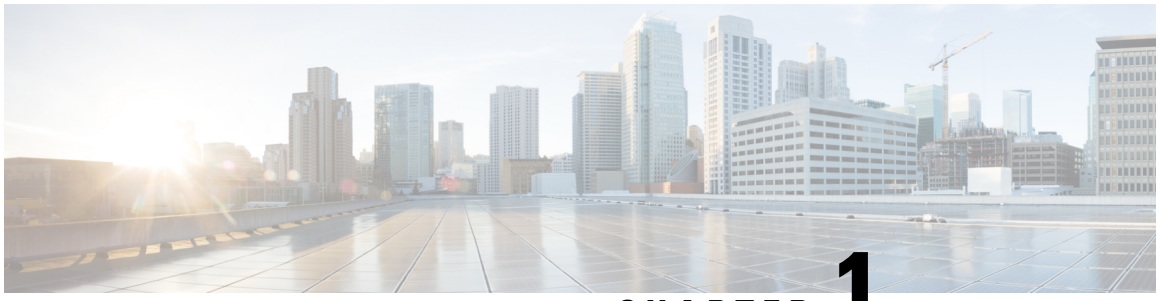
System Management Configuration Guide, Cisco IOS XE 17.14.x (Catalyst 9600 Switches)

First Published: 2024-03-29

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

CONTENTS



CHAPTER 1

Administering the Device

Restriction for Administering the Device

Unicast MAC address filtering is not supported on the Cisco Catalyst 9600 Series Supervisor 2 Module (C9600X-SUP-2).

Information About Administering the Device

The following sections provide information about administering the device:

System Time and Date Management

You can manage the system time and date on your device using automatic configuration methods (RTC and NTP), or manual configuration methods.

Note For complete syntax and usage information for the commands used in this section, see the *Cisco IOS Configuration Fundamentals Command Reference* on *Cisco.com*.

System Clock

The basis of the time service is the system clock. This clock runs from the moment the system starts up and keeps track of the date and time.

The system clock can then be set from these sources:

- RTC

- NTP
- Manual configuration

The system clock can provide time to these services:

- User **show** commands
- Logging and debugging messages

The system clock keeps track of time internally based on Coordinated Universal Time (UTC), also known as Greenwich Mean Time (GMT). You can configure information about the local time zone and summer time (daylight saving time) so that the time appears correctly for the local time zone.

The system clock keeps track of whether the time is *authoritative* or not (that is, whether it has been set by a time source considered to be authoritative). If it is not authoritative, the time is available only for display purposes and is not redistributed.

Network Time Protocol

The NTP is designed to time-synchronize a network of devices. NTP runs over User Datagram Protocol (UDP), which runs over IP. NTP is documented in RFC 1305. The current protocol is version 4 (NTPv4), which is a proposed standard as documented in RFC 5905. It is backward compatible with version 3, specified in RFC 1305.

An NTP network usually gets its time from an authoritative time source, such as a radio clock or an atomic clock attached to a time server. NTP then distributes this time across the network. NTP is extremely efficient; no more than one packet per minute is necessary to synchronize two devices to within a millisecond of one another.

NTP uses the concept of a *stratum* to describe how many NTP hops away a device is from an authoritative time source. A stratum 1 time server has a radio or atomic clock directly attached, a stratum 2 time server receives its time through NTP from a stratum 1 time server, and so on. A device running NTP automatically chooses as its time source the device with the lowest stratum number with which it communicates through NTP. This strategy effectively builds a self-organizing tree of NTP speakers.

NTP avoids synchronizing to a device whose time might not be accurate by never synchronizing to a device that is not synchronized. NTP also compares the time reported by several devices and does not synchronize to a device whose time is significantly different than the others, even if its stratum is lower.

The communications between devices running NTP (known as associations) are usually statically configured; each device is given the IP address of all devices with which it should form associations. Accurate timekeeping is possible by exchanging NTP messages between each pair of devices with an association. However, in a LAN environment, NTP can be configured to use IP broadcast messages instead. This alternative reduces configuration complexity because each device can simply be configured to send or receive broadcast messages. However, in that case, information flow is one-way only.

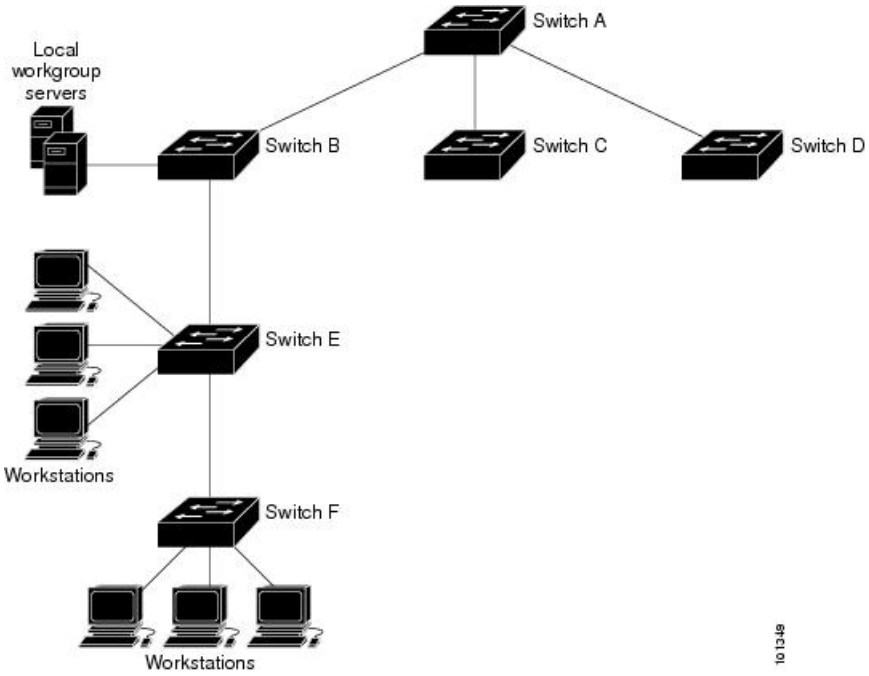
The time kept on a device is a critical resource; you should use the security features of NTP to avoid the accidental or malicious setting of an incorrect time. Two mechanisms are available: an access list-based restriction scheme and an encrypted authentication mechanism.

Cisco’s implementation of NTP does not support stratum 1 service; it is not possible to connect to a radio or atomic clock. We recommend that the time service for your network be derived from the public NTP servers available on the IP Internet.

The Figure shows a typical network example using NTP. Device A is the primary NTP, with the **Device** B, C, and D configured in NTP server mode, in server association with Device A. Device E is configured as an NTP peer to the upstream and downstream device, Device B and Device F, respectively.

Figure 1: NTP Network Configuration

An example of a typical network using NTP



If the network is isolated from the Internet, Cisco's implementation of NTP allows a device to act as if it is synchronized through NTP, when in fact it has learned the time by using other means. Other devices then synchronize to that device through NTP.

When multiple sources of time are available, NTP is always considered to be more authoritative. NTP time overrides the time set by any other method.

Several manufacturers include NTP software for their host systems, and a publicly available version for systems running UNIX and its various derivatives is also available. This software allows host systems to be time-synchronized as well.

NTP Stratum

NTP uses the concept of a *stratum* to describe how many NTP hops away a device is from an authoritative time source. A stratum 1 time server has a radio or atomic clock directly attached, a stratum 2 time server receives its time through NTP from a stratum 1 time server, and so on. A device running NTP automatically chooses as its time source the device with the lowest stratum number with which it communicates through NTP. This strategy effectively builds a self-organizing tree of NTP speakers.

NTP avoids synchronizing to a device whose time might not be accurate by never synchronizing to a device that is not synchronized. NTP also compares the time reported by several devices and does not synchronize to a device whose time is significantly different than the others, even if its stratum is lower.

NTP Associations

The communications between devices running NTP (known as *associations*) are usually statically configured; each device is given the IP address of all devices with which it should form associations. Accurate timekeeping is possible by exchanging NTP messages between each pair of devices with an association. However, in a LAN environment, NTP can be configured to use IP broadcast messages instead. This alternative reduces configuration complexity because each device can simply be configured to send or receive broadcast messages. However, in that case, information flow is one-way only.

Poll-Based NTP Associations

Networking devices running NTP can be configured to operate in variety of association modes when synchronizing time with reference time sources. A networking device can obtain time information on a network in two ways—by polling host servers and by listening to NTP broadcasts. This section focuses on the poll-based association modes. Broadcast-based NTP associations are discussed in the *Broadcast-Based NTP Associations* section.

The following are the two most commonly used poll-based association modes:

- Client mode
- Symmetric active mode

The client and the symmetric active modes should be used when NTP is required to provide a high level of time accuracy and reliability.

When a networking device is operating in the client mode, it polls its assigned time-serving hosts for the current time. The networking device will then pick a host from among all the polled time servers to synchronize with. Because the relationship that is established in this case is a client-host relationship, the host will not capture or use any time information sent by the local client device. This mode is most suited for file-server and workstation clients that are not required to provide any form of time synchronization to other local clients. Use the **ntp server** command to individually specify the time server that you want your networking device to consider synchronizing with and to set your networking device to operate in the client mode.

When a networking device is operating in the symmetric active mode, it polls its assigned time-serving hosts for the current time and it responds to polls by its hosts. Because this is a peer-to-peer relationship, the host will also retain time-related information of the local networking device that it is communicating with. This mode should be used when a number of mutually redundant servers are interconnected via diverse network paths. Most stratum 1 and stratum 2 servers on the Internet adopt this form of network setup. Use the **ntp peer** command to individually specify the time serving hosts that you want your networking device to consider synchronizing with and to set your networking device to operate in the symmetric active mode.

The specific mode that you should set for each of your networking devices depends primarily on the role that you want them to assume as a timekeeping device (server or client) and the device's proximity to a stratum 1 timekeeping server.

A networking device engages in polling when it is operating as a client or a host in the client mode or when it is acting as a peer in the symmetric active mode. Although polling does not usually place a burden on memory and CPU resources such as bandwidth, an exceedingly large number of ongoing and simultaneous polls on a system can seriously impact the performance of a system or slow the performance of a given network. To avoid having an excessive number of ongoing polls on a network, you should limit the number of direct, peer-to-peer or client-to-server associations. Instead, you should consider using NTP broadcasts to propagate time information within a localized network.

Broadcast-Based NTP Associations

Broadcast-based NTP associations should be used when time accuracy and reliability requirements are modest and if your network is localized and has more than 20 clients. Broadcast-based NTP associations are also recommended for use on networks that have limited bandwidth, system memory, or CPU resources.

A networking device operating in the broadcast client mode does not engage in any polling. Instead, it listens for NTP broadcast packets that are transmitted by broadcast time servers. Consequently, time accuracy can be marginally reduced because time information flows only one way.

Use the **ntp broadcast client** command to set your networking device to listen for NTP broadcast packets propagated through a network. For broadcast client mode to work, the broadcast server and its clients must be located on the same subnet. You must enable the time server that transmits NTP broadcast packets on the interface of the given device by using the **ntp broadcast** command.

NTP Security

The time kept on a device is a critical resource; you should use the security features of NTP to avoid the accidental or malicious setting of an incorrect time. Two mechanisms are available: an access list-based restriction scheme and an encrypted authentication mechanism.

Note We do not recommend configuring Message Direct 5 (MD5) authentication. You can use other supported authentication methods for stronger encryption.

NTP Access Group

The access list-based restriction scheme allows you to grant or deny certain access privileges to an entire network, a subnet within a network, or a host within a subnet. To define an NTP access group, use the ntp access-group command in global configuration mode.

The access group options are scanned in the following order, from least restrictive to the most restrictive:

1. ipv4 —Configures IPv4 access lists.
2. ipv6 —Configures IPv6 access lists.
3. peer —Allows time requests and NTP control queries, and allows the system to synchronize itself to a system whose address passes the access list criteria.
4. serve —Allows time requests and NTP control queries, but does not allow the system to synchronize itself to a system whose address passes the access list criteria.
5. serve-only —Allows only time requests from a system whose address passes the access list criteria.
6. query-only —Allows only NTP control queries from a system whose address passes the access list criteria.

If the source IP address matches the access lists for more than one access type, the first type is granted access. If no access groups are specified, all access types are granted access to all systems. If any access groups are specified, only the specified access types will be granted access.

For details on NTP control queries, see RFC 1305.

The encrypted NTP authentication scheme should be used when a reliable form of access control is required. Unlike the access list-based restriction scheme that is based on IP addresses, the encrypted authentication scheme uses authentication keys and an authentication process to determine if NTP synchronization packets

sent by designated peers or servers on a local network are deemed as trusted before the time information that they carry along with them is accepted.

The authentication process begins from the moment an NTP packet is created. Cryptographic checksum keys are generated using the message digest algorithm 5 (MD5) and are embedded into the NTP synchronization packet that is sent to a receiving client. Once a packet is received by a client, its cryptographic checksum key is decrypted and checked against a list of trusted keys. If the packet contains a matching authentication key, the time-stamp information that is contained within the packet is accepted by the receiving client. NTP synchronization packets that do not contain a matching authenticator key are ignored.

Note In large networks, where many trusted keys must be configured, the Range of Trusted Key Configuration feature enables configuring multiple keys simultaneously.

It is important to note that the encryption and decryption processes used in NTP authentication can be very CPU-intensive and can seriously degrade the accuracy of the time that is propagated within a network. If your network setup permits a more comprehensive model of access control, you should consider the use of the access list-based form of control.

After NTP authentication is properly configured, your networking device will synchronize with and provide synchronization only to trusted time sources.

NTP Services on a Specific Interface

Network Time Protocol (NTP) services are disabled on all interfaces by default. NTP is enabled globally when any NTP commands are entered. You can selectively prevent NTP packets from being received through a specific interface by using the **ntp disable** command in interface configuration mode.

Source IP Address for NTP Packets

When the system sends an NTP packet, the source IP address is normally set to the address of the interface through which the NTP packet is sent. Use the **ntp source** *interface* command in global configuration mode to configure a specific interface from which the IP source address will be taken.

This interface will be used for the source address for all packets sent to all destinations. If a source address is to be used for a specific association, use the **source** keyword in the **ntp peer** or **ntp server** command.

NTP Implementation

Implementation of NTP does not support stratum 1 service; it is not possible to connect to a radio or atomic clock. We recommend that the time service for your network be derived from the public NTP servers available on the IP Internet.

If the network is isolated from the Internet, NTP allows a device to act as if it is synchronized through NTP, when in fact it has learned the time by using other means. Other devices then synchronize to that device through NTP.

When multiple sources of time are available, NTP is always considered to be more authoritative. NTP time overrides the time set by any other method.

Several manufacturers include NTP software for their host systems, and a publicly available version for systems running UNIX and its various derivatives is also available. This software allows host systems to be time-synchronized as well.

DNS

The DNS protocol controls the Domain Name System (DNS), a distributed database with which you can map hostnames to IP addresses. When you configure DNS on your device, you can substitute the hostname for the IP address with all IP commands, such as **ping**, **telnet**, **connect**, and related Telnet support operations.

IP defines a hierarchical naming scheme that allows a device to be identified by its location or domain. Domain names are pieced together with periods (.) as the delimiting characters. For example, Cisco Systems is a commercial organization that IP identifies by a *com* domain name, so its domain name is *cisco.com*. A specific device in this domain, for example, the File Transfer Protocol (FTP) system is identified as *ftp.cisco.com*.

To keep track of domain names, IP has defined the concept of a domain name server, which holds a cache (or database) of names mapped to IP addresses. To map domain names to IP addresses, you must first identify the hostnames, specify the name server that is present on your network, and enable the DNS.

Default DNS Settings

Table 1: Default DNS Settings

Feature	Default Setting
DNS enable state	Enabled.
DNS default domain name	None configured.
DNS servers	No name server addresses are configured.

Login Banners

You can configure a message-of-the-day (MOTD) and a login banner. The MOTD banner is displayed on all connected terminals at login and is useful for sending messages that affect all network users (such as impending system shutdowns).

The login banner is also displayed on all connected terminals. It appears after the MOTD banner and before the login prompts.

In default banner configuration, the MOTD and login banners are not configured

Note For complete syntax and usage information for the commands used in this section, see the *Cisco IOS Configuration Fundamentals Command Reference, Release 12.4*.

Default Banner Configuration

The MOTD and login banners are not configured.

MAC Address Table

The MAC address table contains address information that the device uses to forward traffic between ports. All MAC addresses in the address table are associated with one or more ports. The address table includes these types of addresses:

- Dynamic address—A source MAC address that the device learns and then ages when it is not in use.
- Static address—A manually entered unicast address that does not age and that is not lost when the device resets.

The address table lists the destination MAC address, the associated VLAN ID, and port number associated with the address and the type (static or dynamic).

Note For complete syntax and usage information for the commands used in this section, see the command reference for this release.

MAC Address Table Creation

With multiple MAC addresses supported on all ports, you can connect any port on the device to other network devices. The device provides dynamic addressing by learning the source address of packets it receives on each port and adding the address and its associated port number to the address table. As devices are added or removed from the network, the device updates the address table, adding new dynamic addresses and aging out those that are not in use.

The aging interval is globally configured. However, the device maintains an address table for each VLAN, and STP can accelerate the aging interval on a per-VLAN basis.

The device sends packets between any combination of ports, based on the destination address of the received packet. Using the MAC address table, the device forwards the packet only to the port associated with the destination address. If the destination address is on the port that sent the packet, the packet is filtered and not forwarded. The device always uses the store-and-forward method: complete packets are stored and checked for errors before transmission.

MAC Addresses and VLANs

All addresses are associated with a VLAN. An address can exist in more than one VLAN and have different destinations in each. Unicast addresses, for example, could be forwarded to port 1 in VLAN 1 and ports 9, 10, and 1 in VLAN 5.

Each VLAN maintains its own logical address table. A known address in one VLAN is unknown in another until it is learned or statically associated with a port in the other VLAN.

Default MAC Address Table Settings

The following table shows the default settings for the MAC address table.

Table 2: Default Settings for the MAC Address

Feature	Default Setting
Aging time	300 seconds
Dynamic addresses	Automatically learned
Static addresses	None configured

ARP Table Management

To communicate with a device (over Ethernet, for example), the software first must learn the 48-bit MAC address or the local data link address of that device. The process of learning the local data link address from an IP address is called *address resolution*.

The Address Resolution Protocol (ARP) associates a host IP address with the corresponding media or MAC addresses and the VLAN ID. Using an IP address, ARP finds the associated MAC address. When a MAC address is found, the IP-MAC address association is stored in an ARP cache for rapid retrieval. Then the IP datagram is encapsulated in a link-layer frame and sent over the network. Encapsulation of IP datagrams and ARP requests and replies on IEEE 802 networks other than Ethernet is specified by the Subnetwork Access Protocol (SNAP). By default, standard Ethernet-style ARP encapsulation (represented by the **arpa** keyword) is enabled on the IP interface.

ARP entries added manually to the table do not age and must be manually removed.

For CLI procedures, see the Cisco IOS Release 12.4 documentation on *Cisco.com*.

How to Administer the Device

Configuring the Time and Date Manually

System time remains accurate through restarts and reboot, however, you can manually configure the time and date after the system is restarted.

We recommend that you use manual configuration only when necessary. If you have an outside source to which the device can synchronize, you do not need to manually set the system clock.

Setting the System Clock

If you have an outside source on the network that provides time services, such as an NTP server, you do not need to manually set the system clock.

Follow these steps to set the system clock:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	Use one of the following: • **clock set** *hh:mm:ss day month year* • **clock set** *hh:mm:ss month day year* **Example:**	Manually set the system clock using one of these formats: • *hh:mm:ss*—Specifies the time in hours (24-hour format), minutes, and seconds. The time specified is relative to the configured time zone.

	Command or Action	Purpose
	`Device# clock set 13:32:00 23 March 2013`	• *day*—Specifies the day by date in the month. • *month*—Specifies the month by name. • *year*—Specifies the year (no abbreviation).

Configuring the Time Zone

Follow these steps to manually configure the time zone:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**clock timezone** *zone hours-offset* [*minutes-offset*] **Example:** `Device(config)# clock timezone AST -3 30`	Sets the time zone. Internal time is kept in Coordinated Universal Time (UTC), so this command is used only for display purposes and when the time is manually set. • *zone*—Enters the name of the time zone to be displayed when standard time is in effect. The default is UTC. • *hours-offset*—Enters the hours offset from UTC. • (Optional) *minutes-offset*—Enters the minutes offset from UTC. This available where the local time zone is a percentage of an hour different from UTC.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 5	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring Summer Time (Daylight Saving Time)

To configure summer time (daylight saving time) in areas where it starts and ends on a particular day of the week each year, perform this task:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**clock summer-time** *zone* **date** *date month year hh:mm date month year hh:mm* [*offset*]] **Example:** `Device(config)# clock summer-time PDT date 10 March 2013 2:00 3 November 2013 2:00`	Configures summer time to start and end on specified days every year.
Step 4	**clock summer-time** *zone* **recurring** [*week day month hh:mm week day month hh:mm* [*offset*]] **Example:** `Device(config)# clock summer-time PDT recurring 10 March 2013 2:00 3`	Configures summer time to start and end on the specified days every year. All times are relative to the local time zone. The start time is relative to standard time. The end time is relative to summer time. Summer time is disabled by default. If you

	Command or Action	Purpose
	`November 2013 2:00`	specify **clock summer-time** *zone* **recurring** without parameters, the summer time rules default to the United States rules. If the starting month is after the ending month, the system assumes that you are in the southern hemisphere. • *zone*—Specifies the name of the time zone (for example, PDT) to be displayed when summer time is in effect. • (Optional) *week*— Specifies the week of the month (1 to 4, **first**, or **last**). • (Optional) *day*—Specifies the day of the week (Sunday, Monday...). • (Optional) *month*—Specifies the month (January, February...). • (Optional) *hh:mm*—Specifies the time (24-hour format) in hours and minutes. • (Optional) *offset*—Specifies the number of minutes to add during summer time. The default is 60.
Step 5	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 6	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring NTP

The device does not have a hardware-supported clock and cannot function as an NTP primary clock to which peers synchronize themselves when an external NTP source is not available. The device also has no hardware

support for a calendar. As a result, the **ntp update-calendar** and the **ntp master** commands in global configuration mode are not available.

These following sections provide configuration information on NTP:

Default NTP Configuration

shows the default NTP configuration.

Table 3: Default NTP Configuration

Feature	Default Setting
NTP authentication	Disabled. No authentication key is specified.
NTP peer or server associations	None configured.
NTP broadcast service	Disabled; no interface sends or receives NTP broadcast packets.
NTP access restrictions	No access control is specified.
NTP packet source IP address	The source address is set by the outgoing interface.

NTP is enabled on all interfaces by default. All interfaces receive NTP packets.

Configuring NTP Authentication

To configure NTP authentication, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	[**no**] **ntp authenticate** **Example:** `Device(config)# ntp authenticate`	Enables NTP authentication. Use the **no** form of this command to disable NTP authentication

	Command or Action	Purpose

Step 4	[**no**] **ntp authentication-key** *number* {**md5** | **cmac-aes-128** | **hmac-sha1** | **hmac-sha2-256**} *value*
	Example:
	`Device(config)# ntp authentication-key 42 md5 aNiceKey`
		Defines the authentication keys.
		• Each key has a key number, a type, and a value.
		• Keys can be one of the following types:
		  • **md5**: Authentication using the MD5 algorithm.
		  • **cmac-aes-128**: Authentication using Cipher-based message authentication codes (CMAC) with the AES-128 algorithm. The digest length is 128 bits and the key length is 16 or 32 bytes.
		  • **hmac-sha1**: Authentication using Hash-based Message Authentication Code (HMAC) using the SHA1 hash function. The digest length is 128 bits and the key length is 1 to 32 bytes.
		  • **hmac-sha2-256**: Authentication using HMAC using the SHA2 hash function. The digest length is 256 bits and the key length is 1 to 32 bytes
		Use the **no** form of this command to remove authentication key.

