
UCS-A# /eth-uplink/fabric # scope interface 1 1
UCS-A# /eth-uplink/fabric/interface # set macsec-intf-config-name macsec_ifconfig
UCS-A# /eth-uplink/fabric/interface* # commit-buffer
UCS-A# /eth-uplink/fabric/interface #
```

Configuring MACsec on an Uplink Port Channel Member Interface

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters ethernet uplink mode.
Step 2	UCS-A# /eth-uplink # **scope fabric** {**a** | **b**}	Enters ethernet uplink fabric interconnect mode for the specified fabric interconnect (A or B).
Step 3	UCS-A# /eth-uplink/fabric # **create port-channel** <*port-id*>	Creates a port channel.
Step 4	UCS-A# /eth-uplink/fabric/port-channel # **create member-port**<*slot-id*> <*port-id*>	Creates a member port channel.
Step 5	UCS-A# /eth-uplink/fabric/port-channel/member-port* # **set macsec-intf-config-name** <*name*>	Sets the MACsec interface configuration name.
Step 6	UCS-A# /eth-uplink/fabric/port-channel/member-port* # **commit-buffer**	Commits the transaction to the system configuration.

Example

```
UCS-A# scope eth-uplink
UCS-A# /eth-uplink # scope fabric a
UCS-A# /eth-uplink/fabric # create port-channel 1
UCS-A# /eth-uplink/fabric/port-channel # create member-port 1 1
UCS-A# /eth-uplink/fabric/port-channel/member-port* # set macsec-intf-config-name
macsec_ifconfig
UCS-A# /eth-uplink/fabric/port-channel/member-port* # commit-buffer
UCS-A# /eth-uplink/fabric/port-channel/member-port #
```

Viewing MACsec on an Uplink Port Channel Member Interface

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters ethernet uplink mode.
Step 2	UCS-A# /eth-uplink # **scope fabric** {**a** | **b**}	Enters ethernet uplink fabric interconnect mode for the specified fabric interconnect (A or B).
Step 3	UCS-A# /eth-uplink/fabric # **scope port-channel** *<port-id>*	Enters the port channel configuration mode.
Step 4	UCS-A# /eth-uplink/fabric/port-channel # **scope member-port** *<slot-id> <port-id>*	Enters the member port configuration mode.
Step 5	UCS-A# /eth-uplink/fabric/port-channel* # **show detail**	Displays the uplink port channel member interface.

Example

```
UCS-A# scope eth-uplink
UCS-A# /eth-uplink # scope fabric a
UCS-A# /eth-uplink/fabric # scope port-channel 1
UCS-A# /eth-uplink/fabric/port-channel # scope member-port 1 1
UCS-A# /eth-uplink/fabric/port-channel* # show detail
Member Ports:
Slot Id: 1
Port Id: 5
Membership: Down
Oper State: Sfp Not Present
State Reason: xcvr-absent
Lic State: License Ok
Grace Period: 0
Ethernet Link Profile name: default
Oper Ethernet Link Profile name: fabric/lan/eth-link-prof-default
Udld Oper State: Unknown
MACsec Interface Config name: macsec_ifconfig
Licensing Message: Perpetual software license is installed. All ports on this Fabric
Interconnect are licensed
```

Deleting MACsec on an Uplink Port Channel Member Interface

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**	Enters ethernet uplink mode.

	Command or Action	Purpose
Step 2	UCS-A# /eth-uplink # **scope fabric** {**a** | **b**}	Enters ethernet uplink fabric interconnect mode for the specified fabric interconnect (A or B).
Step 3	UCS-A# /eth-uplink/fabric # **scope port-channel** *<name>*	Enters the port channel configuration mode.
Step 4	UCS-A# /eth-uplink/fabric/port-channel # **scope member-port** *<name>*	Enters the member port channel configuration mode.
Step 5	UCS-A# /eth-uplink/fabric/port-channel/member-port* # **set macsec-intf-config-name ""**	Sets the MACsec interface configuration name.
Step 6	UCS-A# /eth-uplink/fabric/port-channel/member-port* # **commit-buffer**	Commits the transaction to the system configuration.

Example