Step 5	[**no**] **ntp trusted-key** *key-number* **Example:** `Device(config)# ntp trusted-key 42`	Defines trusted authentication keys that a peer NTP device must provide in its NTP packets for this device to synchronize to it. Use the **no** form of this command to disable trusted authentication.
Step 6	[**no**] **ntp server** *ip-address* **key** *key-id* [**prefer**] **Example:** `Device(config)# ntp server 172.16.22.44 key 42`	Allows the software clock to be synchronized by an NTP time server. • *ip-address*: The IP address of the time server providing the clock synchronization. • *key-id*: Authentication key defined with the **ntp authentication-key** command. • **prefer**: Sets this peer as the preferred one that provides synchronization. This keyword reduces clock hop among peers. Use the **no** form of this command to remove a server association.

	Command or Action	Purpose
Step 7	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Configuring Poll-Based NTP Associations

To configure poll-based NTP associations, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	[**no**] **ntp peer** *ip-address* [**version** *number*] [**key** *key-id*] [**source** *interface*] [**prefer**] **Example:** `Device(config)# ntp peer 172.16.22.44 version 2`	Configures the device system clock to synchronize a peer or to be synchronized by a peer (peer association). • *ip-address*: The IP address of the peer providing or being provided, the clock synchronization. • *number*: NTP version number. The range is 1 to 4. By default, version 4 is selected. • *key-id*: Authentication key defined with the **ntp authentication-key** command. • *interface*: The interface from which to pick the IP source address. By default, the source IP address is taken from the outgoing interface. • **prefer**: Sets this peer as the preferred one that provides synchronization. This keyword reduces switching back and forth between peers.

	Command or Action	Purpose
		Use the **no** form of this command to remove a peer association.
Step 4	[**no**] **ntp server** *ip-address* [**version** *number*] [**key** *key-id*] [**source** *interface*] [**prefer**] **Example:** `Device(config)# ntp server 172.16.22.44 version 2`	Configures the device's system clock to be synchronized by a time server (server association). • *ip-address*: The IP address of the time server providing the clock synchronization. • *number*: NTP version number. The range is 1 to 4. By default, version 4 is selected. • *key-id*: Authentication key defined with the **ntp authentication-key** command. • *interface*: The interface from which to pick the IP source address. By default, the source IP address is taken from the outgoing interface. • **prefer**: Sets this peer as the preferred one that provides synchronization. This keyword reduces clock hop among peers. Use the **no** form of this command to remove a server association.
Step 5	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Configuring Broadcast-Based NTP Associations

To configure broadcast-based NTP associations, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet1/0/1`	Configures an interface and enters interface configuration mode.
Step 4	[**no**] **ntp broadcast** [**version** *number*] [**key** *key-id*] [*destination-address*] **Example:** `Device(config-if)# ntp broadcast version 2`	Enables the interface to send NTP broadcast packets to a peer. • *number*: NTP version number. The range is 1 to 4. By default, version 4 is used. • *key-id*: Authentication key. • *destination-address*: IP address of the peer that is synchronizing its clock to this switch. Use the **no** form of this command to disable the interface from sending NTP broadcast packets.
Step 5	[**no**] **ntp broadcast client** **Example:** `Device(config-if)# ntp broadcast client`	Enables the interface to receive NTP broadcast packets. Use the **no** form of this command to disable the interface from receiving NTP broadcast packets.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Returns to privileged EXEC mode.
Step 7	[**no**] **ntp broadcastdelay** *microseconds* **Example:** `Device(config)# ntp broadcastdelay 100`	(Optional) Change the estimated round-trip delay between the device and the NTP broadcast server The default is 3000 microseconds. The range is from 1 to 999999. Use the **no** form of this command to disable the interface from receiving NTP broadcast packets.
Step 8	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Configuring NTP Access Restrictions

You can control NTP access on two levels as described in these sections:

Creating an Access Group and Assigning a Basic IP Access List

To create an access group and assign a basic IP access list, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	[**no**] **ntp access-group** {**query-only** | **serve-only** | **serve** | **peer**} *access-list-number* **Example:** `Device(config)# ntp access-group peer 99`	Create an access group, and apply a basic IP access list.. • **query-only**: NTP control queries. • **serve-only**: Time requests. • **serve**: Allows time requests and NTP control queries, but does not allow the device to synchronize to the remote device. • **peer**: Allows time requests and NTP control queries and allows the device to synchronize to the remote device. • *access-list-number*: IP access list number. The range is from 1 to 99. Use the **no** form of this command to remove access control to the switch NTP services.
Step 4	**access-list** *access-list-number* **permit** *source* [*source-wildcard*] **Example:** `Device(config)# access-list 99 permit 172.20.130.5`	Create the access list. • *access-list-number*: IP access list number. The range is from 1 to 99. • **permit**: Permits access if the conditions are matched. • *source*: IP address of the device that is permitted access to the device.

	Command or Action	Purpose
		• *source-wildcard*: Wildcard bits to be applied to the source. **Note** When creating an access list, remember that, by default, the end of the access list contains an implicit deny statement for everything if it did not find a match before reaching the end. Use the **no** form of this command to remove authentication key.
Step 5	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Disabling NTP Services on a Specific Interface

To disable NTP packets from being received on an interface, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-id* **Example:** `Device(config)# interface gigabitethernet1/0/1`	Enters global configuration mode.
Step 4	[**no**] **ntp disable** **Example:**	Disables NTP packets from being received on the interface. Use the **no** form of this command to re-enable receipt of NTP packets on an interface.

	Command or Action	Purpose
	`Device(config-if)# ntp disable`	
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Configuring a System Name

Follow these steps to manually configure a system name:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**hostname** *name* **Example:** `Device(config)# hostname remote-users`	Configures a system name. When you set the system name, it is also used as the system prompt. The default setting is Switch. The name must follow the rules for ARPANET hostnames. They must start with a letter, end with a letter or digit, and have as interior characters only letters, digits, and hyphens. Names can be up to 63 characters.

Step 4	**end**
	Example:
	`remote-users(config)#end`
	`remote-users#`
		Returns to priviliged EXEC mode.

Step 5	**show running-config** **Example:**	Verifies your entries.

	Command or Action	Purpose
	`Device# show running-config`	
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Setting Up DNS

If you use the device IP address as its hostname, the IP address is used and no DNS query occurs. If you configure a hostname that contains no periods (.), a period followed by the default domain name is appended to the hostname before the DNS query is made to map the name to an IP address. The default domain name is the value set by the **ip domain name** command in global configuration mode. If there is a period (.) in the hostname, the Cisco IOS software looks up the IP address without appending any default domain name to the hostname.

Follow these steps to set up your switch to use the DNS:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip domain name** *name* **Example:** `Device(config)# ip domain name Cisco.com`	Defines a default domain name that the software uses to complete unqualified hostnames (names without a dotted-decimal domain name). Do not include the initial period that separates an unqualified name from the domain name. At boot time, no domain name is configured; however, if the device configuration comes from a BOOTP or Dynamic Host Configuration Protocol (DHCP) server, then the default domain name might be set by the BOOTP or

	Command or Action	Purpose
		DHCP server (if the servers were configured with this information).
Step 4	**ip name-server** *server-address1* [*server-address2 ... server-address6*] **Example:** `Device(config)# ip name-server 192.168.1.100 192.168.1.200 192.168.1.300`	Specifies the address of one or more name servers to use for name and address resolution. You can specify up to six name servers. Separate each server address with a space. The first server specified is the primary server. The device sends DNS queries to the primary server first. If that query fails, the backup servers are queried.
Step 5	**ip domain lookup** [**nsap** | **source-interface** *interface*] **Example:** `Device(config)# ip domain-lookup`	(Optional) Enables DNS-based hostname-to-address translation on your device. This feature is enabled by default. If your network devices require connectivity with devices in networks for which you do not control name assignment, you can dynamically assign device names that uniquely identify your devices by using the global Internet naming scheme (DNS).
Step 6	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 7	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 8	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring a Message-of-the-Day Login Banner

You can create a single or multiline message banner that appears on the screen when someone logs in to the device.

Follow these steps to configure a MOTD login banner:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**banner motd** *c message c*
	Example:
	`Device(config)# banner motd #`
	`This is a secure site. Only`
	`authorized users are allowed.`
	`For access, contact technical`
	`support.`
	`#`
		Specifies the message of the day.
		c—Enters the delimiting character of your choice, for example, a pound sign (#), and press the **Return** key. The delimiting character signifies the beginning and end of the banner text. Characters after the ending delimiter are discarded.
		message—Enters a banner message up to 255 characters. You cannot use the delimiting character in the message.

Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 5	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring a Login Banner

You can configure a login banner to be displayed on all connected terminals. This banner appears after the MOTD banner and before the login prompt.

Follow these steps to configure a login banner:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**banner login** *c message c*
	Example:
	`Device(config)# banner login $`
	`Access for authorized users only.`
	`Please enter your username and password.`
	`$`
		Specifies the login message.
		c— Enters the delimiting character of your choice, for example, a pound sign (#), and press the **Return** key. The delimiting character signifies the beginning and end of the banner text. Characters after the ending delimiter are discarded.
		message—Enters a login message up to 255 characters. You cannot use the delimiting character in the message.

Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 5	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Managing the MAC Address Table

Changing the Address Aging Time

Note This is not supported on Cisco Catalyst 9600 Series Supervisor 2 Module (C9600-SUP-2).

Follow these steps to configure the dynamic address table aging time:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mac address-table aging-time** [*0* | *10-1000000*] [**routed-mac** | **vlan** *vlan-id*] **Example:** `Device(config)# mac address-table aging-time 500 vlan 2`	Sets the length of time that a dynamic entry remains in the MAC address table after the entry is used or updated. The range is 10 to 1000000 seconds. The default is 300. You can also enter 0, which disables aging. Static address entries are never aged or removed from the table. *vlan-id*—Valid IDs are 1 to 4094.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 5	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 6	**copy running-config startup-config** **Example:**	(Optional) Saves your entries in the configuration file.

	Command or Action	Purpose
	`Device# copy running-config startup-config`	

Configuring MAC Address Change Notification Traps

Follow these steps to configure the switch to send MAC address change notification traps to an NMS host:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server host** *host-addr community-string notification-type* { **informs** | **traps** } {**version** {**1** | **2c** | **3**}} {**vrf** *vrf instance name*} **Example:** `Device(config)# snmp-server host 172.20.10.10 traps private mac-notification`	Specifies the recipient of the trap message. • *host-addr*—Specifies the name or address of the NMS. • **traps** (the default)—Sends SNMP traps to the host. • **informs**—Sends SNMP informs to the host. • **version**—Specifies the SNMP version to support. Version 1, the default, is not available with informs. • *community-string*—Specifies the string to send with the notification operation. Though you can set this string by using the **snmp-server host** command, we recommend that you define this string by using the **snmp-server community** command before using the **snmp-server host** command. • *notification-type*—Uses the **mac-notification** keyword.

	Command or Action	Purpose
		• **vrf** *vrf instance name*—Specifies the VPN routing/forwarding instance for this host.
Step 4	**snmp-server enable traps mac-notification change** **Example:** `Device(config)# snmp-server enable traps mac-notification change`	Enables the device to send MAC address change notification traps to the NMS.
Step 5	**mac address-table notification change** **Example:** `Device(config)# mac address-table notification change`	Enables the MAC address change notification feature.

Step 6	**mac address-table notification change** [**interval** *value*] [**history-size** *value*]
	Example:
	`Device(config)# mac address-table notification change interval 123`
	`Device(config)#mac address-table notification change history-size 100`
		Enters the trap interval time and the history table size.
		• (Optional) **interval** *value*—Specifies the notification trap interval in seconds between each set of traps that are generated to the NMS. The range is 0 to 2147483647 seconds; the default is 1 second.
		• (Optional) **history-size** *value*—Specifies the maximum number of entries in the MAC notification history table. The range is 0 to 500; the default is 1.

Step 7	**interface** *interface-id* **Example:** `Device(config)# interface fortygigabitethernet1/0/2`	Enters interface configuration mode, and specifies the Layer 2 interface on which to enable the SNMP MAC address notification trap.
Step 8	**snmp trap mac-notification change** {**added** | **removed**} **Example:** `Device(config-if)# snmp trap mac-notification change added`	Enables the MAC address change notification trap on the interface. • Enables the trap when a MAC address is **added** on this interface. • Enables the trap when a MAC address is **removed** from this interface.
Step 9	**end** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config)# end`	
Step 10	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 11	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring MAC Address Move Notification Traps

When you configure MAC-move notification, an SNMP notification is generated and sent to the network management system whenever a MAC address moves from one port to another within the same VLAN.

Follow these steps to configure the device to send MAC address-move notification traps to an NMS host:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server host** *host-addr* {**traps** | **informs**} {**version** {**1** | **2c** | **3**}} *community-string notification-type* **Example:** `Device(config)# snmp-server host 172.20.10.10 traps private mac-notification`	Specifies the recipient of the trap message. • *host-addr*—Specifies the name or address of the NMS. • **traps** (the default)—Sends SNMP traps to the host. • **informs**—Sends SNMP informs to the host.

	Command or Action	Purpose
		• **version**—Specifies the SNMP version to support. Version 1, the default, is not available with informs. • *community-string*—Specifies the string to send with the notification operation. Though you can set this string by using the **snmp-server host** command, we recommend that you define this string by using the **snmp-server community** command before using the **snmp-server host** command. • *notification-type*—Uses the **mac-notification** keyword.
Step 4	**snmp-server enable traps mac-notification move** **Example:** `Device(config)# snmp-server enable traps mac-notification move`	Enables the device to send MAC address move notification traps to the NMS.
Step 5	**mac address-table notification mac-move** **Example:** `Device(config)# mac address-table notification mac-move`	Enables the MAC address move notification feature.
Step 6	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 7	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 8	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

What to do next

To disable MAC address-move notification traps, use the **no snmp-server enable traps mac-notification move** global configuration command. To disable the MAC address-move notification feature, use the **no mac address-table notification mac-move** global configuration command.

You can verify your settings by entering the **show mac address-table notification mac-move** privileged EXEC commands.

Configuring MAC Threshold Notification Traps

When you configure MAC threshold notification, an SNMP notification is generated and sent to the network management system when a MAC address table threshold limit is reached or exceeded.

Follow these steps to configure the switch to send MAC address table threshold notification traps to an NMS host:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server host** *host-addr* {**traps** | **informs**} {**version** {**1** | **2c** | **3**}} *community-string notification-type* **Example:** `Device(config)# snmp-server host 172.20.10.10 traps private mac-notification`	Specifies the recipient of the trap message. • *host-addr*—Specifies the name or address of the NMS. • **traps** (the default)—Sends SNMP traps to the host. • **informs**—Sends SNMP informs to the host. • **version**—Specifies the SNMP version to support. Version 1, the default, is not available with informs. • *community-string*—Specifies the string to send with the notification operation. You can set this string by using the **snmp-server host** command, but we recommend that you define this string by using the **snmp-server community**

	Command or Action	Purpose
		command before using the **snmp-server host** command. • *notification-type*—Uses the **mac-notification** keyword.
Step 4	**snmp-server enable traps mac-notification threshold** **Example:** `Device(config)# snmp-server enable traps mac-notification threshold`	Enables MAC threshold notification traps to the NMS.
Step 5	**mac address-table notification threshold** **Example:** `Device(config)# mac address-table notification threshold`	Enables the MAC address threshold notification feature.

Step 6	**mac address-table notification threshold** [**limit** *percentage*] | [**interval** *time*]
	Example:
	`Device(config)# mac address-table notification threshold interval 123`
	`Device(config)# mac address-table notification threshold limit 78`
		Enters the threshold value for the MAC address threshold usage monitoring.
		• (Optional) **limit** *percentage*—Specifies the percentage of the MAC address table use; valid values are from 1 to 100 percent. The default is 50 percent.
		• (Optional) **interval** *time*—Specifies the time between notifications; valid values are greater than or equal to 120 seconds. The default is 120 seconds.

Step 7	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 8	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 9	**copy running-config startup-config** **Example:**	(Optional) Saves your entries in the configuration file.

	Command or Action	Purpose
	`Device# copy running-config startup-config`	

Disabling MAC Address Learning on VLAN

You can control MAC address learning on a VLAN to manage the available MAC address table space by controlling which VLANs can learn MAC addresses. Before you disable MAC address learning, be sure that you are familiar with the network topology. Disabling MAC address learning on VLAN could cause flooding in the network.

Beginning in privileged EXEC mode, follow these steps to disable MAC address learning on a VLAN:

Before you begin

Follow these guidelines when disabling MAC address learning on a VLAN:

- Use caution before disabling MAC address learning on a VLAN with a configured switch virtual interface (SVI). The switch then floods all IP packets in the Layer 2 domain.
- You can disable MAC address learning on a single VLAN ID from 2 - 4093 (for example, no mac address-table learning vlan 223) or a range of VLAN IDs, separated by a hyphen or comma (for example, no mac address-table learning vlan 1-10, 15).
- It is recommended that you disable MAC address learning only in VLANs with two ports. If you disable MAC address learning on a VLAN with more than two ports, every packet entering the switch is flooded in that VLAN domain.
- If you disable MAC address learning on a VLAN that includes a secure port, MAC address learning is not disabled on that port.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters the global configuration mode.
Step 2	**no mac-address-table learning vlan**[*vlan-id* |,*vlan-id*	*-vlan-id*,] **Example:** `Device(config)# no mac-address-table learning {vlan vlan-id [,vlan-id	-vlan-id]`	Disable MAC address learning on a specified VLAN or VLANs. You can specify a single VLAN ID or a range of VLAN IDs separated by a hyphen or comma. Valid VLAN IDs range from 2 - 4093. It cannot be an internal VLAN.
Step 3	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 4	**show mac-address-table learning vlan**[*vlan-id*] **Example:** `Device# show mac-address-table learning [vlan vlan-id]`	Verify the configuration. You can display the MAC address learning status of all VLANs or a specified VLAN by entering the show mac-address-table learning [vlan vlan-id] privileged EXEC command.
Step 5	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Save your entries in the configuration file.
Step 6	**default mac address-table learning** **Example:** `Device# default mac address-table`	(Optional) Reenable MAC address learning on VLAN in a global configuration mode.

Adding and Removing Static Address Entries

Follow these steps to add a static address:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mac address-table static** *mac-addr* **vlan** *vlan-id* **interface** *interface-id* **Example:** `Device(config)# mac address-table static c2f3.220a.12f4 vlan 4 interface fortygigabitethernet 1/0/1`	Adds a static address to the MAC address table. • *mac-addr*—Specifies the destination MAC unicast address to add to the address table. Packets with this destination address received in the specified VLAN are forwarded to the specified interface. • *vlan-id*—Specifies the VLAN for which the packet with the specified MAC address is received. Valid VLAN IDs are 1 to 4094.

	Command or Action	Purpose
		• *interface-id*—Specifies the interface to which the received packet is forwarded. Valid interfaces include physical ports or port channels. For static multicast addresses, you can enter multiple interface IDs. For static unicast addresses, you can enter only one interface at a time, but you can enter the command multiple times with the same MAC address and VLAN ID.
Step 4	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 5	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring Unicast MAC Address Filtering

Follow these steps to configure the device to drop a source or destination unicast static address:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mac address-table static** *mac-addr* **vlan** *vlan-id* **drop** **Example:** `Device(config)# mac address-table static c2f3.220a.12f4 vlan 4 drop`	Enables unicast MAC address filtering and configure the device to drop a packet with the specified source or destination unicast static address. • *mac-addr*—Specifies a source or destination unicast MAC address (48-bit). Packets with this MAC address are dropped.

	Command or Action	Purpose
		• *vlan-id*—Specifies the VLAN for which the packet with the specified MAC address is received. Valid VLAN IDs are 1 to 4094.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 5	**show running-config** **Example:** `Device# show running-config`	Verifies your entries.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Monitoring and Maintaining Administration of the Device

Command	Purpose
clear mac address-table dynamic	Removes all dynamic entries.
clear mac address-table dynamic address *mac-address*	Removes a specific MAC address.
clear mac address-table dynamic interface *interface-id*	Removes all addresses on the specified physical port or port channel.
clear mac address-table dynamic vlan *vlan-id*	Removes all addresses on a specified VLAN.
show clock [*detail*]	Displays the time and date configuration.
show ip igmp snooping groups	Displays the Layer 2 multicast entries for all VLANs or the specified VLAN.
show mac address-table address *mac-address*	Displays MAC address table information for the specified MAC address.
show mac address-table aging-time	Displays the aging time in all VLANs or the specified VLAN.
show mac address-table count	Displays the number of addresses present in all VLANs or the specified VLAN.
show mac address-table dynamic	Displays only dynamic MAC address table entries.

Command	Purpose
show mac address-table interface *interface-name*	Displays the MAC address table information for the specified interface.
show mac address-table move update	Displays the MAC address table move update information.
show mac address-table multicast	Displays a list of multicast MAC addresses.
show mac address-table notification {change | mac-move | threshold}	Displays the MAC notification parameters and history table.
show mac address-table secure	Displays the secure MAC addresses.
show mac address-table static	Displays only static MAC address table entries.
show mac address-table vlan *vlan-id*	Displays the MAC address table information for the specified VLAN.

Configuration Examples for Device Administration

Example: Setting the System Clock

This example shows how to manually set the system clock:

```
Device# clock set 13:32:00 23 July 2013
```

Examples: Configuring Summer Time

This example (for daylight savings time) shows how to specify that summer time starts on March 10 at 02:00 and ends on November 3 at 02:00:

```
Device(config)# clock summer-time PDT recurring PST date
10 March 2013 2:00 3 November 2013 2:00
```

This example shows how to set summer time start and end dates:

```
Device(config)#clock summer-time PST date
20 March 2013 2:00 20 November 2013 2:00
```

Example: Configuring a MOTD Banner

This example shows how to configure a MOTD banner by using the pound sign (#) symbol as the beginning and ending delimiter:

```
Device(config)# banner motd #

This is a secure site. Only authorized users are allowed.
For access, contact technical support.

#

Device(config)#
```

This example shows the banner that appears from the previous configuration:

```
Unix> telnet 192.0.2.15

Trying 192.0.2.15...

Connected to 192.0.2.15.

Escape character is '^]'.

This is a secure site. Only authorized users are allowed.

For access, contact technical support.

User Access Verification

Password:
```

Example: Configuring a Login Banner

This example shows how to configure a login banner by using the dollar sign ($) symbol as the beginning and ending delimiter:

```
Device(config)# banner login $

Access for authorized users only. Please enter your username and password.

$

Device(config)#
```

Example: Configuring MAC Address Change Notification Traps

This example shows how to specify 172.20.10.10 as the NMS, enable MAC address notification traps to the NMS, enable the MAC address-change notification feature, set the interval time to 123 seconds, set the history-size to 100 entries, and enable traps whenever a MAC address is added on the specified port:

```
Device(config)# snmp-server host 172.20.10.10 traps private mac-notification
Device(config)# snmp-server enable traps mac-notification change
Device(config)# mac address-table notification change
Device(config)# mac address-table notification change interval 123
Device(config)# mac address-table notification change history-size 100
Device(config)# interface fortygigabitethernet1/0/1
Device(config-if)# snmp trap mac-notification change added
```

Example: Configuring MAC Threshold Notification Traps

This example shows how to specify 172.20.10.10 as the NMS, enable the MAC address threshold notification feature, set the interval time to 123 seconds, and set the limit to 78 per cent:

```
Device(config)# snmp-server host 172.20.10.10 traps private mac-notification
Device(config)# snmp-server enable traps mac-notification threshold
Device(config)# mac address-table notification threshold
Device(config)# mac address-table notification threshold interval 123
Device(config)# mac address-table notification threshold limit 78
```

Example: Adding the Static Address to the MAC Address Table

This example shows how to add the static address c2f3.220a.12f4 to the MAC address table. When a packet is received in VLAN 4 with this MAC address as its destination address, the packet is forwarded to the specified port:

Note You cannot associate the same static MAC address to multiple interfaces. If the command is executed again with a different interface, the static MAC address is overwritten on the new interface.

```
Device(config)# mac address-table static c2f3.220a.12f4 vlan 4 interface
fortygigabitethernet1/0/1
```

Example: Configuring Unicast MAC Address Filtering

This example shows how to enable unicast MAC address filtering and how to configure drop packets that have a source or destination address of c2f3.220a.12f4. When a packet is received in VLAN 4 with this MAC address as its source or destination, the packet is dropped:

```
Device(config)# mac address-table static c2f3.220a.12f4 vlan 4 drop
```

Additional References for Device Administration

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	*Command Reference (Catalyst 9600 Series Switches)*

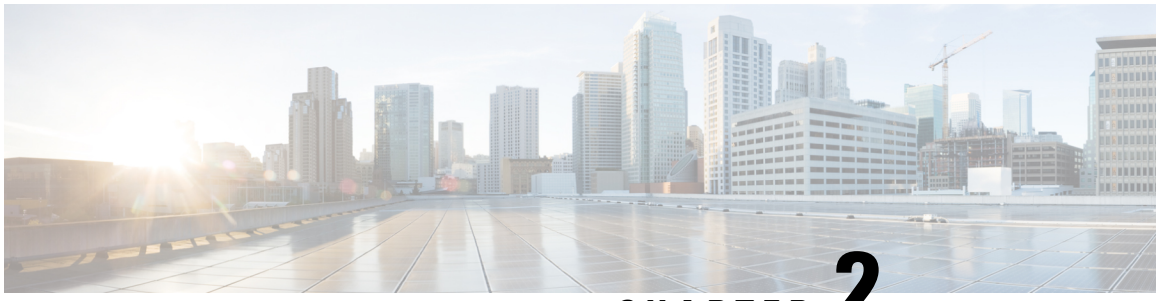
Feature History for Device Administration

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.11.1	Device Administration	The device administration allows to configure the system time and date, system name, a login banner, and set up the DNS.
Cisco IOS XE Cupertino 17.7.1	Device Administration	Support for this feature was introduced on the Cisco Catalyst 9600 Series Supervisor 2 Module (C9600X-SUP-2).

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.



CHAPTER 2

Boot Integrity Visibility

Information About Boot Integrity Visibility

Boot Integrity Visibility allows Cisco's platform identity and software integrity information to be visible and actionable. Platform identity provides the platform's manufacturing installed identity. Software integrity exposes boot integrity measurements that can be used to assess whether the platform has booted trusted code.

During the boot process, the software creates a checksum record of each stage of the bootloader activities.

You can retrieve this record and compare it with a Cisco-certified record to verify if your software image is genuine. If the checksum values do not match, you may be running a software image that is either not certified by Cisco or has been altered by an unauthorized party.

Image Signing and Bootup

The Cisco build servers generate the Cisco IOS XE images. Cisco IOS XE images use the Abraxas image signing system to sign these images securely with the Cisco private RSA keys.

When you copy the Cisco IOS XE image onto a Catalyst 9000 Series Switch, Cisco's ROMMON Boot ROM verifies the image using Cisco release keys. These keys are public keys that correspond to the Cisco release private key that is stored securely on the Abraxas servers. The release key is stored in the ROMMON.

Catalyst 9000 Series Switches support boot integrity visibility feature. Boot integrity visibility serves as a hardware trust anchor which validates the ROMMON software to ensure that the ROMMON software is not tampered with.

The Cisco IOS XE image is digitally signed during the build time. An SHA-512 hash is generated over the entire binary image file, and then the hash is encrypted with a Cisco RSA 2048-bit private key. The ROMMON verifies the signature using the Cisco public key. If the software is not generated by a Cisco build system, the signature verification fails. The device ROMMON rejects the image and stops booting. If the signature verification is successfully, the device boots the image to the Cisco IOS XE runtime environment.

The ROMMON follows these steps when it verifies a signed Cisco IOS XE image during the bootup:

1. Loads the Cisco IOS XE image into the CPU memory.
2. Examines the Cisco IOS XE package header.
3. Runs a non-secure integrity check on the image to ensure that there is no unintentional file corruption from the disk or TFTP. This is performed using a non-secure SHA-1 hash.
4. Copies the Cisco's RSA 2048-bit public release key from the ROMMON storage and validates that the Cisco's RSA 2048-bit public release key is not tampered.
5. Extracts the Code Signing signature (SHA-512 hash) from the package header and verifies it using Cisco's RSA 2048-bit public release key.
6. Performs the Code Signing validation by calculating the SHA-512 hash of the Cisco IOS XE package and compares it with the Code Signing signature. The Signed package is now validated.
7. Examines the Cisco IOS XE package header to validate the platform type and CPU architecture for compatibility.
8. Extracts the Cisco IOS XE software from the Cisco IOS XE package and boots it.

Note In above process, step 3 is a non-secure check of the image which is intended to confirm the image against inadvertent corruption due to disk errors, file transfer errors, or copying errors. This is not part of the image code signing. This check is not intended to detect deliberate image tampering.

Image Code Signing validation occurs in step 4, 5, and 6. This is a secure code signing check of the image using an SHA-512 hash that is encrypted with a 2048-bit RSA key. This check is intended to detect deliberate image tampering.

Verifying the Software Image and Hardware

This task describes how to retrieve the checksum record that was created during a switch bootup. Enter the following commands in privileged EXEC mode.

Note On executing the following commands, you might see the message **% Please Try After Few Seconds** displayed on the CLI. This does not indicate a CLI failure, but indicates setting up of underlying infrastructure required to get the required output. We recommend waiting for a few minutes and then try the command again.

The messages **% Error retrieving SUDI certificate** and **% Error retrieving integrity data** signify a real CLI failure.

Procedure

	Command or Action	Purpose
Step 1	**show platform sudi certificate** [**sign** [**nonce** *nonce*]]	Displays checksum record for the specific SUDI.

	Command or Action	Purpose
	Example: `Device# show platform sudi certificate sign nonce 123`	• (Optional) **sign** - Show signature • (Optional) **nonce** - Enter a nonce value
Step 2	**show platform integrity** [**sign** [**nonce** *nonce*]] **Example:** `Device# show platform integrity sign nonce 123`	Displays checksum record for boot stages. • (Optional) **sign** - Show signature • (Optional) **nonce** - Enter a nonce value

Verifying Platform Identity and Software Integrity

Verifying Platform Identity

The following example displays the Secure Unique Device Identity (SUDI) chain in PEM format. Encoded into the SUDI is the Product ID and Serial Number of each individual device such that the device can be uniquely identified on a network of thousands of devices. The first certificate is the Cisco Root CA 2048 and the second is the Cisco subordinate CA (ACT2 SUDI CA). Both certificates can be verified to match those published on https://www.cisco.com/security/pki/. The third is the SUDI certificate.

Important All the CLI outputs provided here are intended only for reference. The output differs based the configuration of the device.

```
Device# show platform sudi certificate sign nonce 123
-----BEGIN CERTIFICATE-----
MIIDQzCCAiugAwIBAgIQX/h7KCtU3I1CoxW1aMmt/zANBgkqhkiG9w0BAQUFADA1
MRYwFAYDVQQKEw1DaXNjbyBTeXN0ZW1zMRswGQYDVQQDExJDaXNjbyBSb290IENB
IDIwNDgwHhcNMDQwNTE0MjAxNzEyWhcNMjkwNTE0MjAyNTQyWjA1MRYwFAYDVQQK
Ew1DaXNjbyBTeXN0ZW1zMRswGQYDVQQDExJDaXNjbyBSb290IENBIDIwNDgwggEg
MA0GCSqGSIb3DQEBAQUAA4IBDQAwggEIAoIBAQCwmrmrp68Kd6ficba0ZmKUeIhH
xmJVhEAyv8CrLqUccda8bnuoqrpu0hWISEWdovyD0My5jOAmaHBKeN8hF570YQXJ
FcjPFto1YYmUQ6iEqDGYeJu5Tm8sUxJszR2tKyS7McQr/4NEb7Y9JHcJ6r8qqB9q
VvYgDxFUl4F1pyXOWWqCZe+36ufijXWLbvLdT6ZeYpzPEApk0E5tzivMW/VgpSdH
jWn0f84bcN5wGyDWbs2mAag8EtKpP6BrXruOIIt6keO1aO6g58QBdKhTCytKmg9l
Eg6CTY5j/e/rmxrbU6YTYK/CfdfHbBcl1HP7R2RQgYCUTOG/rksc35LtLgXfAgED
o1EwTzALBgNVHQ8EBAMCAYYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUJ/PI
FR5umgIJFq0roIlgX9p7L6owEAYJKwYBBAGCNxUBBAMCAQAwDQYJKoZIhvcNAQEF
BQADggEBAJ2dhISjQal8dwy3U8pORFBi71R803UXHOjgxkhLtv5MOhmBVrBW7hmW
Yqpao2TB9k5UM8Z3/sUcuuVdJcr18JOagxEu5sv4dEX+5wW4q+ffy0vhN4TauYuX
cB7w4ovXsNgOnbFp1iqRe6lJT37mjpXYgyc81WhJDtSd9i7rp77rMKSsH0T8lasz
Bvt9YAretIpjsJyp8qS5UwGH0GikJ3+r/+n6yUA4iGe0OcaEb1fJU9u6ju7AQ7L4
CYNu/2bPPu8Xs1gYJQk0XuPL1hS27PKSb3TkL4Eq1ZKR4OCXPDJoBYVL0fdX4lId
kxpUnwVwwEpxYB5DC2Ae/qPOgRnhCzU=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIEPDCCAySgAwIBAgIKYQlufQAAAAAADDANBgkqhkiG9w0BAQUFADA1MRYwFAYD
VQQKEw1DaXNjbyBTeXN0ZW1zMRswGQYDVQQDExJDaXNjbyBSb290IENBIDIwNDgw
```

```
HhcNMTEwNjMwMTc1NjU3WhcNMjkwNTE0MjAyNTQyWjAnMQ4wDAYDVQQKEwVDaXNj
bzEVMBMGA1UEAxMMQUNUMiBTVURJIENBMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8A
MIIBCgKCAQEA0m5l3THIxA9tN/hS5qR/6UZRpdd+9aE2JbFkNjht6gfHKd477AkS
5XAtUs5oxDYVt/zEbslZq3+LR6qrqKKQVu6JYvH05UYLBqCj38s76NLk53905Wzp
9pRcmRCPuX+a6tHF/qRuOiJ44mdeDYZo3qPCpxzprWJDPclM4iYKHumMQMqmgmg+
xghHIooWS80BOcdiynEbeP5rZ7qRuewKMpl1TiI3WdBNjZjnpfjg66F+P4SaDkGb
BXdGj13oVeF+EyFWLrFjj97fL2+8oauV43Qrvnf3d/GfqXj7ew+z/sXlXtEOjSXJ
URsyMEj53Rdd9tJwHky8neapszS+r+kdVQIDAQABo4IBWjCCAVYwCwYDVR0PBAQD
AgHGMB0GA1UdDgQWBBRI2PHxwnDVW7t8cwmTr7i4MAP4fzAfBgNVHSMEGDAWgBQn
88gVHm6aAgkWrSugiWBf2nsvqjBDBgNVHR8EPDA6MDigNqA0hjJodHRwOi8vd3d3
LmNpc2NvLmNvbS9zZWN1cml0eS9wa2kvY3JsL2NyY2EyMDQ4LmNybDBQBggrBgEF
BQcBAQREMEIwQAYIKwYBBQUHMAKGNGh0dHA6Ly93d3cuY2lzY28uY29tL3NlY3Vy
aXR5L3BraS9jZXJ0cy9jcmNhMjA0OC5jZXIwXAYDVR0gBFUwUzBRBgorBgEEAQkV
AQwAMEMwQQYIKwYBBQUHAgEWNWh0dHA6Ly93d3cuY2lzY28uY29tL3NlY3VyaXR5
L3BraS9wb2xpY2llcy9pbmRleC5odG1sMBIGA1UdEwEB/wQIMAYBAf8CAQAwDQYJ
KoZIhvcNAQEFBQADggEBAGh1qclr9tx4hzWgDERm371yeuEmqcIfi9b9+GbMSJbi
ZHc/CcCl0lJu0a9zTXA9w47H9/t6leduGxb4WeLxcwCiUgvFtCa51Iklt8nNbcKY
/4dw1ex+7amATUQO4QggIE67wVIPu6bgAE3Ja/nRS3xKYSnj8H5TehimBSv6TECi
i5jUhOWryAK4dVo8hCjkjEkzu3ufBTJapnv89g9OE+H3VKM4L+/KdkUO+52djFKn
hyl47d7cZR4DY4LIuFM2P1As8YyjzoNpK/urSRI14WdIlplR1nH7KNDl5618yfVP
0IFJZBGrooCRBjOSwFv8cpWCbmWdPaCQT2nwIjTfY8c=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIDfTCCAmWgAwIBAgIEAwQD7zANBgkqhkiG9w0BAQsFADAnMQ4wDAYDVQQKEwVD
aXNjbzEVMBMGA1UEAxMMQUNUMiBTVURJIENBMB4XDTE4MDkyMzIyMzIwNloXDTI5
MDUxNDIwMjU0MVowaTEnMCUGA1UEBRMeUElEOkM5NjAwLVNVUC0xIFNOOkNBVDIy
MzZMMFE5MQ4wDAYDVQQKEwVDaXNjbzEYMBYGA1UECxMPQUNULTIgTGl0ZSBTVURJ
MRQwEgYDVQQDEwtDOTYwMC1TVVAtMTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCC
AQoCggEBANsh0jcvgh1pdOjP9KnffDnDc/zEHDzbCTWPJi2FZcsaSE5jvq6CUqc4
MYpNAZU2Jym7NSD8iQbMXwbnCtoL64QtxQeFhRYmc4d5o933M7GwpEH0I7HUSbO/
Fxyp7JBmGPPgAkY7rKsYENiNK2hiR7Q2O7X2BidOKknEuofWdJMNyMaZgLYLOHbJ
5oXaORxhUy3VRaxNl6qI7kYxuugg2LcAbZ539sRXe8JtHyK8llURNSGMiQ0S17pS
idGmrJJ0pEHA0EUVTZqEny3z+NW9uxLVSzu6+hEJYlqfI+YEf0DbVZly1cy5r/jF
yNdGuGKvd5agvgCly8aYMZa3P+D5S8sCAwEAAaNvMG0wDgYDVR0PAQH/BAQDAgXg
MAwGA1UdEwEB/wQCMAAwTQYDVR0RBEYwRKBCBgkrBgEEAQkVAgOgNRMzQ2hpcElE
PVUxUk5TVEl3TVRjd05qSTFBQUFwZndBQUFBQUFBQUFBQUFBQUhtSlU9MA0G
CSqGSIb3DQEBCwUAA4IBAQCrpHo/CUyk5Hs/asIcYW0ep8KocSkbNh8qamyd4oWD
e/MGJW9Bs5f09IEbILWPdytCCS2lSyJbxz2HvVDzdxQdxjDwUNiWuu3dWMXN/i67
yuCGM+lA1AAG5dT6lNgWYHh+YzsZm9eoq1+4NM+JuMXWsnzAK8rSy+dSpBxqFsBq
E0OlPsaK7y2h8gs+XrV9x+D48OZQkTRXpxhJfiWvs+EbdgsAM/vBxTAoTJPVmXWN
Cmcj9X52Xl3i4MdOUXocZLO2kh6JSgOYGkFeZifJ0iDvMfAf0cJ6+cEF6bSxAqBL
veel+8LmeiE/2O9h6qGHPPDacCaXA2oJCDHveAt8iPTG
-----END CERTIFICATE-----

Signature version: 1
Signature:
[redacted]
```

The optional RSA 2048 signature is across the three certificates, the signature version and the user-provided nonce.

```
RSA PKCS#1v1.5 Sign {<Nonce (UINT64)> || <Signature Version (UINT32)> || <Cisco Root CA
2048 cert (DER)> ||
<Cisco subordinate CA (DER)> || <SUDI certificate (DER)> }
```

Cisco management solutions are equipped with the ability to interpret the above output. However, a simple script using OpenSSL commands can also be used to display the identity of the platform and to verify the signature, thereby ensuring its Cisco unique device identity.

```
[linux-host:~]openssl x509 -in sudi_id.pem -subject -noout
subject= /serialNumber=PID:C9600-SUP-1 SN:CAT2239L06B/CN=C9600-SUP-1-70b3171eaa00
```

Verifying Software Integrity

The following example displays the checksum record for the boot stages. The hash measurements are displayed for each of the three stages of software successively booted. These hashes can be compared against Cisco-provided reference values. An option to sign the output gives a verifier the ability to ensure the output is genuine and is not altered. A nonce can be provided to protect against replay attacks.

Note Boot integrity hashes are not MD5 hashes. For example, if you run **verify /md5 cat9k_iosxe.16.10.01.SPA.bin** command for the bundle file, the hash will not match.

The following is a sample output of the **show platform integrity sign nonce 123** command. This output includes measurements of each installed package file.

```
Device#show platform integrity sign nonce 123
Platform: C9606R
Boot 0 Version: MA0083R06.1810032017
Boot 0 Hash: 535AD9DC3D2A26C030D7DF6D4342FD52AB4DC6B1395DB18E7CA33F678A874B9E
Boot Loader Version: System Bootstrap, Version 16.11.1r[FC2], RELEASE SOFTWARE (P)
Boot Loader Hash:
C66199E7F63242A45BEAA0A8FCB5C17432FA13AF82FB1596D5CFEE1FF1080F2107FEEFB48AC5DF88B41894AEC7AF87052717012BFF6185D34F579D9BF7184597
OS Version: BLD_V1611_THROTTLE_LATEST_20190203_030036
OS Hashes:
cat9k_iosxe.BLD_V1611_THROTTLE_LATEST_20190203_030036.SSA.bin:
3F4A10066EAAA30417D7D17395ADDD71FFCCED6ABAA122ABA439D12A03C78EF38B8D281DEFA2D7CC15AA7FE63AA1344FEABF68AC6409D408F89277F35DB8EE55
cat9k-wlc.BLD_V1611_THROTTLE_LATEST_20190203_030036.SSA.pkg:
2F0894E3F3A1332EDF2B2733EB456A4EB57E1A417BF46B53AD1323D1B02BA7688667C84AC7BD274B6B3A5DD3D19EB7EDA5DAB13E9941A37C73256C7577F3A3A1
cat9k-webui.BLD_V1611_THROTTLE_LATEST_20190203_030036.SSA.pkg:
ADE97B8FA0AC1C2694ECA93C96F77DCC0E96D7D36134795A4197AF60B9B2E9BE582C0535E9CE11A5EBD50542C6A94B55742E916185E333D3EF9E716D16AD0FDD
cat9k-guestshell.BLD_V1611_THROTTLE_LATEST_20190203_030036.SSA.pkg:
174AE72DF46F86D5ADD0A73344295A91C809CD42E6C12FEB29024215DAC89140511EE2EDFEF8E5CEAD731B4276C85B3F7D5BF9386083CCBE3E4C504E1E0400E1
cat9k-srdriver.BLD_V1611_THROTTLE_LATEST_20190203_030036.SSA.pkg:
64884593C2281B687374B283E14BFCF89F69D37EB4C238E7D71FA280B940FD0D11F57BAFF16788AA054AEEE6B898BC689D623DDB25C743069538A7E83F146240
cat9k-sipbase.BLD_V1611_THROTTLE_LATEST_20190203_030036.SSA.pkg:
0AFF960435A97C9FA3522AC93E5CE1A683003C93CEBD4288AA8AF481E3D9D8806451A23022AE5E810A010B6196B802CFA5D1354DDCC6B7A7120FF4A915B9ECF9
cat9k-espbase.BLD_V1611_THROTTLE_LATEST_20190203_030036.SSA.pkg:
6D5324CD00E578EFFE5C874620900ADEBFC38CD05B01F43B4E579E267D581145FE5BEFCE5EDD09EE12338FDB2A162A389BED6C951AF8C394AE5FBAF4EAE4D7E9
cat9k-cc_srdriver.BLD_V1611_THROTTLE_LATEST_20190203_030036.SSA.pkg:
59362BDDD62AB1E94297891D8ECEEB467FB28261B6D75F6442610DD41A8E54D69609C94D081D32142412CC69C5C88036F26BE5F356B848ACBCEB5692A423D92F
cat9k-sipspa.BLD_V1611_THROTTLE_LATEST_20190203_030036.SSA.pkg:
708B0D0869E841CD9220C916C566C46D07CE206FBAD294498E81A915E69F33063B9AFC0BBB5B048F250150E07EA37160AA8E5AA4CD491E402C836A6322631175
cat9k-rpbase.BLD_V1611_THROTTLE_LATEST_20190203_030036.SSA.pkg:
F24FB8347047A3D0930F8B353B2494EFCB6E0FB60B2A1BFE5F9C322EBC675A0A5D94CDC36195B41971F5B47383FB095BC731FB45407D42DE57BA14E3E6DEEFBE
PCR0: 7803FB049E7B111131B2FDACAF9B1918C28448E250054FE0C65D0317427A5EB1
PCR8: 0B65A1D00AA4AC815552170D11E5B4405C6D4B80453925E54F866D5BDF2B718A
Signature version: 1
Signature:
[redacted]
```