```
UCS-A# scope eth-uplink
UCS-A# /eth-uplink # scope fabric a
UCS-A# /eth-uplink/fabric # scope port-channel 1
UCS-A# /eth-uplink/fabric/port-channel # scope member-port 1 1
UCS-A# /eth-uplink/fabric/port-channel/member-port* # set macsec-intf-config-name ""
UCS-A# /eth-uplink/fabric/port-channel/member-port* # commit-buffer
UCS-A# /eth-uplink/fabric/port-channel/member-port #
```

Configurable EAPOL Destination and Ethernet Type

Configurable EAPOL MAC and Ethernet type provides you the ability to change the MAC address and the Ethernet type of the MKA packet, to allow CE device to form MKA sessions over the ethernet networks that consume the standard MKA packets.

The EAPOL destination Ethernet type can be changed from the default Ethernet type of 0x888E to an alternate value or, the EAPOL destination MAC address can be changed from the default DMAC of 01:80:C2:00:00:03 to an alternate value, to avoid being consumed by a provider bridge.

This feature is available at the interface level and the alternate EAPOL configuration can be changed on any interface at any given time as follows:

- If the MACsec is already configured on an interface, the sessions comes up with a new alternate EAPOL configuration.
- When MACsec is not configured on an interface, the EAPOL configuration is applied to the interface and is effective when MACsec is configured on that interface.

Enabling EAPOL Configuration

You can enable the EAPOL configuration on any available interface.

Before you begin

Ensure that MACsec is enabled.

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope macsec**	Enters the MACsec configuration mode.
Step 2	UCS-A /macsec # **create macsec-eapol** <*name*>	Creates a MACsec EAPOL configuration.
Step 3	UCS-A /macsec/macsec-eapol* # **set macaddress** <*AA:BB:CC:DD:EE:FF*>	Enables the MAC addresses.
Step 4	UCS-A /macsec/macsec-eapol* # **set ethertype** <*0x600-0xffff*>.	Enables the EAPOL configuration on the specified interface type and identity. If the ethernet type is not specified, the default ethernet type of MKA packets, which is 0x888e, is considered.
Step 5	UCS-A /macsec/macsec-eapol* # **exit**	Exits MACsec EAPOL configuration mode.
Step 6	UCS-A /macsec* # **scope macsec-interface-config** <*name*>.	Enters the MACsec interface configuration mode.
Step 7	UCS-A /macsec/macsec-interface-config* # **set eapol-name** <*eapol-name*>	Apply the MACsec EAPOL configuration on an interface.
Step 8	UCS-A# **scope eth-uplink**	Enters Ethernet uplink mode.
Step 9	UCS-A /eth-uplink # **scope fabric** { **a** | **b** }	Enters Ethernet uplink fabric interconnect mode for the specified fabric interconnect (A or B).
Step 10	UCS-A /eth-uplink/fabric # **scope interface** <*slot-id*><*port-id*>	Displays the Ethernet uplink fabric interconnect mode for the specified interface.
Step 11	UCS-A /eth-uplink/fabric/interface # **set macsec-interface-config-name** <*interface name*>	Sets the interface configuration name.
Step 12	UCS-A /eth-uplink/fabric/interface* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example enables a MACsec EAPOL configuration and applies it on an interface.

```
UCS-A# scope macsec
UCS-A /macsec # create macsec-eapol custom-eapol
UCS-A /macsec/macsec-eapol* # set macaddress 65:25:22:22:15:71
UCS-A /macsec/macsec-eapol* # set ethertype 0x888e
```

```
UCS-A /macsec/macsec-eapol* # exit
UCS-A  /macsec* # scope macsec-interface-config <name>
UCS-A /macsec/macsec-interface-config* # set eapol-name <eapol-name>
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric a
UCS-A /eth-uplink/fabric # scope interface 1 4
UCS-A /eth-uplink/fabric/interface # set macsec-intf-config-name macsec-ifconfig
UCS-A /eth-uplink/fabric/interface* # commit-buffer
UCS-A /eth-uplink/fabric/interface #
```

Disabling EAPOL Configuration

Procedure

	Command or Action	Purpose
Step 1	UCS-A# **scope eth-uplink**.	Enters Ethernet uplink mode.
Step 2	UCS-A /eth-uplink # **scope fabric** {**a** | **b**}	Enters Ethernet uplink fabric interconnect mode for the specified fabric interconnect (A or B).
Step 3	UCS-A /eth-uplink/fabric # **set interface** *<slot-id>* *<port-id>*	Sets the interface configuration name.
Step 4	UCS-A /eth-uplink/fabric/interface # **set macsec-intf-config-name** *<interface-name>*	Sets the MACsec interface configuration name.
Step 5	UCS-A /eth-uplink/fabric/interface/macsec-interface-config* # **commit-buffer**	Commits the transaction to the system configuration.

Example

The following example shows how to disable a MACsec EAPOL configuration:

```
UCS-A# scope eth-uplink
UCS-A /eth-uplink # scope fabric a
UCS-A /eth-uplink/fabric # scope interface 1 4
UCS-A /eth-uplink/fabric/interface # set macsec-intf-config-name macsec-ifconfig
UCS-A /eth-uplink/fabric/interface* # commit-buffer
```

Displaying MACsec Sessions

The Operational states of the MACsec session on an interface are displayed as follows:

```
UCS-A /eth-uplink/fabric/interface # show macsec-session

Interface:

MACsec State             MACsec State Reason             MACsec Auth-Mode             MACsec
 Key-Server
------------------------ ------------------------------- ----------------------------
-----------------
```

```
Secured                  Secured MKA Session with MACsec Primary Psk                  No

Interface:

MACsec State              MACsec State Reason              MACsec Auth-Mode
MACsec Key-Server
------------------------ ------------------------------ ----------------------------
-----------------

UCS-A /eth-uplink/fabric/interface # show macsec-session detail
MACsec session:
    MACsec State: Secured
    MACsec State Reason: Secured MKA Session with MACsec
    MACsec Auth-Mode: Primary Psk
    MACsec Key-Server: No
    MACsec Cipher Suite: GCM AES XPN 256
    MACsec Confidentiality Offset: Conf Offset 0

    MACsec State:
    MACsec State Reason:
    MACsec Auth-Mode:
    MACsec Key-Server:
    MACsec Cipher Suite:
    MACsec Confidentiality Offset:
```

The possible values for operational states are as follows:

- MACsec Status—Init, Pending, Secured, Rekeyed
- MACsec Key-server—yes, no
- MACsec Auth-mode—Primary-PSK, Fallback-PSK

The following CLI will have two more additional possible values of **State Reason** to represent the state of interface based on status of the MACsec session configured on it.

```
UCS-A /eth-uplink/fabric/interface # show interface

Interface:

Slot Id    Port Id    Admin State Oper State       Lic State            Grace Period
State Reason Ethernet Link Profile name Oper Ethernet Link Profile name
---------- ---------- ----------- ---------------- -------------------- ---------------
------------ ------------------------- -------------------------------
1          1          Enabled     Link Down        License Ok           0
link-failure default                    fabric/lan/eth-link-prof-default
```

Displaying MACsec Statistics

You can display MACsec statistics using the following commands:

Command	Description
show stats macsec-tx-stats	Displays the MACsec transmitter status.
show stats macsec-rx-stats	Displays the MACsec receiver status.

The following example shows the MACsec security statistics for a specific Ethernet interface.

Note The following differences exist for uncontrolled and controlled packets in Rx and Tx statistics:

Rx statistics:

- Uncontrolled = Encrypted and unencrypted
- Controlled = Decrypted

Tx statistics:

- Uncontrolled = Unencrypted
- Controlled = Encrypted

The following example shows the MACsec statistics:

```
UCS-A /eth-uplink/fabric/interface # show stats ether-macsec-rx-stats

Ether Macsec Rx Stats:
Time Collected: 2024-05-07T15:59:30.243
Monitored Object: sys/switch-A/slot-1/switch-ether/port-8
Suspect: No
Unicast Uncontrolled Packets (packets): 459227
Multicast Uncontrolled Packets (packets): 3648755
Broadcast Uncontrolled Packets (packets): 9494097
Uncontrolled Rx Drop Packets (packets): 0
Uncontrolled Rx Error Packets (packets): 0
Unicast Controlled Packets (packets): 0
Multicast Controlled Packets (packets): 0
Broadcast Controlled Packets (packets): 0
Controlled Rx Drop Packets (packets): 0
Controlled Rx Error Packets (packets): 0
Controlled Packets: 12902005
Thresholded: Unicast Uncontrolled Packets Delta Min


UCS-A /eth-uplink/fabric/interface # show stats ether-macsec-tx-stats

Ether Macsec Tx Stats:
Time Collected: 2024-05-07T15:59:30.243
Monitored Object: sys/switch-A/slot-1/switch-ether/port-8
Suspect: No
Unicast Uncontrolled Packets (packets): 0
Multicast Uncontrolled Packets (packets): 0
Broadcast Uncontrolled Packets (packets): 0
Uncontrolled Rx Drop Packets (packets): 0
Uncontrolled Rx Error Packets (packets): 0
Unicast Controlled Packets (packets): 0
Multicast Controlled Packets (packets): 0
Broadcast Controlled Packets (packets): 0
Controlled Rx Drop Packets (packets): 0
Controlled Rx Error Packets (packets): 0
Controlled Packets: 883044
Thresholded: Unicast Uncontrolled Packets Delta Min
```