Verifying Image Signing

The following example displays the secure code signing check of the image during bootup using an SHA-512 hash.

```
switch:boot flash:packages.conf
boot: attempting to boot from [flash:packages.conf]
```

```
boot: reading file packages.conf
#
Performing Integrity Check ...
boot: parsed image from conf file: cat9k-rpboot.17.02.01.SSA.pkg
##########################################################################################################################################################



Loading image in Verbose mode: 1



Image Base is: 0x100099000
Image Size is: 0x2C83487
Package header rev 3 structure detected
Package type:30001, flags:0x0
IsoSize = 0
Parsing package TLV info:
000: 000000090000001D4B45595F544C565F -         KEY_TLV_
010: 5041434B4147455F434F4D5041544942 - PACKAGE_COMPATIB
020: 494C4954590000000000000090000000B - ILITY
030: 4652555F52505F545950450000000009 - FRU_RP_TYPE
040: 000000184B45595F544C565F5041434B -     KEY_TLV_PACK
050: 4147455F424F4F54415243480000000009 - AGE_BOOTARCH
060: 0000000E415243485F693638365F5459 -     ARCH_i686_TY
070: 50450000000000009000000144B45595F - PE          KEY_
080: 544C565F424F4152445F434F4D504154 - TLV_BOARD_COMPAT
090: 0000000900000010424F4152445F6361 -         BOARD_ca
0A0: 74396B5F545950450000000900000018 - t9k_TYPE
0B0: 4B45595F544C565F43525950544F5F4B - KEY_TLV_CRYPTO_K
0C0: 4559535452494E470000000900000004 - EYSTRING

TLV: T=9, L=29, V=KEY_TLV_PACKAGE_COMPATIBILITY
TLV: T=9, L=11, V=FRU_RP_TYPE
TLV: T=9, L=24, V=KEY_TLV_PACKAGE_BOOTARCH
TLV: T=9, L=14, V=ARCH_i686_TYPE
TLV: T=9, L=20, V=KEY_TLV_BOARD_COMPAT
TLV: T=9, L=16, V=BOARD_cat9k_TYPE
TLV: T=9, L=24, V=KEY_TLV_CRYPTO_KEYSTRING
TLV: T=9, L=4, V=none
TLV: T=9, L=11, V=CW_BEGIN=$$
TLV: T=9, L=17, V=CW_FAMILY=$cat9k$
TLV: T=9, L=74, V=CW_IMAGE=$cat9k-rpboot.17.02.01.SSA.pkg$
TLV: T=9, L=20, V=CW_VERSION=$17.2.01$
IOS version is 17.2.1
TLV: T=9, L=53, V=CW_FULL_VERSION=$17.2.01.0.869.1580816579..Amsterdam$
TLV: T=9, L=52, V=CW_DESCRIPTION=$Cisco IOS Software, IOS-XE Software$
TLV: T=9, L=9, V=CW_END=$$
Found DIGISIGN TLV type 12 length = 392
RSA Self Test Passed

Expected hash:
DDAF35A193617ABACC417349AE204131
12E6FA4E89A97EA20A9EEEE64B55D39A
2192992A274FC1A836BA3C23A3FEEBBD
454D4423643CE80E2A9AC94FA54CA49F

Obtained hash:
DDAF35A193617ABACC417349AE204131
12E6FA4E89A97EA20A9EEEE64B55D39A
2192992A274FC1A836BA3C23A3FEEBBD
454D4423643CE80E2A9AC94FA54CA49F
Sha512 Self Test Passed
Found package arch type ARCH_i686_TYPE
```

```
Found package FRU type FRU_RP_TYPE
Performing Integrity Check ...

RSA Signed DEVELOPMENT Image Signature Verification Successful.
```

Additional References for Boot Integrity Visibility

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	*Command Reference (Catalyst 9600 Series Switches)*

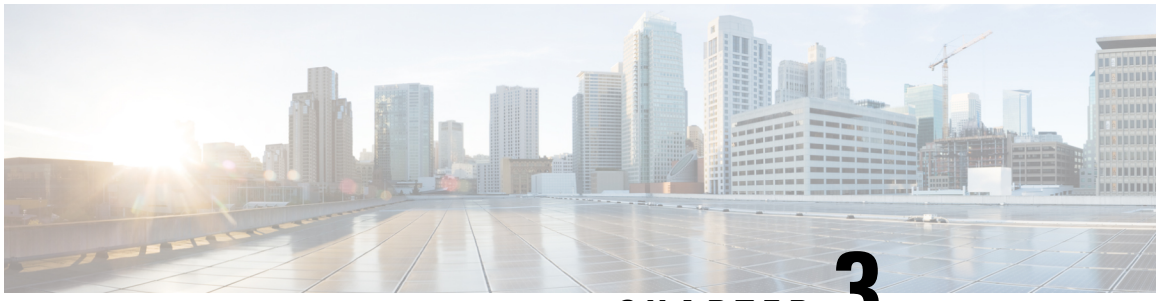
Feature History for Boot Integrity Visibility

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.11.1	Boot Integrity Visibility	Boot Integrity Visibility allows Cisco's platform identity and software integrity information to be visible and actionable. Platform identity provides the platform's manufacturing installed identity.
Cisco IOS XE Cupertino 17.7.1	Boot Integrity Visibility	Boot Integrity Visibility allows Cisco's platform identity and software integrity information to be visible and actionable. Platform identity provides the platform's manufacturing installed identity. Support for this feature was introduced on the Cisco Catalyst 9600 Series Supervisor 2 Module.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.



CHAPTER 3

Performing Device Setup Configuration

Information About Performing Device Setup Configuration

The following sections provide information about how to perform a device setup configuration, including IP address assignments and Dynamic Host Configuration Protocol (DHCP) auto configuration.

Device Boot Process

To start your device, you need to follow the procedures described in the *Cisco Catalyst 9600 Series Switches Hardware Installation Guide* for installing and powering on the device and setting up the initial device configuration.

The normal boot process involves the operation of the boot loader software and includes these activities:

- Performs low-level CPU initialization. This process initializes the CPU registers that control where physical memory is mapped, the quantity and speed of the physical memory, and so forth.
- Initializes the file systems on the system board.
- Loads a default operating system software image into memory and boots up the device.
- Performs power-on self-test (POST) for the CPU subsystem and tests the system DRAM. As part of POST, the following test is also performed:
  - MAC loopback test to verify the data path between the CPU and network ports connected to each module. If this test fails for any of the ports, the ports are forced into error-disabled state, and the module is marked as *post-fail* in the **show module** command output.

  For information about the complete list of supported online diagnostics, see the Configuring Online Diagnostics chapter.

The boot loader provides access to the file systems before the operating system is loaded. Normally, the boot loader is used only to load, decompress, and start the operating system. After the boot loader gives the operating system control of the CPU, the boot loader is not active until the next system reset or power-on.

Before you can assign device information, make sure you have connected a PC or terminal to the console port or a PC to the Ethernet management port, and make sure you have configured the PC or terminal-emulation software baud rate and character format to match these of the device console port:

- Baud rate default is 9600.
- Data bits default is 8.

Note If the data bits option is set to 8, set the parity option to none.

- Stop bits default is 2 (minor).
- Parity settings default is none.

Devices Information Assignment

You can assign IP information through the device setup program, through a DHCP server, or manually.

Use the device setup program if you want to be prompted for specific IP information. With this program, you can also configure a hostname and an enable secret password. For a new switch, enter a new password for enable secret password.

It gives you the option of assigning a Telnet password (to provide security during remote management) and configuring your switch as a command or member switch of a cluster or as a standalone switch.

Use a DHCP server for centralized control and automatic assignment of IP information after the server is configured.

Note If you are using DHCP, do not respond to any of the questions in the setup program until the device receives the dynamically assigned IP address and reads the configuration file.

If you are an experienced user familiar with the device configuration steps, manually configure the device. Otherwise, use the setup program described in section Device Boot Process, on page 49.

Default Switch Information

Table 4: Default Switch Information

Feature	Default Setting
IP address and subnet mask	No IP address or subnet mask are defined.
Default gateway	No default gateway is defined.
Enable secret password	No password is defined.

Feature	Default Setting
Hostname	The factory-assigned default hostname is device.
Telnet password	No password is defined.
Cluster command switch functionality	Disabled.
Cluster name	No cluster name is defined.

DHCP-Based Autoconfiguration Overview

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

During DHCP-based autoconfiguration, your device (DHCP client) is automatically configured at startup with IP address information and a configuration file.

With DHCP-based autoconfiguration, no DHCP client-side configuration is needed on your device. However, you need to configure the DHCP server for various lease options associated with IP addresses.

If you want to use DHCP to relay the configuration file location on the network, you might also need to configure a Trivial File Transfer Protocol (TFTP) server and a Domain Name System (DNS) server.

The DHCP server for your device can be on the same LAN or on a different LAN than the device. If the DHCP server is running on a different LAN, you should configure a DHCP relay device between your device and the DHCP server. A relay device forwards broadcast traffic between two directly connected LANs. A router does not forward broadcast packets, but it forwards packets based on the destination IP address in the received packet.

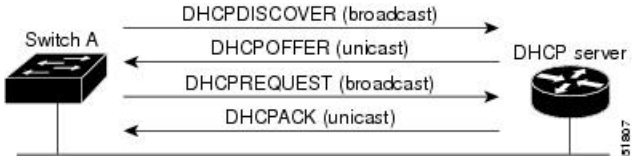
DHCP-based autoconfiguration replaces the BOOTP client functionality on your device.

DHCP Client Request Process

When you boot up your device, the DHCP client is invoked and requests configuration information from a DHCP server when the configuration file is not present on the device. If the configuration file is present and the configuration includes the **ip address dhcp** interface configuration command on specific routed interfaces, the DHCP client is invoked and requests the IP address information for those interfaces.

This is the sequence of messages that are exchanged between the DHCP client and the DHCP server.

Figure 2: DHCP Client and Server Message Exchange



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

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

The DHCP server confirms that the IP address has been allocated to the client by returning a DHCPACK unicast message to the client. With this message, the client and server are bound, and the client uses configuration information received from the server. The amount of information the device receives depends on how you configure the DHCP server.

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

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

A DHCP client might receive offers from multiple DHCP or BOOTP servers and can accept any of the offers; however, the client usually accepts the first offer it receives. The offer from the DHCP server is not a guarantee that the IP address is allocated to the client; however, the server usually reserves the address until the client has had a chance to formally request the address. If the device accepts replies from a BOOTP server and configures itself, the device broadcasts, instead of unicasts, TFTP requests to obtain the device configuration file.

The DHCP hostname option allows a group of devices to obtain hostnames and a standard configuration from the central management DHCP server. A client (device) includes in its DCHPDISCOVER message an option 12 field used to request a hostname and other configuration parameters from the DHCP server. The configuration files on all clients are identical except for their DHCP-obtained hostnames.

DHCP-Based Autoconfiguration and Image Update

You can use the DHCP image upgrade features to configure a DHCP server to download both a new image and a new configuration file to one or more devices in a network. Simultaneous image and configuration upgrade for all switches in the network helps ensure that each new device added to a network receives the same image and configuration.

There are two types of DHCP image upgrades: DHCP autoconfiguration and DHCP auto-image update.

Restrictions for DHCP-Based Autoconfiguration

- The DHCP-based autoconfiguration with a saved configuration process stops if there is not at least one Layer 3 interface in an up state without an assigned IP address in the network.
- Unless you configure a timeout, the DHCP-based autoconfiguration with a saved configuration feature tries indefinitely to download an IP address.
- The auto-install process stops if a configuration file cannot be downloaded or if the configuration file is corrupted.
- The configuration file that is downloaded from TFTP is merged with the existing configuration in the running configuration but is not saved in the NVRAM unless you enter the **write memory** or **copy running-configuration startup-configuration** privileged EXEC command. If the downloaded configuration is saved to the startup configuration, the feature is not triggered during subsequent system restarts.

DHCP Autoconfiguration

DHCP autoconfiguration downloads a configuration file to one or more devices in your network from a DHCP server. The downloaded configuration file becomes the running configuration of the device. It does not over write the bootup configuration saved in the flash, until you reload the device.

DHCP Auto-Image Update

You can use DHCP auto-image upgrade with DHCP autoconfiguration to download both a configuration and a new image to one or more devices in your network. The devices (or devices) downloading the new configuration and the new image can be blank (or only have a default factory configuration loaded).

If the new configuration is downloaded to a switch that already has a configuration, the downloaded configuration is appended to the configuration file stored on the switch. (Any existing configuration is not overwritten by the downloaded one.)

To enable a DHCP auto-image update on the device, the TFTP server where the image and configuration files are located must be configured with the correct option 67 (the configuration filename), option 66 (the DHCP server hostname) option 150 (the TFTP server address), and option 125 (description of the Cisco IOS image file) settings.

After you install the device in your network, the auto-image update feature starts. The downloaded configuration file is saved in the running configuration of the device, and the new image is downloaded and installed on the device. When you reboot the device, the configuration is stored in the saved configuration on the device.

DHCP Server Configuration Guidelines

Follow these guidelines if you are configuring a device as a DHCP server:

- You should configure the DHCP server with reserved leases that are bound to each device by the device hardware address.
- If you want the device to receive IP address information, you must configure the DHCP server with these lease options:
  - IP address of the client (required)
  - Subnet mask of the client (required)
  - DNS server IP address (optional)
  - Router IP address (default gateway address to be used by the device) (required)
- If you want the device to receive the configuration file from a TFTP server, you must configure the DHCP server with these lease options:
  - TFTP server name (required)
  - Boot filename (the name of the configuration file that the client needs) (recommended)
  - Hostname (optional)
- Depending on the settings of the DHCP server, the device can receive IP address information, the configuration file, or both.

- If you do not configure the DHCP server with the lease options described previously, it replies to client requests with only those parameters that are configured. If the IP address and the subnet mask are not in the reply, the device is not configured. If the router IP address or the TFTP server name are not found, the device might send broadcast, instead of unicast, TFTP requests. Unavailability of other lease options does not affect autoconfiguration.
- The device can act as a DHCP server. By default, the Cisco IOS DHCP server and relay agent features are enabled on your device but are not configured. (These features are not operational.)

Purpose of the TFTP Server

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

If you did not specify the configuration filename, the TFTP server, or if the configuration file could not be downloaded, the device attempts to download a configuration file by using various combinations of filenames and TFTP server addresses. The files include the specified configuration filename (if any) and these files: network-config, cisconet.cfg, *hostname*.config, or *hostname*.cfg, where *hostname* is the device's current hostname. The TFTP server addresses used include the specified TFTP server address (if any) and the broadcast address (255.255.255.255).

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

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

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

If the TFTP server to be used is on a different LAN from the device, or if it is to be accessed by the device through the broadcast address (which occurs if the DHCP server response does not contain all the required information described previously), a relay must be configured to forward the TFTP packets to the TFTP server. The preferred solution is to configure the DHCP server with all the required information.

Purpose of the DNS Server

The DHCP server uses the DNS server to resolve the TFTP server name to an IP address. You must configure the TFTP server name-to-IP address map on the DNS server. The TFTP server contains the configuration files for the device.

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

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

How to Obtain Configuration Files

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

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

  The device receives its IP address, subnet mask, TFTP server address, and the configuration filename from the DHCP server. The device sends a unicast message to the TFTP server to retrieve the named configuration file from the base directory of the server and upon receipt, it completes its boot up process.

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

  The device receives its IP address, subnet mask, and the configuration filename from the DHCP server. The device sends a broadcast message to a TFTP server to retrieve the named configuration file from the base directory of the server, and upon receipt, it completes its boot-up process.

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

  The device receives its IP address, subnet mask, and the TFTP server address from the DHCP server. The device sends a unicast message to the TFTP server to retrieve the network-confg or cisconet.cfg default configuration file. (If the network-confg file cannot be read, the device reads the cisconet.cfg file.)

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

  After obtaining its hostname from the default configuration file or the DHCP reply, the device reads the configuration file that has the same name as its hostname (*hostname*-confg or *hostname*.cfg, depending on whether network-confg or cisconet.cfg was read earlier) from the TFTP server. If the cisconet.cfg file is read, the filename of the host is truncated to eight characters.

  If the device cannot read the network-confg, cisconet.cfg, or the hostname file, it reads the router-confg file. If the device cannot read the router-confg file, it reads the ciscortr.cfg file.

Note The device broadcasts TFTP server requests if the TFTP server is not obtained from the DHCP replies, if all attempts to read the configuration file through unicast transmissions fail, or if the TFTP server name cannot be resolved to an IP address.

How to Control Environment Variables

With a normally operating device, you enter the boot loader mode only through the console connection configured for 9600 bps. Unplug the device power cord, and press the **Mode** button while reconnecting the power cord. The boot loader device prompt then appears.

The device boot loader software provides support for nonvolatile environment variables, which can be used to control how the boot loader, or any other software running on the system, operates. Boot loader environment variables are similar to environment variables that can be set on UNIX or DOS systems.

Environment variables that have values are stored in flash memory outside of the flash file system.

Each line in these files contains an environment variable name and an equal sign followed by the value of the variable. A variable has no value if it is not present; it has a value if it is listed even if the value is a null string. A variable that is set to a null string (for example, " ") is a variable with a value. Many environment variables are predefined and have default values.

You can change the settings of the environment variables by accessing the boot loader or by using Cisco IOS commands. Under normal circumstances, it is not necessary to alter the setting of the environment variables.

Scheduled Reload of the Software Image

You can schedule a reload of the software image to occur on the device at a later time (for example, late at night or during the weekend when the device is used less), or you can synchronize a reload network-wide (for example, to perform a software upgrade on all device in the network).

Note A scheduled reload must take place within approximately 24 days.

You have these reload options:

- Reload of the software to take affect in the specified minutes or hours and minutes. The reload must take place within approximately 24 hours. You can specify the reason for the reload in a string up to 255 characters in length.
- Reload of the software to take place at the specified time (using a 24-hour clock). If you specify the month and day, the reload is scheduled to take place at the specified time and date. If you do not specify the month and day, the reload takes place at the specified time on the current day (if the specified time is later than the current time) or on the next day (if the specified time is earlier than the current time). Specifying 00:00 schedules the reload for midnight.

The **reload** command halts the system. If the system is not set to manually boot up, it reboots itself.

If your device is configured for manual booting, do not reload it from a virtual terminal. This restriction prevents the device from entering the boot loader mode and then taking it from the remote user's control.

If you modify your configuration file, the device prompts you to save the configuration before reloading. During the save operation, the system requests whether you want to proceed with the save if the CONFIG_FILE environment variable points to a startup configuration file that no longer exists. If you proceed in this situation, the system enters setup mode upon reload.

To cancel a previously scheduled reload, use the **reload cancel** privileged EXEC command.

How to Perform Device Setup Configuration

Using DHCP to download a new image and a new configuration to a device requires that you configure at least two devices. One device acts as a DHCP and TFTP server and the second device (client) is configured to download either a new configuration file or a new configuration file and a new image file.

Configuring DHCP Autoconfiguration (Only Configuration File)

This task describes how to configure DHCP autoconfiguration of the TFTP and DHCP settings on an existing device in the network so that it can support the autoconfiguration of a new device.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**ip dhcp pool** *poolname* **Example:** `Device(config)# ip dhcp pool pool`	Creates a name for the DHCP server address pool, and enters DHCP pool configuration mode.
Step 3	**boot** *filename* **Example:** `Device(dhcp-config)# boot config-boot.text`	Specifies the name of the configuration file that is used as a boot image.
Step 4	**network** *network-number mask prefix-length* **Example:** `Device(dhcp-config)# network 10.10.10.0 255.255.255.0`	Specifies the subnet network number and mask of the DHCP address pool. **Note** The prefix length specifies the number of bits that comprise the address prefix. The prefix is an alternative way of specifying the network mask of the client. The prefix length must be preceded by a forward slash (/).
Step 5	**default-router** *address* **Example:** `Device(dhcp-config)# default-router 10.10.10.1`	Specifies the IP address of the default router for a DHCP client.
Step 6	**option 150** *address* **Example:** `Device(dhcp-config)# option 150 10.10.10.1`	Specifies the IP address of the TFTP server.

	Command or Action	Purpose
Step 7	**exit** **Example:** `Device(dhcp-config)# exit`	Returns to global configuration mode.
Step 8	**tftp-server flash**:*filename.text* **Example:** `Device(config)# tftp-server flash:config-boot.text`	Specifies the configuration file on the TFTP server.
Step 9	**interface** *interface-id* **Example:**	Specifies the address of the client that will receive the configuration file.
Step 10	**no switchport** **Example:** `Device(config-if)# no switchport`	Puts the interface into Layer 3 mode.
Step 11	**ip address** *address mask* **Example:** `Device(config-if)# ip address 10.10.10.1 255.255.255.0`	Specifies the IP address and mask for the interface.
Step 12	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Manually Assigning IP Information to Multiple SVIs

This task describes how to manually assign IP information to multiple switched virtual interfaces (SVIs):

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface vlan** *vlan-id* **Example:** `Device(config)# interface vlan 99`	Enters interface configuration mode, and enters the VLAN to which the IP information is assigned. The range is 1 to 4094.
Step 4	**ip address** *ip-address subnet-mask* **Example:** `Device(config-vlan)# ip address 10.10.10.2 255.255.255.0`	Enters the IP address and subnet mask.
Step 5	**exit** **Example:** `Device(config-vlan)# exit`	Returns to global configuration mode.
Step 6	**ip default-gateway** *ip-address* **Example:** `Device(config)# ip default-gateway 10.10.10.1`	Enters the IP address of the next-hop router interface that is directly connected to the device where a default gateway is being configured. The default gateway receives IP packets with unresolved destination IP addresses from the device. Once the default gateway is configured, the device has connectivity to the remote networks with which a host needs to communicate. **Note** When your device is configured to route with IP, it does not need to have a default gateway set. **Note** The device capwap relays on default-gateway configuration to support routed access point join the device.
Step 7	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

	Command or Action	Purpose
Step 8	**show interfaces vlan** *vlan-id* **Example:** `Device# show interfaces vlan 99`	Displays the interfaces status for the specified VLAN.
Step 9	**show ip redirects** **Example:** `Device# show ip redirects`	Displays the Internet Control Message Protocol (ICMP) redirect messages.

Modifying Device Startup Configuration

The following sections provide information on how to modify the startup configuration of a device.

Specifying a Filename to Read and Write a System Configuration

By default, the Cisco IOS software uses the config.text file to read and write a nonvolatile copy of the system configuration. However, you can specify a different filename, which will be loaded during the next boot cycle.

Before you begin

Use a standalone device for this task.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**boot flash**:*/file-url* **Example:** `Device(config)# boot flash:config.text`	Specifies the configuration file to load during the next boot cycle. • *file-url*: The path (directory) and the configuration filename. • Filenames and directory names are case-sensitive.

	Command or Action	Purpose
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 5	**show boot** **Example:** `Device# show boot`	Lists the contents of the BOOT environment variable (if set), the name of the configuration file pointed to by the CONFIG_FILE environment variable, and the contents of the BOOTLDR environment variable. • The **boot** global configuration command changes the setting of the CONFIG_FILE environment variable.
Step 6	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	(Optional) Saves your entries in the configuration file.

Configuring a Scheduled Software Image Reload

This task describes how to configure your device to reload the software image at a later time.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	Saves your device configuration information to the startup configuration before you use the **reload** command.

	Command or Action	Purpose

Step 4	**reload in** [*hh*:]*mm* [*text*]
	Example:
	`Device# reload in 12`
	`System configuration has been modified. Save? [yes/no]: y`
		Schedules a reload of the software to take affect in the specified minutes or hours and minutes. The reload must take place within approximately 24 days. You can specify the reason for the reload in a string up to 255 characters in length.

Step 5	**reload at** *hh*: *mm* [*month day* | *day month*] [*text*] **Example:** `Device(config)# reload at 14:00`	Specifies the time in hours and minutes for the reload to occur. **Note** Use the **at** keyword only if the device system clock has been set (through Network Time Protocol (NTP), the hardware calendar, or manually). The time is relative to the configured time zone on the device. To schedule reloads across several devices to occur simultaneously, the time on each device must be synchronized with NTP.
Step 6	**reload cancel** **Example:** `Device(config)# reload cancel`	Cancels a previously scheduled reload.
Step 7	**show reload** **Example:** `show reload`	Displays information about a previously scheduled reload or identifies if a reload has been scheduled on the device.

Configuration Examples for Device Setup Configuration

The following sections provide configuration examples for device setup.

Example: Configuring a Device to Download Configurations from a DHCP Server

The following example shows how to use a Layer 3 SVI interface on VLAN 99 to enable DHCP-based autoconfiguration with a saved configuration:

```
Device# configure terminal
Device(config)# boot host dhcp
Device(config)# boot host retry timeout 300
```

```
Device(config)# banner config-save ^C Caution - Saving Configuration File to NVRAM May Cause
 You to No longer Automatically Download Configuration Files at Reboot^C
Device(config)# vlan 99
Device(config-vlan)# interface vlan 99
Device(config-if)# no shutdown
Device(config-if)# end
Device# show boot

BOOT path-list:
Config file:          flash:/config.text
Private Config file:  flash:/private-config.text
Enable Break:         no
Manual Boot:          no
HELPER path-list:
NVRAM/Config file
      buffer size:    32768
Timeout for Config
          Download:    300 seconds
Config Download
       via DHCP:       enabled (next boot: enabled)
Device#
```

Example: Scheduling Software Image Reload

This example shows how to reload the software on a device on the current day at 7:30 p.m:

```
Device# reload at 19:30

Reload scheduled for 19:30:00 UTC Wed Jun 5 2013 (in 2 hours and 25 minutes)
Proceed with reload? [confirm]
```

This example shows how to reload the software on a device at a future date and time:

```
Device# reload at 02:00 jun 20

Reload scheduled for 02:00:00 UTC Thu Jun 20 2013 (in 344 hours and 53 minutes)
Proceed with reload? [confirm]
```

Additional References For Performing Device Setup

Related Documents

Related Topic	Document Title
Device setup commands Boot loader commands	*Command Reference (Catalyst 9600 Series Switches)*
Hardware installation	*Cisco Catalyst 9600 Series Switches Hardware Installation Guide* .

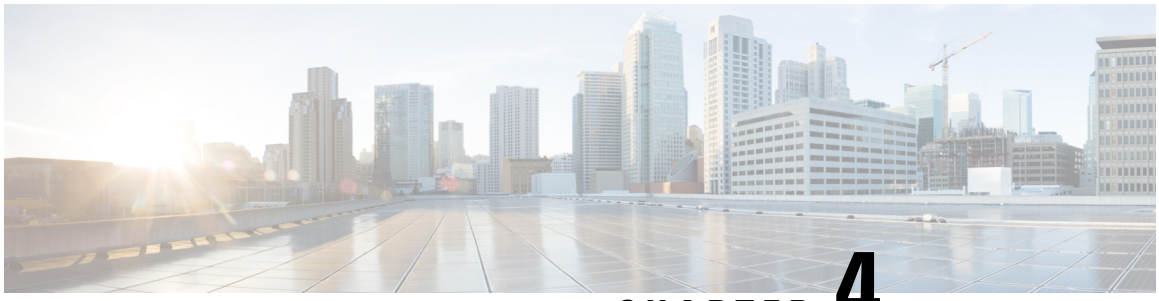
Feature History for Performing Device Setup Configuration

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.11.1	Device Setup Configuration	A device setup configuration can be performed, including auto configuration of IP address assignments and DHCP.
Cisco IOS XE Cupertino 17.7.1	Device Setup Configuration	Support for this feature was introduced on the Cisco Catalyst 9600 Series Supervisor 2 Module.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.



CHAPTER 4

Available Licenses

Information About Available Licenses

This section provides information about the licenses that are available on Cisco Catalyst 9600 Series Switches running Cisco IOS-XE software. The information applies to all models in the series, unless indicated otherwise.

Base and Add-On Licenses

The software features available on the switch fall under base or add-on license levels.

A base license is a perpetually valid, or permanent license. There is no expiration date for such a license.

An add-on license provides Cisco innovations on the switch, and on the Cisco Digital Network Architecture Center (Cisco DNA Center). An add-on license is valid only until a certain date. You can purchase an add-on license for a three, five, or seven year subscription period.

The following base and add-on licenses are available:

Base Licenses

Network Advantage

Add-On Licenses

DNA Advantage

Guidelines for Using Base and Add-On Licenses

- A base license (Network-Advantage) is ordered and fulfilled only with a perpetual or permanent license type.
- An add-on license (DNA Advantage) is ordered and fulfilled only with a subscription or term license type.

- An add-on license level is included when you choose a network license level. If you use DNA features, renew the license before term expiry, to continue using it. If you don't want to continue using DNA features, deactivate the add-on license and then reload the switch to continue operating with the base license capabilities.
- To know which license level a feature is available with, use Cisco Feature Navigator. To access Cisco Feature Navigator, go to https://cfnng.cisco.com. An account on cisco.com is not required.

Export Control Key for High Security

Products and features that provide cryptographic functionality are within the purview of U.S. export control laws [1]. The Export Control Key for High Security (HSECK9 key) is an export-controlled license, which authorizes the use of cryptographic functionality.

This subsection provides information about the Cisco Catalyst 9600 Series Switches that support the HSECK9 key, the cryptographic features that require the HSECK9 key, what to consider when ordering it, prerequisites, and how to configure it on supported platforms.

Supported Platforms and Releases

The HSECK9 key is supported on the Cisco Catalyst 9600 Series 40-Port 50G, 2-Port 200G, 2-Port 400G Line Card (C9600-LC-40YL4CD), starting with Cisco IOS XE Cupertino 17.8.1. This line card is compatible only with Cisco Catalyst 9600 Series Supervisor Engine 2 (C9600X-SUP-2).

For more information about the line card and compatibility, see Cisco Catalyst 9600 Series Line Card Installation Note and Cisco Catalyst 9600 Series Switches Hardware Installation Guide.

When an HSECK9 Key Is Required

An HSECK9 key is required only if you want to use certain cryptographic features that are restricted by U.S. export control laws. You cannot enable restricted cryptographic features without it.

The WAN MACsec feature requires an HSECK9 key. More specifically, the HSECK9 key is required on *customer edge devices* in a point-to-point (P2P) and point-to-multipoint (P2MP) network where the WAN MACsec feature is configured.

Prerequisites for Using an HSECK9 Key

Ensure you meet the following requirements:

- The device is one that supports the HSECK9 key. See Supported Platforms and Releases, on page 66.
- You have configured the DNA Advantage license on the device. You cannot use an HSECK9 key without DNA Advantage configured.
- You have the required number of HSECK9 keys in the applicable Smart Account and Virtual Account in Cisco Smart Software Manager (CSSM).

  The HSECK9 key is tied to the *chassis*. Each *chassis* UDI where you want to use a cryptographic feature requires one HSECK9 key. To understand this requirement in the context of a High Availability setup, see High Availability Considerations, on page 67.

[1] the U.S. Government Encryption and Export Administration Regulations (EAR)

- You have implemented one of the supported Smart Licensing Using Policy topologies. This enables you to install a Smart Licensing Authorization Code (SLAC) for each HSECK9 key you want to use.

  An HSECK9 key requires authorization *before* use, because it is restricted by U.S. trade-control laws (export-controlled). A SLAC provides this authorization and allows activation and continued use of an export-controlled license. A SLAC is generated in and obtained from CSSM. There are multiple ways in which a device can be connected to CSSM, to obtain a SLAC. Each way of connecting to CSSM is called a topology. The configuration section shows you how to obtain a SLAC with each topology (Installing SLAC for an HSECK9 Key, on page 71).

Note To obtain and install SLAC on supported platforms that are within the scope of this document (Supported Platforms and Releases, on page 66), refer to the configuration section in *this* document. There are differences in the configuration process when compared to other Cisco products.

- You configure the cryptographic feature only after you have installed SLAC. If not, you have to reconfigure the cryptographic feature after installing SLAC.
- The interface on which you configure the cryptographic feature must correspond with a linecard slot where a line card supporting the cryptographic feature is installed.

Ordering Considerations

This section covers important ordering considerations for an HSECK9 key.

The HSECK9 key is tied to the chassis UDI. Regardless of whether you have a single or dual supervisor set-up, and regardless of the number of linecards where the cryptographic feature is configured, only one license is required for a chassis. A separate HSECK9 key is required for each *chassis* UDI where you want to use a cryptographic feature.

If you plan to use cryptographic functionality on new hardware that you are ordering (supported platforms), provide your Smart Account and Virtual Account information with the order. This enables Cisco to factory-install SLAC.

For information about ordering the key, see the Cisco Catalyst 9600 Series Switches Ordering Guide.

High Availability Considerations

This section covers the High Availability considerations that apply *when using the HSECK9 key*.

- Supported High Availability setups:

  A dual-supervisor setup, where two supervisor modules are installed in a chassis, one being the active and the other, the standby.

  All licensing information, such as trust codes, SLAC, RUM reports, are stored on the active supervisor (active product instance) and synchronised with the standby.

Note You cannot use the HSECK9 key in any other High Availability setup. For example, it is not supported in a Cisco Stackwise Virtual setup and in a Quad-Supervisor setup (Quad-Supervisor with Route Processor Redundancy).

- The number of HSECK9 keys required in a High Availability setup:

  The HSECK9 key is tied to the chassis UDI and regardless of the number of supervisors installed, only one HSECK9 key is required for each chassis UDI. The following sample output shows you how the chassis UDI is displayed. The same chassis UDI is also displayed for the active and standby:

```
Device# show license udi
UDI: PID:C9606R,SN:FXS241201WP <<< chassis UDI

HA UDI List:
    Active:PID:C9606R,SN:FXS241201WP
    Standby:PID:C9606R,SN:FXS241201WP
```

- The number of SLACs required in a High Availability setup:

  Each HSECK9 key requires one SLAC.

  The following sample output shows you how SLAC information is displayed. Because they have the same UDI, note how the same SLAC confirmation code is displayed for all connected devices. Also note the `Total available count`, for HSECK9 key - only one is required for each chassis.

```
Device# show license authorization
Overall status:
  Active: PID:C9606R,SN:FXS241201WP
      Status: SMART AUTHORIZATION INSTALLED on Dec 13 05:18:07 2021 UTC
      Last Confirmation code: 7cf1f54a
  Standby: PID:C9606R,SN:FXS241201WP
      Status: SMART AUTHORIZATION INSTALLED on Dec 13 05:18:07 2021 UTC
      Last Confirmation code: 7cf1f54a

Authorizations:
  C9K HSEC (Cat9K HSEC):
    Description: HSEC Key for Export Compliance on Cat9K Series Switches
    Total available count: 1

<output truncated>
```

- Behavior in the event of a switchover:

  The system continues uninterrupted operation of the cryptographic feature in case of a switchover.

  Because the HSECK9 key is tied to the *chassis* UDI and not a supervisor module, and because licensing information on the active is synchronized with the standby, a switchover can never result in an interruption in the operation of the cryptographic feature.

- Hardware removal and replacement in High Availability setup:

  See Hardware Removal and Replacement, on page 68.

Hardware Removal and Replacement

The following constitues the basis of what you must consider when removing and replacing a supervisor module or linecard:

- The HSECK9 key is tied to the chassis.
- Licensing information is saved on the active product instance (active supervisor module). In a High Availability setup, licensing information is synchronized with the standby.
- The cryptographic feature is configured in interface configuration mode. It corresponds with the line card slot where a linecard supporting the cryptographic feature is installed.

The above principles have the following implications when you remove and replace a supervisor module or a linecard:

- In a single supervisor set-up, if you remove the active supervisor module and replace it with another one, you must install SLAC again.

  If you remove and reinstall the *same* supervisor module, you do not have to reinstall SLAC.

- In a dual supervisor set-up, remove and replace one supervisor module at-a-time. You can start with the active followed by the standby or vice versa. Removing and replacing supervisor modules one at-a-time enables the required licensing information to be retained on the device at all times. It also ensures the operation of the cryptographic feature without any interruptions. If you remove both supervisor modules simultaneously and replace them with other supervisor modules, required licensing information will no longer be available on the device, and you will have to install SLAC again.

  If you remove and reinstall the *same* supervisor module, you do not have to reinstall SLAC.

- You can remove and replace a linecard without any interruptions in the operation of the cryptographic functionality, as long as the replacement line card is installed in the *same line card slot*.

  If you remove a linecard where cryptographic functionality is configured and install the replacement linecard in a different slot, you may have to reconfigure the cryptographic feature.

For information about the removal and replacement procedures, refer to the Cisco Catalyst 9600 Series Supervisor Engine Installation Note and Cisco Catalyst 9600 Series Line Card Installation Note as required.

How to Configure Available Licenses

This section provides information about how to configure available licenses.

Configuring Base and Add-On Licenses

After you order and purchase a base or add-on license, you must configure the license on the device before you can use it.

This task sets a license level and requires a reload before the configured changes are effective. You can use this task to

- Change the current license.
- Add another license. For example, if you are currently using Network Advantage and you also want to use features available with the corresponding Digital Networking Architecture (DNA) Advantage license.
- Remove a license.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**license boot level** { **network-advantage** [**addon dna-advantage**] | **network-essentials** [**addon dna-essentials**] } **Example:** `Device(config)# license boot level network-advantage add-on dna-advantage`	Activates the configured license on the product instance. • **network-advantage** [**addon dna-advantage**]: Configures the Network Advantage license. Optionally, you can also configure the Digital Networking Architecture (DNA) Advantage license. • **network-advantage** [**addon dna-advantage**]: Configures the Network Essentials license. Optionally, you can also configure the Digital Networking Architecture (DNA) Essentials license. In the accompanying example, the DNA Advantage license will be activated on the product instance after reload.
Step 4	**exit** **Example:** `Device(config)# exit`	Returns to the privileged EXEC mode.
Step 5	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	Saves changes in the configuration file.

Step 6	**show version**
	Example:
	`Device# show version`
	
	`<output truncated>`
	`Technology Package License Information:`
	
	`------------------------------------------------`
	`Technology-package`
	`Technology-package`
	`Current Type`
	`Next reboot`
	`------------------------------------------------`
	`network-advantage Smart License`
	`network-advantage`
	`Subscription Smart License dna-advantage`
	
	`<output truncated>`
		Shows currently configured license information and the license that is applicable after reload.
		The "`Technology-package Next reboot`" column displays the change in the configured license that is effective after reload, only if you save the configuration change.
		In the accompanying example, the current license level is Network Advantage. Because the configuration change was saved, the "`Technology-package Next reboot`" column shows that the DNA Advantage license will be activated after reload.

	Command or Action	Purpose
Step 7	**reload** **Example:** `Device# reload`	Reloads the device.

Step 8	**show version**
	Example:
	`Device# show version`
	
	`<output truncated>`
	`Technology Package License Information:`
	
	`------------------------------------------------`
	`Technology-package`
	`Technology-package`
	`Current Type`
	`Next reboot`
	`------------------------------------------------`
	`network-advantage Smart License`
	`network-advantage`
	`dna-advantage Subscription Smart License dna-advantage`
	
	`<output truncated>`
		Shows currently configured license information and the license that is applicable after reload.

What to do next

After you configure a license level, the change is effective after a reload. To know if reporting is required, you can wait for a system message or refer to the policy-using show commands.

- The system message, which indicates that reporting is required: `%SMART_LIC-6-REPORTING_REQUIRED: A Usage report acknowledgment will be required in [dec] days.`

  `[dec]` is the amount of time (in days) left to meet reporting requirements.

- If using **show** commands, refer to the output of the **show license status** privileged EXEC command and check the `Next ACK deadline` field. This means a RUM report must be sent and the ACK must be installed by this date.

The method that you can use to send the RUM report, depends on the topology you have implemented. Refer to the workflow for the applicable topology in theHow to Configure Smart Licensing Using Policy: Workflows by Topology , on page 123 section of the *Smart Licensing Using Policy* chapter in this guide.

Installing SLAC for an HSECK9 Key

This section shows you the various methods of installing SLAC for an HSECK9 key. Each method corresponds with a particular topology in the Smart Licensing Using Policy environment.

For information about all the supported topologies, see the Supported Topologies, on page 105 section of the *Smart Licensing Using Policy* chapter in this guide.

Note The only topology that you *cannot* implement if you want to use an HSECK9 key, is *Connected to CSSM Through a Controller*. The "controller" here is Cisco DNA Center. The Cisco DNA Center GUI does not provide an option to generate a SLAC for Cisco Catalyst switches that support HSECK9.

Installing SLAC: Connected Directly to CSSM

This task shows you how to request and install SLAC when the device (product instance), is directly connected to CSSM.

Before you begin

- Ensure that the device is one that supports HSECK9. See Supported Platforms and Releases, on page 66.
- Ensure you have the required number of HSECK9 keys in the applicable Smart Account and Virtual Account in CSSM.
- Ensure that you have completed Steps from 1 through 3 of the *Connected Directly to CSSM* topology. See Workflow for Topology: Connected Directly to CSSM, on page 125.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**license smart authorization request** {**add** | **replace**} *feature_name* {**all** | **local**} **Example:** `Device# license smart authorization request add hseck9 all`	Requests a SLAC from CSSM or CSLU or SSM On-Prem. • Specify if you want to add to or replace an existing SLAC: • **add**: This adds the requested key to an existing SLAC. The new SLAC will contain all the keys of the existing SLAC, and the requested key. • **replace**: This replaces the existing SLAC. The new SLAC will contain only the requested key. All HSECK9 keys in the existing SLAC are returned. When you enter this keyword, the product instance checks if these existing keys are in-use. If they are, an error message is displayed, telling you to first disable

	Command or Action	Purpose
		the corresponding cryptographic feature. **Note** On Cisco Catalyst 9300X Series Switches in a stacking setup: If you have added a device (where SLAC is not installed) to an existing stack where SLAC is already installed, use the **replace** and **all** keywords. This returns all HSECK9 keys in the existing SLAC and requests SLAC for all the devices in the stack. You cannot request SLAC for a particular member. Your only options are: either the active, or the entire stack. **Note** This keyword is not supported on Cisco Catalyst 9400 Series Supervisor Modules in a Cisco StackWise Virtual set-up. If SLAC is installed only on the active and you want to install it on the standby as well, return the SLAC which is on the active and then request and install SLAC on the active and standby again. • *feature_name*: Enter the name of the export-controlled license for which you want to request an addition or a replacement of the SLAC. Enter "hseck9" to request and install SLAC for the HSECK9 key. • Specify the device by entering one of these options:

	Command or Action	Purpose
		• **all**: Gets the authorization code for *all* devices in a High Availability and stacking set-up. In case of a stacking setup or a Cisco StackWise Virtual setup, we recommend that you use this option and install SLAC for the active and the standby. This ensures uninterrupted use of the cryptographic feature in the event of a switchover. • **local**: Gets the authorization code for the *active* device in a High Availability and stacking set-up. This is the default option.
Step 3	(Optional) **license smart sync** {**all** | **local**} **Example:** `Device# license smart sync all`	Triggers the product instance to synchronize with CSSM, or CSLU, or SSM On-Prem, to send and receive any pending data. This step applies only to topologies where the product instance is connected to CSSM, or CSLU or SSM On-Prem, and where the product instance initiates communication. The topologies are: *Connected Directly to CSSM*, *Connected to CSSM Through CSLU* (product instance-initiated), and SSM On-Prem Deployment (product instance-initiated). By triggering an on-demand synchronization, you can ensure that the SLAC installation process is completed soon after you request SLAC. Otherwise, SLAC is applied to the product instance only the next time the product instance is *scheduled* to contact CSSM, or CSLU or SSM On-Prem.

What to do next

Required Tasks After Installing SLAC, on page 87

Installing SLAC: No Connectivity to CSSM and No CSLU

This task shows you how to request and install SLAC in an air-gapped network, where a device (product instance) cannot communicate online, with anything outside its network.

Here you generate and save the SLAC request to a file, upload it to the CSSM Web UI, download the SLAC code from the CSSM Web UI, and finaly, install it on the product instance.

Before you begin

- Ensure that the device is one that supports HSECK9. See Supported Platforms and Releases, on page 66.
- Ensure you have the required number of HSECK9 keys in the applicable Smart Account and Virtual Account in CSSM.
- Ensure that you have completed Step 1 of the *No Connectivity to CSSM and No CSLU* topology. See Workflow for Topology: No Connectivity to CSSM and No CSLU, on page 131.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**license smart authorization request** {**add** | **replace**} *feature_name* {**all**| **local**} **Example:** `Device# license smart authorization request add hseck9 all`	Generates a SLAC request with all the required information. Specify if you want to add to or replace an existing SLAC: • **add**: Adds the requested key to an existing SLAC. The new authorization code will contain all the keys of the existing SLAC, and the requested license. • **replace**: Replaces the existing SLAC. The new SLAC will contain only the requested HSECK9 key. All keys in the existing SLAC are returned. When you enter this keyword, the product instance checks if these existing keys are in-use. If they are, an error message is displayed, telling you to first disable the corresponding feature.

	Command or Action	Purpose
		Note For a stacking scenario (Cisco Catalyst 9300X Series Switches): If you have added a device (where SLAC is not installed) to an existing stack where SLAC is already installed, use the **replace** and **all** keywords. This returns all HSECK9 keys in the existing SLAC and requests SLAC for all the devices in the stack. You cannot request SLAC for a particular member. Your only options are: either the active, or the entire stack. **Note** This keyword is not supported on Cisco Catalyst 9400 Series Supervisor Modules in a Cisco StackWise Virtual set-up. If SLAC is installed only on the active and you want to install it on the standby as well, return the SLAC which is on the active and then request and install SLAC on the active and standby again. For *feature_name*, enter the name of the export-controlled license for which you want to request an addition or a replacement of the SLAC. Enter "hseck9" to request and install SLAC for the HSECK9 key. Specify the device by entering one of these options: • **all**: Gets the SLAC for *all* devices in a High Availability set-up In case of a stacking setup or a Cisco StackWise Virtual setup, we recommend that you use this option and install SLAC for the active and the standby. This ensures uninterrupted use of the cryptographic feature in the event of a switchover.

	Command or Action	Purpose
		• **local**: Gets the SLAC for the *active* device in a High Availability set-up. This is the default option.
Step 3	**license smart authorization request save***path* **Example:** `Device# license smart authorization request save bootflash:slac.txt`	Saves the required UDI information for the SLAC request in a .txt file, in the specified location.
Step 4	Uploading Data or Requests to CSSM and Downloading a File, on page 210	This task is performed on the CSSM Web UI. **Note** This provision to upload a SLAC *request* file and to then download a SLAC file is supported starting with Cisco IOS XE Cupertino 17.7.1 only. With earlier releases, you have to enter the required information in the CSSM Web UI, generate a SLAC code in the CSSM Web UI, and then download and install it. The older method continues to be available, but the new method is prone to fewer manual errors and is the recommended way for this topology.
Step 5	**copy** *source filename* **bootflash:** **Example:** `Device# copy tftp://10.8.0.6/user01/example.txt bootflash:`	(Optional) Copies the file from its source location or directory to the flash memory of the product instance. You can also import the file *directly* from a remote location and install it on the product instance (next step). • *source*: This is the source location of file. The source can be either local or remote. • **bootflash:**: This is the destination for boot flash memory.
Step 6	**license smart import** *filepath_filename* **Example:** `Device# license smart import bootflash:example.txt`	Imports and installs the file on the product instance. For *filepath_filename*, specify the location, including the filename. After installation, a system message displays the type of file you installed.

What to do next

Required Tasks After Installing SLAC, on page 87

Installing SLAC: Connected to CSSM Through CSLU (Product Instance-Initiated)

This task shows you how to request and install SLAC when the device (product instance) is connected to CSSM through CSLU and the product instance initiates communication, that is, the product instance is configured to *push* the required information to CSLU.

Before you begin

- Ensure that the device is one that supports the HSECK9 key. See Supported Platforms and Releases, on page 66.
- Ensure you have the required number of HSECK9 keys in the applicable Smart Account and Virtual Account in CSSM.
- Ensure that you have completed Steps 1 through 3 of the *Connected to CSSM Through CSLU* (Product Instance-Initiated Communication) topology. See Workflow for Topology: Connected to CSSM Through CSLU, on page 123 → Tasks for Product Instance-Initiated Communication, on page 123.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.

Step 2	**license smart authorization request** {**add** | **replace**} *feature_name* {**all** | **local**}
	Example:
	`Device# license smart authorization request add hseck9 all`
		Requests a SLAC from CSSM or CSLU or SSM On-Prem.
		• Specify if you want to add to or replace an existing SLAC:
		  • **add**: This adds the requested key to an existing SLAC. The new SLAC will contain all the keys of the existing SLAC, and the requested key.
		  • **replace**: This replaces the existing SLAC. The new SLAC will contain only the requested key. All HSECK9 keys in the existing SLAC are returned. When you enter this keyword, the product instance checks if these existing keys are in-use. If they are, an error message is displayed, telling you to first disable the corresponding cryptographic feature.

	Command or Action	Purpose
		Note On Cisco Catalyst 9300X Series Switches in a stacking setup: If you have added a device (where SLAC is not installed) to an existing stack where SLAC is already installed, use the **replace** and **all** keywords. This returns all HSECK9 keys in the existing SLAC and requests SLAC for all the devices in the stack. You cannot request SLAC for a particular member. Your only options are: either the active, or the entire stack. **Note** This keyword is not supported on Cisco Catalyst 9400 Series Supervisor Modules in a Cisco StackWise Virtual set-up. If SLAC is installed only on the active and you want to install it on the standby as well, return the SLAC which is on the active and then request and install SLAC on the active and standby again. • *feature_name*: Enter the name of the export-controlled license for which you want to request an addition or a replacement of the SLAC. Enter "hseck9" to request and install SLAC for the HSECK9 key. • Specify the device by entering one of these options:

	Command or Action	Purpose
		• **all**: Gets the authorization code for *all* devices in a High Availability and stacking set-up. In case of a stacking setup or a Cisco StackWise Virtual setup, we recommend that you use this option and install SLAC for the active and the standby. This ensures uninterrupted use of the cryptographic feature in the event of a switchover. • **local**: Gets the authorization code for the *active* device in a High Availability and stacking set-up. This is the default option.
Step 3	(Optional) **license smart sync** {**all** | **local**} **Example:** `Device# license smart sync all`	Triggers the product instance to synchronize with CSSM, or CSLU, or SSM On-Prem, to send and receive any pending data. This step applies only to topologies where the product instance is connected to CSSM, or CSLU or SSM On-Prem, and where the product instance initiates communication. The topologies are: *Connected Directly to CSSM*, *Connected to CSSM Through CSLU* (product instance-initiated), and SSM On-Prem Deployment (product instance-initiated). By triggering an on-demand synchronization, you can ensure that the SLAC installation process is completed soon after you request SLAC. Otherwise, SLAC is applied to the product instance only the next time the product instance is *scheduled* to contact CSSM, or CSLU or SSM On-Prem.

What to do next

Required Tasks After Installing SLAC, on page 87

Installing SLAC: Connected to CSSM Through CSLU (CSLU-Initiated)

This task shows you how to request and install SLAC when the device (product instance) is connected to CSSM through CSLU and where CSLU initiates communication, that is, CSLU is configured to *pull* the required information from the product instance.

This task requires you to configure certain commands on the product instance, certain tasks in the CSSM Web UI, and certain tasks in the CSLU interface.

Before you begin

- Ensure that the device is one that supports the HSECK9 key. See Supported Platforms and Releases, on page 66.
- Ensure you have the required number of HSECK9 keys in the applicable Smart Account and Virtual Account in CSSM.
- Ensure that you have completed Steps 1 through 3 of the *Connected to CSSM Through CSLU* (Product Instance-Initiated Communication) topology. See Workflow for Topology: Connected to CSSM Through CSLU, on page 123 → Tasks for CSLU-Initiated Communication, on page 125.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.

Step 2	**license smart authorization request** {**add** | **replace**} *feature_name* {**all** | **local**}
	Example:
	`Device# license smart authorization request add hseck9 all`
		Requests a SLAC from CSSM or CSLU or SSM On-Prem.
		• Specify if you want to add to or replace an existing SLAC:
		  • **add**: This adds the requested key to an existing SLAC. The new SLAC will contain all the keys of the existing SLAC, and the requested key.
		  • **replace**: This replaces the existing SLAC. The new SLAC will contain only the requested key. All HSECK9 keys in the existing SLAC are returned. When you enter this keyword, the product instance checks if these existing keys are in-use. If they are, an error message is displayed, telling you to first disable the corresponding cryptographic feature.

	Command or Action	Purpose
		Note On Cisco Catalyst 9300X Series Switches in a stacking setup: If you have added a device (where SLAC is not installed) to an existing stack where SLAC is already installed, use the **replace** and **all** keywords. This returns all HSECK9 keys in the existing SLAC and requests SLAC for all the devices in the stack. You cannot request SLAC for a particular member. Your only options are: either the active, or the entire stack. **Note** This keyword is not supported on Cisco Catalyst 9400 Series Supervisor Modules in a Cisco StackWise Virtual set-up. If SLAC is installed only on the active and you want to install it on the standby as well, return the SLAC which is on the active and then request and install SLAC on the active and standby again. • *feature_name*: Enter the name of the export-controlled license for which you want to request an addition or a replacement of the SLAC. Enter "hseck9" to request and install SLAC for the HSECK9 key. • Specify the device by entering one of these options:

	Command or Action	Purpose
		• **all**: Gets the authorization code for *all* devices in a High Availability and stacking set-up. In case of a stacking setup or a Cisco StackWise Virtual setup, we recommend that you use this option and install SLAC for the active and the standby. This ensures uninterrupted use of the cryptographic feature in the event of a switchover. • **local**: Gets the authorization code for the *active* device in a High Availability and stacking set-up. This is the default option.
Step 3	Requesting SLAC for One or More Product Instance (CSLU Interface), on page 171	This task is performed on the CSLU interface.
Step 4	Generating and Downloading SLAC from CSSM to a File, on page 199	This task is performed on the CSSM Web UI.
Step 5	Import from CSSM (CSLU Interface), on page 166	This task is performed on the CSLU interface. After you have completed it, the uploaded codes are applied to the product instances the next time CSLU runs an update.

What to do next

Required Tasks After Installing SLAC, on page 87

Installing SLAC: SSM On-Prem Deployment (Product Instance-Initiated)

This task shows you how to request and install SLAC when the device (product instance) is connected to SSM On-Prem and where the product instance initiates communication, that is, the product instance is configured to *push* the required information to SSM On-Prem.

Here you first create a request file in SSM On-Prem, upload the request in the CSSM Web UI, generate SLAC, import the SLAC into the SSM On-Prem server. Finally configure the commands on the product instance to request and install SLAC.

Before you begin

- Ensure that the device is one that supports the HSECK9 key. See Supported Platforms and Releases, on page 66.
- Ensure you have the required number of HSECK9 keys in the applicable Smart Account and Virtual Account in CSSM.

- Ensure that you have completed Steps 1 through 3 c. of the *SSM On-Prem Deployment* (Product Instance-Initiated) topology. See Workflow for Topology: SSM On-Prem Deployment, on page 132 → Tasks for Product Instance-Initiated Communication, on page 132.

Procedure

	Command or Action	Purpose
Step 1	Submitting an Authorization Code Request (SSM On-Prem UI), on page 190	This task is performed on the SSM On-Prem UI.
Step 2	Generating and Downloading SLAC from CSSM to a File, on page 199	This task is performed on the CSSM Web UI.
Step 3	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 4	**license smart authorization request** {**add** | **replace**} *feature_name* {**all** | **local**} **Example:** `Device# license smart authorization request add hseck9 all`	Requests a SLAC from CSSM or CSLU or SSM On-Prem. • Specify if you want to add to or replace an existing SLAC: • **add**: This adds the requested key to an existing SLAC. The new SLAC will contain all the keys of the existing SLAC, and the requested key. • **replace**: This replaces the existing SLAC. The new SLAC will contain only the requested key. All HSECK9 keys in the existing SLAC are returned. When you enter this keyword, the product instance checks if these existing keys are in-use. If they are, an error message is displayed, telling you to first disable the corresponding cryptographic feature.

	Command or Action	Purpose
		Note On Cisco Catalyst 9300X Series Switches in a stacking setup: If you have added a device (where SLAC is not installed) to an existing stack where SLAC is already installed, use the **replace** and **all** keywords. This returns all HSECK9 keys in the existing SLAC and requests SLAC for all the devices in the stack. You cannot request SLAC for a particular member. Your only options are: either the active, or the entire stack. **Note** This keyword is not supported on Cisco Catalyst 9400 Series Supervisor Modules in a Cisco StackWise Virtual set-up. If SLAC is installed only on the active and you want to install it on the standby as well, return the SLAC which is on the active and then request and install SLAC on the active and standby again. • *feature_name*: Enter the name of the export-controlled license for which you want to request an addition or a replacement of the SLAC. Enter "hseck9" to request and install SLAC for the HSECK9 key. • Specify the device by entering one of these options:

	Command or Action	Purpose
		• **all**: Gets the authorization code for *all* devices in a High Availability and stacking set-up. In case of a stacking setup or a Cisco StackWise Virtual setup, we recommend that you use this option and install SLAC for the active and the standby. This ensures uninterrupted use of the cryptographic feature in the event of a switchover. • **local**: Gets the authorization code for the *active* device in a High Availability and stacking set-up. This is the default option.
Step 5	(Optional) **license smart sync** {**all** | **local**} **Example:** `Device# license smart sync all`	Triggers the product instance to synchronize with CSSM, or CSLU, or SSM On-Prem, to send and receive any pending data. This step applies only to topologies where the product instance is connected to CSSM, or CSLU or SSM On-Prem, and where the product instance initiates communication. The topologies are: *Connected Directly to CSSM*, *Connected to CSSM Through CSLU* (product instance-initiated), and SSM On-Prem Deployment (product instance-initiated). By triggering an on-demand synchronization, you can ensure that the SLAC installation process is completed soon after you request SLAC. Otherwise, SLAC is applied to the product instance only the next time the product instance is *scheduled* to contact CSSM, or CSLU or SSM On-Prem.

What to do next

Required Tasks After Installing SLAC, on page 87

Installing SLAC: SSM On-Prem Deployment (SSM On-Prem-Initiated)

This task shows you how to request and install SLAC when the device (product instance), is connected to SSM On-Prem and where SSM On-Prem initiates communication, that is, SSM On-Prem is configured to *pull* the required information from the product instance.

Here you create a request file in SSM On-Prem, upload the request in the CSSM Web UI, generate SLAC, import it into the SSM On-Prem server. Finally, synchronize SSM On-Prem with the product instance.

Before you begin

- Ensure that the device is one that supports the HSECK9 key. See Supported Platforms and Releases, on page 66.
- Ensure you have the required number of the HSECK9 keys in the applicable Smart Account and Virtual Account in CSSM.
- Ensure that you have completed Steps 1 through 3 a. of the *SSM On-Prem Deployment* (Product Instance-Initiated) topology. See Workflow for Topology: SSM On-Prem Deployment, on page 132 → Tasks for SSM On-Prem Instance-Initiated Communication, on page 134.

Procedure

	Command or Action	Purpose
Step 1	Submitting an Authorization Code Request (SSM On-Prem UI), on page 190.	This task is performed in the SSM On-Prem UI.
Step 2	In the SSM On-Prem UI, navigate to **Reports** > **Synchronisation pull schedule with the devices** > **Synchronise now with the device**.	This step is optional. If you don't synchronize immediately after importing the codes, the uploaded codes are applied to the product instances the next time SSM On-Prem runs an update.

What to do next

Required Tasks After Installing SLAC, on page 87

Required Tasks After Installing SLAC

This task shows you the activities that you must complete after installing SLAC. The information here applies to all methods of installing SLAC.

Procedure

Step 1 Verify SLAC installation and HSECK9 key usage.

- The following system messages are displayed after SLAC installation:
  - `%SMART_LIC-6-AUTHORIZATION_INSTALL_SUCCESS: A new licensing authorization code was successfully installed on: [chars].`

    [chars] is the UDI where the SLAC was installed.
  - `%SMART_LIC-6-EXPORT_CONTROLLED: Usage of export controlled features is allowed for feature hseck9.`
- Check that the output of the **show license authorization** privileged EXEC command displays a timestamp and a last confirmation code.

  In the `Overall Status` section of the output, look for `Status: SMART AUTHORIZATION INSTALLED on <timestamp>` and `Last Confirmation code: <code>`. This means SLAC is installed.

If you have installed SLAC in a High Availability setup, note that the same SLAC installation timestamp and last confirmation code is displayed for all connected devices. In the sample output below, SLAC is installed in a High Availability setup.

- Check that the *usage* count and status for "C9K HSEC" in the output of the **show license summary** privileged EXEC command displays `0` and `NOT IN USE` respectively. This means that the HSECK9 key is available but is not in-use yet.

Example:

```
Device# show license authorization
Overall status:
  Active: PID:C9606R,SN:FXS241201WP
      Status: SMART AUTHORIZATION INSTALLED on Dec 13 05:18:07 2021 UTC
      Last Confirmation code: 7cf1f54a
  Standby: PID:C9606R,SN:FXS241201WP
      Status: SMART AUTHORIZATION INSTALLED on Dec 13 05:18:07 2021 UTC
      Last Confirmation code: 7cf1f54a

Authorizations:
  C9K HSEC (Cat9K HSEC):
    Description: HSEC Key for Export Compliance on Cat9K Series Switches
    Total available count: 1
    Enforcement type: EXPORT RESTRICTED
    Term information:
      Active: PID:C9606R,SN:FXS241201WP
        Authorization type: SMART AUTHORIZATION INSTALLED
        License type: PERPETUAL
          Term Count: 1
      Standby: PID:C9606R,SN:FXS241201WP
        Authorization type: SMART AUTHORIZATION INSTALLED
        License type: PERPETUAL
          Term Count: 1

Purchased Licenses:
  No Purchase Information Available
```

```
Device# show license summary
Account Information:
  Smart Account: Eg-SA As of Oct 07 05:13:33 2021 UTC
  Virtual Account: Eg-VA

License Usage:
  License                 Entitlement Tag               Count Status
  -----------------------------------------------------------------------------
  network-advantage       (C9600-NW-A)                      2 IN USE
  dna-advantage           (C9600-DNA-A)                     1 IN USE
  C9K HSEC                (Cat9K HSEC)                      0 NOT IN USE
```

Step 2 Configure the cryptographic feature.

The following WAN MACsec configuration is for example purposes only. For information about configuring the feature, see the *MACsec Encryption* chapter of the *Security Configuration Guide, Cisco IOS XE <applicable release number> (Catalyst 9600 Switches)*

Example:

```
Device# show module
Chassis Type: C9606R

Mod Ports Card Type                                  Model          Serial No.
---+-----+--------------------------------------+--------------+--------------
2   24   24-Port 40GE / 100GE                     C9600-LC-24C     FDO24300SBD
```

```
3   0    Supervisor 2 Module                          C9600X-SUP-2     FDO24410996
4   0    Supervisor 2 Module                          C9600X-SUP-2     FDO2441090F
5   44   40x10/25/50GE + 2x200GE + 2x400GE            C9600-LC-40YL4CD FDO245106QU

<output truncated>

Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# interface FourHundredGigE5/0/44
Device(config-if)# macsec dot1q-in-clear 1
Device(config-if)#
*Dec  13 05:20:04.221: %SMART_LIC-6-EXPORT_CONTROLLED: Usage of export controlled features
 is
 allowed for feature hseck9
Device(config-if)#

Device# show running-config interface FourHundredGigE5/0/44
Building configuration...

Current configuration : 160 bytes
!
interface FourHundredGigE5/0/44
 no switchport
 no ip address
 macsec dot1q-in-clear 1
 eapol destination-address broadcast-address
 eapol eth-type 876F
end
```

Step 3 Again check HSECK9 key usage.

After you configure the cryptographic feature, the usage count and status of HSECK9 key in the output of the **show license summary** privileged EXEC command changes to `1` and `IN USE`, respectively.

Example:

```
Device# show license summary
Account Information:
  Smart Account: Eg-SA As of Oct 07 05:13:33 2021 UTC
  Virtual Account: Eg-VA

License Usage:
  License                 Entitlement Tag               Count Status
  -----------------------------------------------------------------------------
  network-advantage       (C9600-NW-A)                      2 IN USE
  dna-advantage           (C9600-DNA-A)                     1 IN USE
  C9K HSEC                (Cat9K HSEC)                      1 IN USE
```

Step 4 Check if reporting is required. The method that you can use to send the RUM report, depends on the topology you have implemented. Refer to the workflow for the applicable topology in the How to Configure Smart Licensing Using Policy: Workflows by Topology , on page 123 section of the *Smart Licensing Using Policy* chapter in this guide.

To know if reporting is required, you can wait for a system message or refer to the policy using **show** commands.

- The system message, which indicates that reporting is required: `%SMART_LIC-6-REPORTING_REQUIRED: A Usage report acknowledgement will be required in [dec] days.`

  `[dec]` is the amount of time (in days) left to meet reporting requirements.

- If using **show** commands, refer to the output of the **show license status** privileged EXEC command. Check the `Next ACK deadline` field. You must send the RUM report and ensure that the ACK is installed by this date.

Returning a SLAC

This task shows you how to return a SLAC and return the HSECK9 key to your license pool in CSSM. You can use this task with all topologies.

You may want to return a SLAC and HSECK9 key under these circumstances:

- You no longer want to use the cryptographic feature, which requires an HSECK9 key.
- You want to return the device for Return Material Authorization (RMA), or decommission it permanently. When you return a device to Cisco, you have to configure the **licence smart factory reset** privileged EXEC command, which removes all licensing information (except the licenses in-use) from the product instance, including any authorization codes, RUM reports and so on. *Before* you perform a factory reset, return the SLAC code. We also recommend that you send a RUM report to CSSM before removing licensing information from the product instance.

Before you begin

Disable or unconfigure the cryptographic feature for which you used the HSECK9 key.

Procedure

	Command or Action	Purpose
Step 1	Disable or unconfigure the cryptographic feature for which you used the HSECK9 key.	For information about disabling the WANMACsec feature, see the *MACsec Encryption* chapter of the *Security Configuration Guide, Cisco IOS XE <applicable release number> (Catalyst 9600 Switches)* If the cryptographic feature you are disabling is the WAN MACsec feature, note the following: Even after disabling the cryptographic feature, the output of the **show license summary** command displays the usage count and status for the HSECK9 key as `1` and `IN USE`. This is as expected. The steps in this task show you how to *release* the key, which changes the count and status to `0` and `NOT IN USE`. But you must disable the WAN MACsec feature before you try to release the HSECK9 key.
Step 2	**enable** **Example:**	Enables privileged EXEC mode. Enter your password, if prompted.

	Command or Action	Purpose
	`Device> enable`	

Step 3	**show license summary**
	Example:
	`Device# show license summary`
	`Account Information:`
	`  Smart Account: Eg-SA As of Oct 07 05:13:33 2021 UTC`
	`  Virtual Account: Eg-VA`
	
	`License Usage:`
	`  License  Entitlement Tag  Count Status`
	`---`
	`  network-advantage  (C9600-NW-A)  2 IN USE`
	`  dna-advantage  (C9600-DNA-A)  1 IN USE`
	`  C9K HSEC  (Cat9K HSEC)  1 IN USE`
		(Optional) Displays license usage summary. This step applies only if you are returning a SLAC.
		If the status of the HSECK9 key is displayed as `NOT IN USE` skip to Step 5.
		If the status of the HSECK9 key is displayed as `IN USE` even after the cryptographic feature is disabled, then perform the next step. This is the case in the accompanying example.

Step 4	**platform wanmacsec hsec-license-release**
	Example:
	`Device# configure terminal`
	`Device(config)# platform wanmacsec hsec-license-release`
	`HSEC license is released`
	`Device(config)# exit`
		Enters the global configuration mode, releases the HSECK9 license, and returns to privileged EXEC mode.

Step 5	**show license summary**
	Example:
	`Device# show license summary`
	`Account Information:`
	`  Smart Account: Eg-SA As of Oct 07 05:13:33 2021 UTC`
	`  Virtual Account: Eg-VA`
	
	`License Usage:`
	`  License  Entitlement Tag  Count Status`
	`---`
	`  network-advantage  (C9600-NW-A)  2 IN USE`
	`  dna-advantage  (C9600-DNA-A)  1 IN USE`
	`  C9K HSEC  (Cat9K HSEC)  0 NOT IN USE`
		(Optional) Displays license usage summary. This step applies only if you are returning a SLAC.
		Ensure that the status of the license that you want to return is `NOT IN USE`.

Step 6	**license smart authorization return** {**all** |**local**} {**offline** [*path*] |**online**} **Example:**	Returns an authorization code back to the license pool in CSSM. A return code is displayed after you enter this command. Specify the product instance:

	Command or Action	Purpose

	`Device# license smart authorization return all online`
	
	`OR`
	
	`Device# license smart authorization return all offline`
	`Enter this return code in Cisco Smart Software Manager portal:`
	
	`UDI: PID:C9606R,SN:FXS241201WP`
	`Return code:`
	`Cr9JHx-L1x5Rj-ftwzg1-h9QZAU-LE5DT1-babWeL-FABPt9-Wr1Dn7-Rp7`
	
	`OR`
	
	`Device# license smart authorization return all`
	`offline bootflash:return-code.txt`
		• **all**: Performs the action for all connected product instances in a High Availability or stacking set-up.
		
		• **local**: Performs the action for the active product instance. This is the default option.
		
		Specify if you are connected to CSSM or not:
		
		• If connected to CSSM, or if you have implemented a topology where the product instance-initiates communication (CSLU or SSM On-Prem), enter **online**. The code is automatically returned to CSSM and a confirmation is returned and installed on the product instance. If you choose this option, the return code is automatically submitted to CSSM.
		
		• If not connected to CSSM, or if you have implemented a topology with CSLU-initiated or SSM On-Prem initiated communication, enter **offline** [*filepath_filename*].
		
		  • If you enter only the **offline** keyword, copy the return code that is displayed on the CLI and enter it in the CSSM Web UI.
		
		  Complete this task to enter the return code in the CSSM Web UI: Entering a SLAC Return Code in CSSM and Removing a Product Instance, on page 205.
		
		  • If you save the return code to a file, upload the file to CSSM Web UI.
		
		  For example: `Device# license smart authorization return local offline bootflash:return-code.txt`
		
		  Note This method of returning SLAC is supported starting with Cisco IOS XE Cupertino 17.7.1 only.
		
		  Complete this task to upload the return request in the CSSM Web UI:

	Command or Action	Purpose
		Uploading Data or Requests to CSSM and Downloading a File, on page 210.

Step 7	**show license authorization**
	Example:
	`Device# show license authorization`
	`Overall status:`
	`  Active: PID:C9606R,SN:FXS241201WP`
	`  Status: NOT INSTALLED`
	`  Last return code:`
	`Cr9JHx-L1x5Rj-ftwzg1-h9QZAU-LE5DT1-`
	`babWeL-FABPt9-Wr1Dn7-Rp7`
	`  Standby: PID:C9606R,SN:FXS241201WP`
	`  Status: NOT INSTALLED`
	`  Last return code:`
	`Cr9JHx-L1x5Rj-ftwzg1-h9QZAU-LE5DT1-`
	`babWeL-FABPt9-Wr1Dn7-Rp7`
	
	`<output truncated>`
		Displays licensing information. Check under the `License Authorizations` header in the output. If the return process is completed correctly, the `Last return code:` field displays the return code.

Feature History for Available Licenses

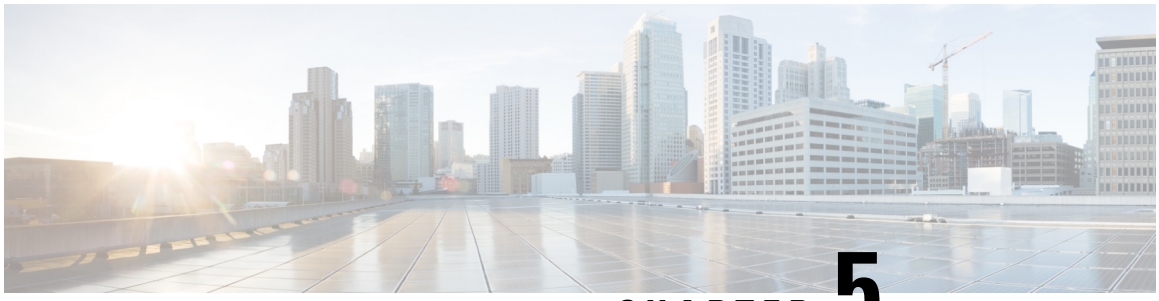
This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.11.1	Base and Add-On Licenses	This feature was introduced. The software features available on Cisco Catalyst 9600 Series Switches fall under base and add-on license levels. See Base and Add-On Licenses, on page 65 and Configuring Base and Add-On Licenses, on page 69.
Cisco IOS XE Cupertino 17.7.1	Base and Add-On Licenses	This feature was implemented on Cisco Catalyst 9600 Series Supervisor Engine 2 (C9600X-SUP-2), which was introduced in this release. See Base and Add-On Licenses, on page 65 and Configuring Base and Add-On Licenses, on page 69.

Release	Feature	Feature Information
Cisco IOS XE Cupertino 17.8.1	Export Control Key for High Security (HSECK9)	Support for the HSECK9 key was introduced on the Cisco Catalyst 9600 Series Supervisor Engine 2 (C9600X-SUP-2) and associated line cards. The HSECK9 key is an export-controlled license, which authorizes the use of cryptographic features that are restricted by U.S. export control laws. If you want to use a restricted cryptographic feature, an HSECK9 key is required. See Export Control Key for High Security, on page 66 and Installing SLAC for an HSECK9 Key, on page 71.

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com.



CHAPTER 5

Smart Licensing Using Policy

Introduction to Smart Licensing Using Policy

Smart Licensing Using Policy is an enhanced version of Smart Licensing, with the overarching objective of providing a licensing solution that does not interrupt the operations of your network, rather, one that enables a compliance relationship to account for the hardware and software licenses you purchase and use.

Smart Licensing Using Policy is supported starting with Cisco IOS XE Amsterdam 17.3.2a.

The primary benefits of this enhanced licensing model are:

- Seamless day-0 operations

  After a license is ordered, no preliminary steps, such as registration or generation of keys etc., are required unless you use an export-controlled or enforced license. Only these licenses require authorization *before* use. For all other licenses, product features can be configured on the device right-away.

- Consistency in Cisco IOS XE

  Campus and industrial ethernet switching, routing, and wireless devices that run Cisco IOS XE software, have a uniform licensing experience.

- Visibility and manageability

  Tools, telemetry and product tagging, to know what is in-use.

- Flexible, time series reporting to remain compliant

  Easy reporting options are available, whether you are directly or indirectly connected to Cisco Smart Software Manager (CSSM), or in an air-gapped network.

This document provides conceptual, configuration, and troubleshooting information for Smart Licensing Using Policy on Cisco Catalyst Access, Core, and Aggregation Switches.

For a more detailed overview on Cisco Licensing, go to cisco.com/go/licensingguide.

Information About Smart Licensing Using Policy

This section provides information about the components that can be part of your implementation of Smart Licensing Using Policy, the key concepts associated with the feature, the supported products, overviews of all supported topologies (the different ways in which you can implement the feature), and how Smart Licensing Using Policy interacts with other features.

Overview

Smart Licensing Using Policy is a software license management solution that provides a seamless experience with the various aspects of licensing. The following summarizes how you operate in this environement:

- Purchase licenses: Purchase licenses through the existing channels and use the Cisco Smart Software Manager (CSSM) portal to view product instances and licenses.

Note For new hardware or software orders, Cisco simplifies the implementation of Smart Licensing Using Policy, by factory-installing the following (terms are explained in the Concepts, on page 100 section further below):

- A custom policy, if available.
- An authorization code, if applicable. For this, you must provide your Smart Account and Virtual Account information when placing the order.
- A trust code, which ensures authenticity of data sent to CSSM. This is installed starting with Cisco IOS XE Cupertino 17.7.1. This trust code cannot be used to *communicate* with CSSM.

- Use: Most licenses are unenforced. This means that you do not have to complete any licensing-specific operations, such as registering or generating keys before you start using the software and the licenses that are tied to it. Only export-controlled and enforced licenses require Cisco authorization *before* use and only certain products support an export-controlled license. License usage is recorded on your device with timestamps and the required workflows can be completed at a later date
- Report license usage to CSSM: Multiple options are available for license usage reporting. You can use the Cisco Smart Licensing Utility (CSLU), report usage information directly to CSSM, use a Controller (like Cisco DNA Center), deploy Smart Software Manager On-Prem (SSM On-Prem) to administer products and licenses on your premises. The usage report is in plain text XML format. See: Sample Resource Utilization Measurement Report, on page 219.
- Reconcile: For situations where delta billing applies (purchased versus consumed).

Supported Products

This section provides information about the Cisco IOS-XE product instances that are within the scope of this document and support Smart Licensing Using Policy. All models (Product IDs or PIDs) in a product series are supported – unless indicated otherwise.

Table 5: Supported Product Instances: Cisco Catalyst Access, Core, and Aggregation Switches

Cisco Catalyst Access, Core, and Aggregation Switches	When Support was Introduced
Cisco Catalyst 9200 Series Switches	Cisco IOS XE Amsterdam 17.3.2a
Cisco Catalyst 9300 Series Switches	Cisco IOS XE Amsterdam 17.3.2a
Cisco Catalyst 9400 Series Switches	Cisco IOS XE Amsterdam 17.3.2a
Cisco Catalyst 9500 Series Switches	Cisco IOS XE Amsterdam 17.3.2a
Cisco Catalyst 9600 Series Switches	Cisco IOS XE Amsterdam 17.3.2a

Architecture

This section explains the various components that can be part of your implementation of Smart Licensing Using Policy.

Product Instance

A product instance is a single instance of a Cisco product, identified by a Unique Device Identifier (UDI).

A product instance records and reports license usage (RUM reports), and provides alerts and system messages about overdue reports, communication failures, etc. RUM reports and usage data are securely stored in the product instance.

Throughout this document, the term *product instance* refers to all supported physical and virtual product instances - unless noted otherwise. For information about the product instances that are within the scope of this document, see Supported Products, on page 97.

CSLU

Cisco Smart License Utility (CSLU) is a Windows-based reporting utility that provides aggregate licensing workflows. This utility performs the following key functions:

- Provides options relating to how workflows are triggered. The workflows can be triggered by CSLU or by the product instance.
- Collects usage reports from the product instance and uploads these usage reports to the corresponding Smart Account or Virtual Account – online, or offline, using files. Similarly, the RUM report ACK is collected online, or offline, and sent back to the product instance.
- Sends authorization code requests to CSSM and receives authorization codes from CSSM, if applicable.

CSLU can be part of your implementation in the following ways:

- Install the windows application, to use CSLU as a standalone tool that is connected to CSSM.
- Install the windows application, to use CSLU as a standalone tool that is disconnected from CSSM. With this option, the required usage information is downloaded to a file and then uploaded to CSSM. This is suited to air-gapped networks.
- Deploy CSLU on a machine (laptop or desktop) running Linux.
- Embedded (by Cisco) in a controller such as Cisco DNA Center.

CSLU supports Windows 10 and Linux operating systems. We recommend that you always use the latest version of CSLU that is available. For the release notes and to download the latest version, click *Smart Licensing Utility* on the Software Download page.

CSSM

Cisco Smart Software Manager (CSSM) is a portal that enables you to manage all your Cisco software licenses from a centralized location. CSSM helps you manage current requirements and review usage trends to plan for future license requirements.

You can access the CSSM Web UI at https://software.cisco.com. Under the **Smart Software Manager** click the **Manage Licenses** link.

The Supported Topologies, on page 105 in this document explains the different ways in which you can connect to CSSM.

In CSSM you can:

- Create, manage, or view virtual accounts.
- Transfer licenses between virtual accounts or view licenses.
- Transfer, remove, or view product instances.
- Run reports against your virtual accounts.
- Modify your email notification settings.
- View overall account information.

Controller

A management application or service that manages multiple product instances.

On Cisco Catalyst Access, Core, and Aggregation Switches, Cisco DNA Center is the supported controller. Information about the controller, product instances that support the controller, and minimum required software versions on the controller and on the product instance is provided below:

Table 6: Support Information for Controller: Cisco DNA Center

Minimum Required Cisco DNA Center Version for Smart Licensing Using Policy[2]	Minimum Required Cisco IOS XE Version[3]	Supported Product Instances
Cisco DNA Center Release 2.2.2	Cisco IOS XE Amsterdam 17.3.2a	• Cisco Catalyst 9200 Series Switches • Cisco Catalyst 9300 Series Switches • Cisco Catalyst 9400 Series Switches • Cisco Catalyst 9500 Series Switches • Cisco Catalyst 9600 Series Switches

[2] The minimum required software version on the controller. This means support continues on all subsequent releases - unless noted otherwise

[3] The minimum required software version on the product instance. This means support continues on all subsequent releases - unless noted otherwise.

For more information about Cisco DNA Center, see the support page at: https://www.cisco.com/c/en/us/support/cloud-systems-management/dna-center/series.html.

SSM On-Prem

Smart Software Manager On-Prem (SSM On-Prem) is an asset manager, which works in conjunction with CSSM. It enables you to administer products and licenses on your premises instead of having to directly connect to CSSM.

Information about the required software versions to implement Smart Licensing Using Policy with SSM On-Prem, is provided below:

Minimum Required SSM On-Prem Version for Smart Licensing Using Policy[4]	Minimum Required Cisco IOS XE Version[5]	Supported Product Instances
Version 8, Release 202102	Cisco IOS XE Amsterdam 17.3.3	• Cisco Catalyst 9200 Series Switches • Cisco Catalyst 9300 Series Switches • Cisco Catalyst 9400 Series Switches • Cisco Catalyst 9500 Series Switches • Cisco Catalyst 9600 Series Switches

[4] The minimum required SSM On-Prem version. This means support continues on all subsequent releases - unless noted otherwise

[5] The minimum required software version on the product instance. This means support continues on all subsequent releases - unless noted otherwise.

For more information about SSM On-Prem, see Smart Software Manager On-Prem on the Software Download page. Hover over the .iso image to display the documentation links.

Concepts

This section explains the key concepts of Smart Licensing Using Policy.

License Enforcement Types

A given license belongs to one of three enforcement types. The enforcement type indicates if the license requires authorization before use, or not.

- Unenforced or Not Enforced

  Unenforced licenses *do not* require authorization before use in air-gapped networks, or registration, in connected networks. The terms of use for such licenses are as per the end user license agreement (EULA).

  All licenses available on Cisco Catalyst Access, Core, and Aggregation Switches are unenforced licenses.

- Enforced

  Licenses that belong to this enforcement type require authorization before use. The required authorization is in the form of an authorization code, which must be installed in the corresponding product instance.

  An example of an enforced license is the Media Redundancy Protocol (MRP) Client license, which is available on Cisco's Industrial Ethernet Switches.

- Export-Controlled

  Licenses that belong to this enforcement type are export-restricted by U.S. trade-control laws and require authorization before use. The required authorization code must be installed in the corresponding product instance for these licenses as well. Cisco may pre-install export-controlled licenses when ordered with hardware purchase.

  An example of an export-controlled license is the High Speed Encryption (HSECK9) license, which is available on certain Cisco Routers.

License Duration

This refers to the duration or term for which a purchased license is valid. A given license may belong to any one of the enforcement types mentioned above and be valid for the following durations:

- Perpetual: There is no expiration date for such a license.

  Network Essentials, Network Advantage, and HSECK9 are examples of perpetual licenses.

- Subscription: The license is valid only until a certain date.

  DNA Essentials and DNA Advantage licenses are examples of subscription licenses.

Authorization Code

The Smart Licensing Authorization Code (SLAC) allows activation and continued use of a license that is export-controlled or enforced. The authorization code is installed on the product instance. If an authorization code is required for the license you are using, you can request one from CSSM.

You can remove and return a SLAC to your CSSM license pool. But in order to do this, you must first disable the feature that uses the license. You cannot return a SLAC if it is in-use.

Table 7: Licenses that Require SLAC, Supported Platforms, and Releases

Export-Controlled License or Key Which Requires SLAC	Enforcement Type	Supporting Products and When Support was Introduced
HSECK9	Export-controlled	Cisco Catalyst 9300X Series Switches, starting from Cisco IOS XE Bengaluru 17.6.2.
		Cisco Catalyst 9500X Series Switches, starting from Cisco IOS XE Cupertino 17.8.1.
		Cisco Catalyst 9600 Series 40-Port 50G, 2-Port 200G, 2-Port 400G Line Card (C9600-LC-40YL4CD) with Cisco Catalyst 9600 Series Supervisor Engine 2 (C9600X-SUP-2), starting from Cisco IOS XE Cupertino 17.8.1.
		Cisco Catalyst 9400 Series Supervisor 2 and 2XL Modules (C9400X-SUP-2 and C9400X-SUP-2XL) of the Cisco Catalyst 9400 Series Switches, starting with Cisco IOS XE Dublin 17.11.1.

For detailed information about the HSECK9 key on supported products, see the Export Control Key for High Security, on page 66 section of the *Available Licenses* chapter in this guide.

SLR Authorization Codes

If you are upgrading from an earlier licensing model to Smart Licensing Using Policy, you may have a Specific License Reservation (SLR) with its own authorization code. An SLR authorization code is supported after upgrade to Smart Licensing Using Policy.

Note While existing SLRs are carried over after upgrade, you cannot request a new SLR in the Smart Licensing Using Policy environment, because the notion of "reservation" does not apply. If you are in an air-gapped network, the No Connectivity to CSSM and No CSLU topology applies instead.

For more information about how the SLR authorization code is handled, see Upgrades, on page 118. If you want to return an SLR authorization code, see Returning an Authorization Code, on page 201.

Policy

A policy provides the product instance with these reporting instructions:

- License usage report acknowledgement requirement (Reporting ACK required): The license usage report is known as a RUM Report and the acknowledgement is referred to as an ACK (See RUM Report and Report Acknowledgement). This is a yes or no value which specifies if the report for this product instance requires CSSM acknowledgement or not. The default policy is always set to “yes”.
- First report requirement (days): The first report must be sent within the duration specified here.

  If the value here is zero, no first report is required.
- Reporting frequency (days): The subsequent report must be sent within the duration specified here.

  If the value here is zero, it means no further reporting is required *unless* there is a usage change.
- Report on change (days): In case of a change in license usage, a report must be sent within the duration specified here.

  If the value here is zero, no report is required on usage change.

  If the value here is not zero, reporting *is* required after the change is made. All the scenarios listed below count as changes in license usage on the product instance:
  - Changing licenses consumed (includes changing to a different license, and, adding or removing a license).
  - Going from consuming zero licenses to consuming one or more licenses.
  - Going from consuming one or more licenses to consuming zero licenses.

Note If a product instance has *never* consumed a license, reporting is not required even if the policy has a non-zero value for any of the reporting requirements (First report requirement, Reporting frequency, Report on change).

Understanding Policy Selection

CSSM determines the policy that is applied to a product instance. Only one policy is in use at a given point in time. The policy and its values are based on a number of factors, including the licenses being used.

`Cisco default` is the default policy that is always available in the product instance. If no other policy is applied, the product instance applies this default policy. The table below (Table 8: Policy: Cisco default, on page 103) shows the `Cisco default` policy values.

While you cannot configure a policy, you can request for a customized one, by contacting the Cisco Global Licensing Operations team. Go to Support Case Manager. Click **OPEN NEW CASE** > Select **Software Licensing**. The licensing team will contact you to start the process or for any additional information. Customized policies are also made available through your Smart account in CSSM.

Note To know which policy is applied (the policy in-use) and its reporting requirements, enter the **show license all** command in privileged EXEC mode.

Table 8: Policy: Cisco default

Policy: `Cisco default`	**Default Policy Values**
Export (Perpetual/Subscription) **Note** Applied only to licenses with enforcement type "Export-Controlled".	Reporting ACK required: Yes First report requirement (days): 0 Reporting frequency (days): 0 Report on change (days): 0
Enforced (Perpetual/Subscription) **Note** Applied only to licenses with enforcement type "Enforced".	Reporting ACK required: Yes First report requirement (days): 0 Reporting frequency (days): 0 Report on change (days): 0
Unenforced/Non-Export Perpetual[6]	Reporting ACK required: Yes First report requirement (days): 365 Reporting frequency (days): 0 Report on change (days): 90
Unenforced/Non-Export Subscription	Reporting ACK required: Yes First report requirement (days): 90 Reporting frequency (days): 90 Report on change (days): 90

[6] For Unenforced/Non-Export Perpetual: the default policy's first report requirement (within 365 days) applies only if you have purchased hardware or software from a distributor or partner.

RUM Report and Report Acknowledgement

A Resource Utilization Measurement report (RUM report) is a license usage report, which fulfils reporting requirements as specified by the policy. RUM reports are generated by the product instance and consumed by CSSM. The product instance records license usage information and all license usage changes in an open RUM report. At system-determined intervals, open RUM reports are closed and new RUM reports are opened to continue recording license usage. A closed RUM report is ready to be sent to CSSM.

A RUM acknowledgement (RUM ACK or ACK) is a response from CSSM and provides information about the status of a RUM report. Once the ACK for a report is available on the product instance, it indicates that the corresponding RUM report is no longer required and can be deleted.

The reporting method, that is, how a RUM report is sent to CSSM, depends on the topology you implement.

CSSM displays license usage information as per the last received RUM report.

A RUM report may be accompanied by other requests, such as a trust code request, or a SLAC request. So in addition to the RUM report IDs that have been received, an ACK from CSSM may include authorization codes, trust codes, and policy files.

The policy that is applied to a product instance determines the following aspects of the reporting requirement:

- Whether a RUM report is sent to CSSM and the maximum number of days provided to meet this requirement.
- Whether the RUM report requires an acknowledgement (ACK) from CSSM.
- The maximum number of days provided to report a change in license consumption.

RUM report generation, storage, and management

Starting with Cisco IOS XE Cupertino 17.7.1, RUM report generation and related processes have been optimized and enhanced as follows:

- You can display the list of all available RUM reports on a product instance (how many there are, the processing state each one is in, if there are errors in any of them, and so on). This information is available in the **show license rum** , **show license all** , and **show license tech** privileged EXEC commands. For detailed information about the fields displayed in the output, see the command reference of the corresponding release.
- RUM reports are stored in a new format that reduces processing time, and reduces memory usage. In order to ensure that there are no usage reporting inconsistencies resulting from the difference in the old and new formats, we recommend that you send a RUM report in the method that will apply to your topology, in these situations:

  When you upgrade from an earlier release supporting Smart Licensing Using Policy, to Cisco IOS XE Cupertino 17.7.1 or a later release.

  When you downgrade from Cisco IOS XE Cupertino 17.7.1 or a later release to an earlier release supporting Smart Licensing Using Policy.
- To ensure continued disk space and memory availability, the product instance detects and triggers deletion of RUM reports that are deemed eligible.

Trust Code

A *UDI-tied public key*, which the product instance uses to

- Sign a RUM report. This prevents tampering and ensures data authenticity.
- Enable secure communication with CSSM.

There are multiple ways to obtain a trust code.

- From Cisco IOS XE Cupertino 17.7.1, a trust code is factory-installed for all new orders.

Note A factory-installed trust code cannot be used for *communication* with CSSM.

- A trust code can obtained from CSSM, using an ID token.

  Here you generate an *ID token* in the CSSM Web UI to obtain a trust code and install it on the product instance. You must overwrite the factory-installed trust code if there is one. If a product instance is directly connected to CSSM, use this method to enable the product instance to communicate with CSSM in a secure manner. This method of obtaining a trust code is applicable to all the options of directly connecting to CSSM. For more information, see Connected Directly to CSSM, on page 107.

- From Cisco IOS XE Cupertino 17.7.1, a trust code is automatically obtained in topologies where the product instance initiates the sending of data to CSLU and in topologies where the product instance is in an air-gapped network.

  From Cisco IOS XE Cupertino 17.9.1, a trust code is automatically obtained in topologies where CSLU initiates the retrieval of data from the product instance.

  If there is a factory-installed trust code, it is automatically overwritten. A trust code obtained this way can be used for secure communication with CSSM.

  Refer to the corresponding topology description and workflow to know how the trust code is requested and installed in each scenario Supported Topologies, on page 105.

If a trust code is installed on the product instance, the output of the **show license status** command displays a timestamp in the `Trust Code Installed:` field.

Supported Topologies

This section describes the various ways in which you can implement Smart Licensing Using Policy. For each topology, refer to the accompanying overview to know the how the set-up is designed to work, and refer to the considerations and recommendations, if any.

After Topology Selection

After you have selected a topology, see How to Configure Smart Licensing Using Policy: Workflows by Topology , on page 123. These workflows are only for new deployments. They provide the simplest and fastest way to implement a topology.

If you are migrating from an existing licensing model, see Migrating to Smart Licensing Using Policy, on page 137.

If you want to perform any additional configuration tasks, for instance, if you want to configure a different license, or use an add-on license, or if you want to configure a narrower reporting interval, see the Task Library for Smart Licensing Using Policy, on page 161. Check the "Supported Topologies" where provided, before you proceed.

Connected to CSSM Through CSLU

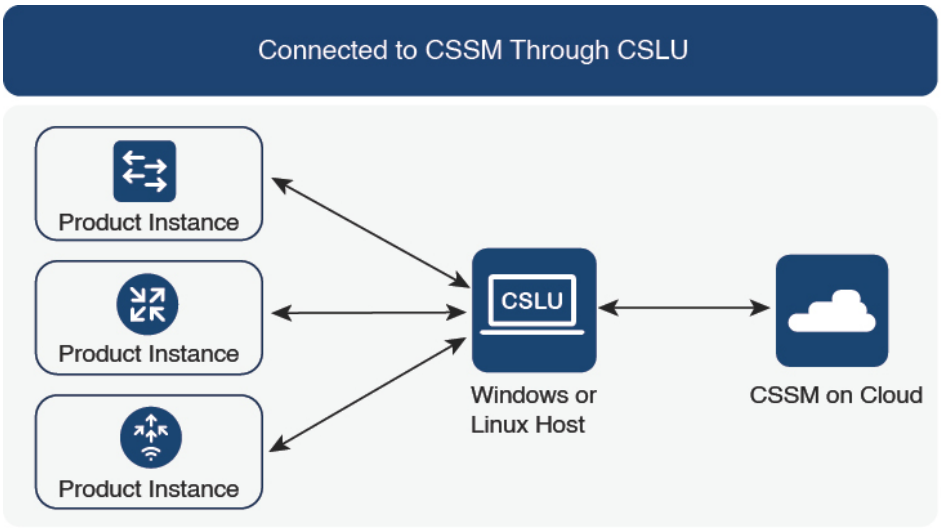
Overview:

Here, product instances in the network are connected to CSLU, and CSLU becomes the single point of interface with CSSM. A product instance can be configured to *push* the required information to CSLU. Alternatively, CSLU can be set-up to *pull* the required information from a product instance at a configurable frequency.

Product instance-initiated communication (push): A product instance initiates communication with CSLU, by connecting to a REST endpoint in CSLU. Data that is sent includes RUM reports and requests for authorization codes, UDI-tied trust codes, and policies. You can configure the product instance to automatically send RUM reports to CSLU at required intervals. This is the default method for a product instance.

CSLU-initiated communication (pull): To initiate the retrieval of information from a product instance, CSLU uses NETCONF, or RESTCONF, or gRPC with YANG models, or native REST APIs, to connect to the product instance. Supported workflows include retrieving RUM reports from the product instance and sending the same to CSSM, authorization code installation, UDI-tied trust code installation, and application of policies.

Figure 3: Topology: Connected to CSSM Through CSLU



Considerations or Recommendations:

Choose the method of communication depending on your network's security policy.

Release-Wise Changes and Enhancements:

This section outlines important release-wise software changes and enhancements that affect this topology.

From Cisco IOS XE Cupertino 17.7.1:

- Trust code request and installation

  If a trust code is not available on the product instance, the product instance detects and automatically includes a request for one, as part of a RUM report. A corresponding ACK from CSSM includes the trust code. If there is an existing factory-installed trust code, it is automatically overwritten. A trust code obtained this way can be used for communication with CSSM.

  This is supported in a standalone, as well as a High Availability set-up. In a High Availability set-up, the active product instance requests the trust code for all connected product instances where a trust code is not available.

  In this release, this enhancement applies only to the product instance-initiated mode.

From Cisco IOS XE Cupertino 17.9.1:

- Trust code request and installation

  From this release, trust code request and installation is supported in the CSLU-initiated mode as well.

- RUM report throttling

  In the product instance-initiated mode, the minimum reporting frequency is throttled to one day. This means the product instance does not send more than one RUM report a day. This resolves the problem of too many RUM reports being generated and sent for certain licenses. It also resolves the memory-related issues and system slow-down caused by an excessive generation of RUM reports.

You can override the throttling restriction by entering the **license smart sync** command in privileged EXEC mode.

RUM report throttling also applies to the Cisco IOS XE Amsterdam 17.3.6 and later releases of the 17.3.x train, and Cisco IOS XE Bengaluru 17.6.4 and later releases of the 17.6.x train. From 17.9.1, RUM report throttling is applicable to *all* subsequent releases.

Where to Go Next:

To implement this topology, see Workflow for Topology: Connected to CSSM Through CSLU, on page 123.

Connected Directly to CSSM

Overview:

This topology is available in the earlier version of Smart Licensing and continues to be supported with Smart Licensing Using Policy.

Here, you establish a *direct* and *trusted* connection from a product instance to CSSM. The direct connection, requires network reachability to CSSM. For the product instance to then exchange messages and communicate with CSSM, configure one of the transport options available with this topology (described below). Lastly, the establishment of trust requires the generation of a token from the corresponding Smart Account and Virtual Account in CSSM, and installation on the product instance.

Note A factory-installed trust code cannot be used for communication with CSSM. This means that for this topology, you must generate an *ID token* in the CSSM Web UI to obtain a trust code and install it on the product instance. You must overwrite the factory-installed trust code if there is one. Also see Trust Code, on page 104.

You can configure a product instance to communicate with CSSM in the following ways:

- Use Smart transport to communicate with CSSM

  Smart transport is a transport method where a Smart Licensing (JSON) message is contained within an HTTPs message, and exchanged between a product instance and CSSM, to communicate. The following Smart transport configuration options are available:

  - Smart transport: In this method, a product instance uses a specific Smart transport licensing server URL. This must be configured exactly as shown in the workflow section.
  - Smart transport through an HTTPs proxy: In this method, a product instance uses a proxy server to communicate with the licensing server, and eventually, CSSM.

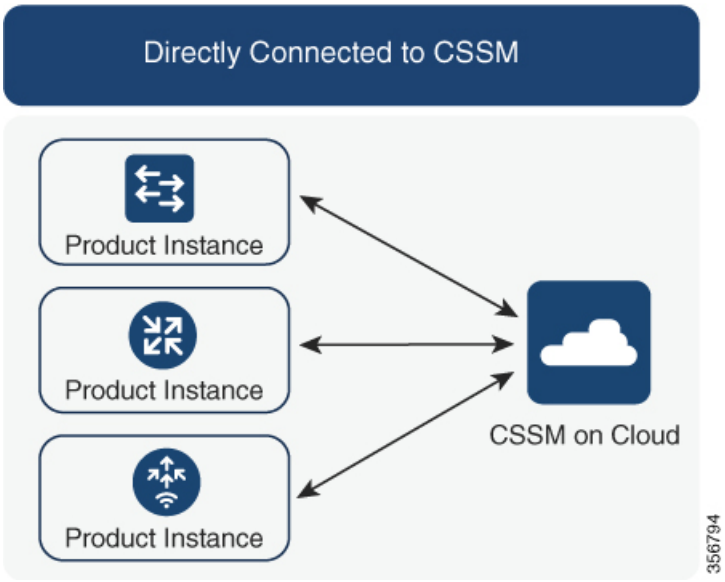
- Use Call Home to communicate with CSSM.

  Call Home provides e-mail-based and web-based notification of critical system events. This method of connecting to CSSM is available in the earlier Smart Licensing environment, and continues to be available with Smart Licensing Using Policy. The following Call Home configuration options are available:

  - Direct cloud access: In this method, a product instance sends usage information directly over the internet to CSSM; no additional components are needed for the connection.

- Direct cloud access through an HTTPs proxy: In this method, a product instance sends usage information over the internet through a proxy server - either a Call Home Transport Gateway or an off-the-shelf proxy (such as Apache) to CSSM.

Figure 4: Topology: Connected Directly to CSSM



Considerations or Recommendations:

Smart transport is the recommended transport method when directly connecting to CSSM. This recommendation applies to:

- New deployments.
- Earlier licensing models. Change configuration after migration to Smart Licensing Using Policy.
- Registered licenses that currently use the Call Home transport method. Change configuration after migration to Smart Licensing Using Policy.
- Evaluation or expired licenses in an earlier licensing model. Change configuration after migration to Smart Licensing Using Policy.

To change configuration after migration, see Workflow for Topology: Connected Directly to CSSM, on page 125 > Product Instance Configuration > Configure a connection method and transport type > Option 1.

Release-Wise Changes and Enhancements:

This section outlines important release-wise software changes and enhancements that affect this topology.

From Cisco IOS XE Cupertino 17.9.1:

- RUM report throttling

  The minimum reporting frequency for this topology, is throttled to one day. This means the product instance does not send more than one RUM report a day. This resolves the problem of too many RUM

reports being generated and sent for certain licenses. It also resolves the memory-related issues and system slow-down caused by an excessive generation of RUM reports.

You can override the throttling restriction by entering the **license smart sync** command in privileged EXEC mode.

RUM report throttling also applies to the Cisco IOS XE Amsterdam 17.3.6 and later releases of the 17.3.x train, and Cisco IOS XE Bengaluru 17.6.4 and later releases of the 17.6.x train. From 17.9.1, RUM report throttling is applicable to *all* subsequent releases.

Where to Go Next:

To implement this topology, see Workflow for Topology: Connected Directly to CSSM, on page 125.

Connected to CSSM Through a Controller

When you use a controller to manage a product instance, the controller connects to CSSM, and is the interface for all communication to and from CSSM. The supported controller for Cisco Catalyst Access, Core, and Aggregation Switches is Cisco DNA Center.

Overview

If a product instance is managed by Cisco DNA Center as the controller, the product instance records license usage and saves the same, but it is the Cisco DNA Center that initiates communication with the product instance to retrieve RUM reports, report to CSSM, and return the ACK for installation on the product instance.

All product instances that must be managed by Cisco DNA Center must be part of its inventory and must be assigned to a site. Cisco DNA Center uses the NETCONF protocol to provision configuration and retrieve the required information from the product instance - the product instance must therefore have NETCONF enabled, to facilitate this.

In order to meet reporting requirements, Cisco DNA Center retrieves the applicable policy from CSSM and provides the following reporting options:

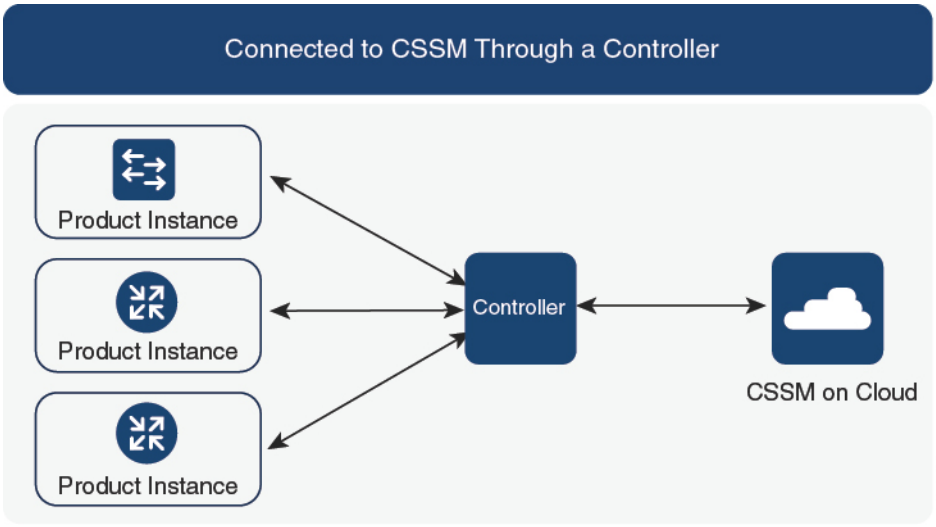
- Ad hoc reporting: You can trigger an ad hoc report when required.
- Scheduled reporting: Corresponds with the reporting frequency specified in the policy and is automatically handled by Cisco DNA Center.

Note Ad hoc reporting must be performed at least once before a product instance is eligible for scheduled reporting.

The first ad hoc report enables Cisco DNA Center to determine the Smart Account and Virtual Account to which subsequent RUM reports must be uploaded. You will receive notifications if ad hoc reporting for a product instance has not been performed even once.

A trust code is *not* required.

Figure 5: Topology: Connected to CSSM Through a Controller



Considerations or Recommendations:

This is the recommended topology if you are using Cisco DNA Center.

Note The HSECK9 key, which is an export-controlled license is supported on certain models of the Cisco Catalyst Access, Core, and Aggregation Switches (See Authorization Code, on page 101). If you are using a product instance where an HSECK9 key is supported, note that the Cisco DNA Center GUI does not provide an option to generate a SLAC.

Where to Go Next:

To implement this topology, see Workflow for Topology: Connected to CSSM Through a Controller, on page 127

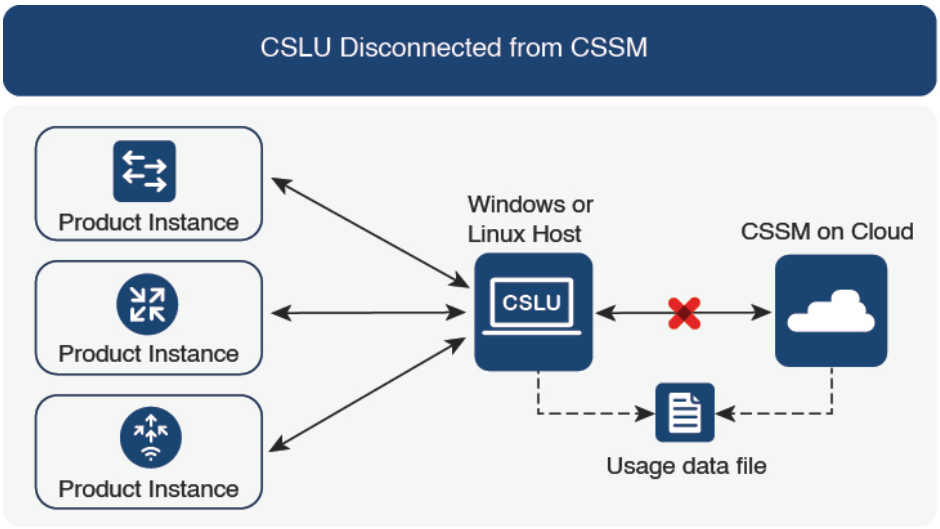
CSLU Disconnected from CSSM

Overview:

Here, a product instance communicates with CSLU, and you have the option of implementing product instance-initiated communication or CSLU-initiated communication (as in the *Connected to CSSM Through CSLU* topology). The other side of the communication, between CSLU and CSSM, is offline. CSLU provides you with the option of working in a mode that is disconnected from CSSM.

Communication between CSLU and CSSM is sent and received in the form of signed files that are saved offline and then uploaded to or downloaded from CSLU or CSSM, as the case may be.

Figure 6: Topology: CSLU Disconnected from CSSM



Considerations or Recommendations:

Choose the method of communication depending on your network's security policy.

Release-Wise Changes and Enhancements:

This section outlines important release-wise software changes and enhancements that affect this topology.

From Cisco IOS XE Cupertino 17.7.1:

- Trust code request and installation

  If a trust code is not available on the product instance, the product instance detects and automatically includes a request for one, as part of a RUM report that is sent to CSLU, which you upload to CSSM. The ACK that you download from CSSM includes the trust code. If there is an existing factory-installed trust code, it is automatically overwritten. A trust code obtained this way can be used for communication with CSSM.

  This is supported in a standalone, as well as a High Availability set-up. In a High Availability set-up, the active product instance requests the trust code for members or standbys where a trust code is not available.

  In this release, this enhancement applies only to the product instance-initiated mode.

From Cisco IOS XE Cupertino 17.9.1:

- Trust code request and installation

  From this release, trust code request and installation is supported in the CSLU-initiated mode as well.h

- RUM report throttling

  In the product instance-initiated mode, the minimum reporting frequency is throttled to one day. This means the product instance does not send more than one RUM report a day. This resolves the problem of too many RUM reports being generated and sent for certain licenses. It also resolves the memory-related issues and system slow-down caused by an excessive generation of RUM reports.

You can override the throttling restriction by entering the **license smart sync** command in privileged EXEC mode.

RUM report throttling also applies to the Cisco IOS XE Amsterdam 17.3.6 and later releases of the 17.3.x train, and Cisco IOS XE Bengaluru 17.6.4 and later releases of the 17.6.x train. From 17.9.1, RUM report throttling is applicable to *all* subsequent releases.

Where to Go Next:

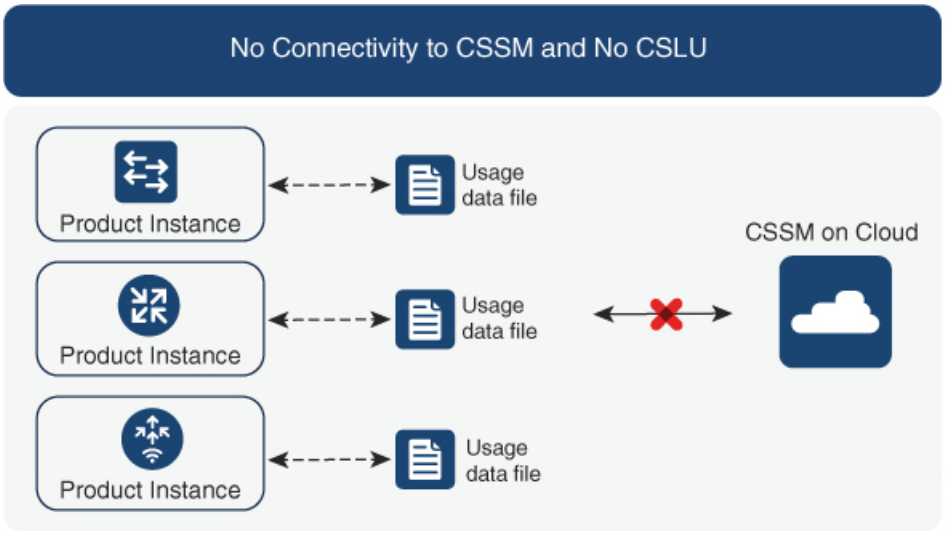
To implement this topology, see Workflow for Topology: CSLU Disconnected from CSSM, on page 128.

No Connectivity to CSSM and No CSLU

Overview:

Here you have a product instance and CSSM disconnected from each other, and without any other intermediary utilities or components. All communication is in the form of uploaded and downloaded files. These files can be RUM reports, requests for UDI-tied trust codes and SLAC request or return files.

Figure 7: Topology: No Connectivity to CSSM and No CSLU



Considerations or Recommendations:

This topology is suited to a high-security deployment where a product instance cannot communicate online, with anything outside its network.

Release-Wise Changes and Enhancements:

This section outlines important release-wise software changes and enhancements that affect this topology.

From Cisco IOS XE Cupertino 17.7.1:

- Trust code request and installation

If a trust code is not available on the product instance, the product instance automatically includes a trust code request in the RUM report that you save, to upload to CSSM. The ACK that you then download from CSSM includes the trust code.

If there is a factory-installed trust code, it is automatically overwritten when you install the ACK. A trust code obtained this way can be used for secure communication with CSSM.

This is supported in a standalone, as well as a High Availability set-up. In a High Availability set-up, the active product instance requests the trust code for all connected product instances where a trust code is not available.

- SLAC request and installation

  You can generate a SLAC request and save it in a file on the product instance. The saved file includes all the required details (UDI, license information etc). With this method you do not have to gather and enter the required details on the CSSM Web UI to generate a SLAC. You have to upload the SLAC request file to CSSM and download the file containing the SLAC code and install it on the product instance - as you would a RUM report and ACK.

  Similarly, when you return a SLAC you do not have to locate the product instance in the correct Virtual Account. Simply upload the SLAC return file, as you would a RUM report.

Where to Go Next:

To implement this topology, see Workflow for Topology: No Connectivity to CSSM and No CSLU, on page 131.

SSM On-Prem Deployment

Overview:

SSM On-Prem is designed to work as an extension of CSSM that is deployed on your premises.

Here, a product instance is connected to SSM On-Prem and SSM On-Prem becomes the single point of interface with CSSM. Each instance of SSM On-Prem must be made known to CSSM through a mandatory registration and synchronization of the local account in SSM On-Prem, with a Virtual Account in CSSM.

When you deploy SSM On-Prem to manage a product instance, the product instance can be configured to *push* the required information to SSM On-Prem. Alternatively, SSM On-Prem can be set-up to *pull* the required information from a product instance at a configurable frequency.

- Product instance-initiated communication (push): The product instance initiates communication with SSM On-Prem, by connecting to a REST endpoint in SSM On-Prem. Data that is sent includes RUM reports and requests for authorization codes, trust codes, and policies.

  Options for communication between the product instance and SSM On-Prem in this mode:

  - Use a CLI command to push information to SSM On-Prem as and when required.
  - Use a CLI command and configure a reporting interval, to automatically send RUM reports to SSM On-Prem at a scheduled frequency.

- SSM On-Prem-initiated communication (pull): To initiate the retrieval of information from a product instance, SSM On-Prem NETCONF, RESTCONF, and native REST API options, to connect to the product instance. Supported workflows include receiving RUM reports from the product instance and

sending the same to CSSM, authorization code installation, trust code installation, and application of policies.

Options for communication between the product instance and SSM On-Prem in this mode:

- Collect usage information from one or more product instances as and when required (on-demand).
- Collect usage information from one or more product instances at a scheduled frequency.

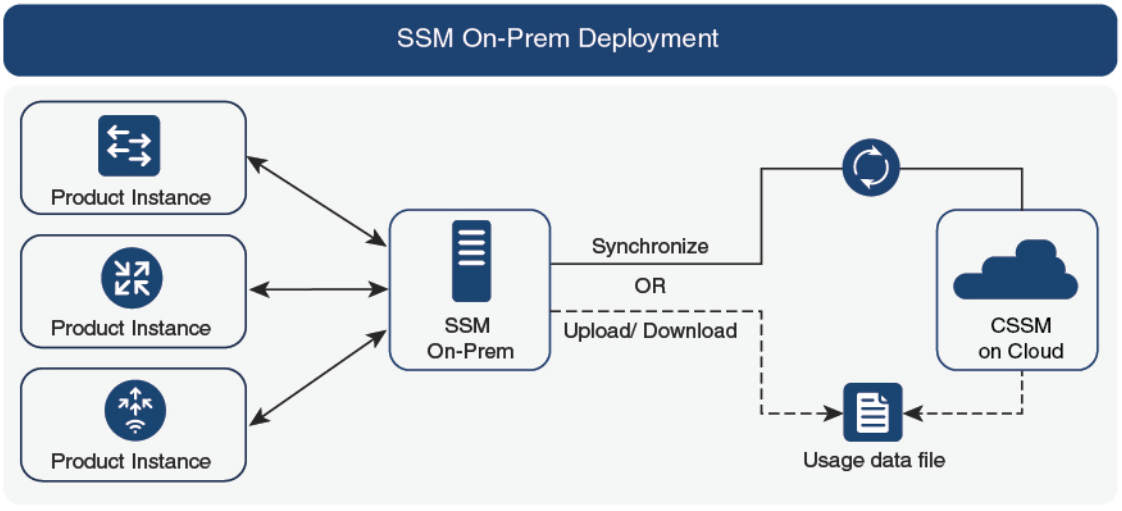
In SSM On-Prem, the reporting interval is set to the default policy on the product instance. You can change this, but only to report more frequently (a narrower interval), or you can install a custom policy if available.

After usage information is available in SSM On-Prem, you must synchronize the same with CSSM, to ensure that the product instance count, license count and license usage information is the same on both, CSSM and SSM On-Prem. Options for usage synchronization between SSM On-Prem and CSSM – for the push *and* pull mode:

- Perform ad-hoc synchronization with CSSM (Synchronize now with Cisco).
- Schedule synchronization with CSSM for specified times.
- Communicate with CSSM through signed files that are saved offline and then upload to or download from SSM On-Prem or CSSM, as the case may be.

Note This topology involves two different kinds of synchronization between SSM On-Prem and CSSM. The first is where the *local account* is synchronized with CSSM - this is for the SSM On-Prem instance to be known to CSSM and is performed by using the **Synchronization** widget in SSM On-Prem. The second is where *license usage* is synchronized with CSSM, either by being connected to CSSM or by downloading and uploading files. You must synchronize the local account before you can synchronize license usage.

Figure 8: Topology: SSM On-Prem Deployment



Considerations or Recommendations:

This topology is suited to the following situations:

- If you want to manage your product instances on your premises, as opposed communicating directly with CSSM for this purpose.
- If your company's policies prevent your product instances from reporting license usage directly to Cisco (CSSM).
- If your product instances are in an air-gapped network and cannot communicate online, with anything outside their network.

Apart from support for Smart Licensing Using Policy, some of the key benefits of SSM On-Prem *Version 8* include:

- Multi-tenancy: One tenant constitutes one Smart Account-Virtual Account pair. SSM On-Prem enables you to manage multiple pairs. Here you create local accounts that reside in SSM On-Prem. Multiple local accounts roll-up to a Smart Account-Virtual Account pair in CSSM. For more information, see the Cisco Smart Software Manager On-Prem User Guide > *About Accounts and Local Virtual Accounts*.

Note The relationship between CSSM and SSM On-Prem instances is still one-to-one.

- Scale: Supports up to a total of 300,000 product instances
- High-Availability: Enables you to run two SSM On-Prem servers in the form of an active-standby cluster. For more information, see the Cisco Smart Software On-Prem Installation Guide > Appendix 4. Managing a High Availability (HA) Cluster in Your System.

  High-Availability deployment is supported in the SSM On-Prem console and the required command details are available in the Cisco Smart Software On-Prem Console Guide.
- Options for online and offline connectivity to CSSM.

SSM On-Prem Limitations:

- Proxy support for communication with CSSM, for the purpose of *license usage* synchronization is available only from Version 8 202108 onwards. The use of a proxy for *local account* synchronization, which is performed by using the **Synchronization** widget, is available from the introductory SSM On-Prem release where Smart Licensing Using Policy is supported.
- SSM On-Prem-initiated communication is not supported on a product instance that is in a Network Address Translation (NAT) set-up. You must use product instance-initiated communication, and further, you must *enable* SSM On-Prem to support a product instance that is in a NAT setup. Details are provided in the workflow for this topology.

Release-Wise Changes and Enhancements:

This section outlines important release-wise software changes and enhancements that affect this topology.

From Cisco IOS XE Cupertino 17.9.1:

- RUM report throttling

In the product instance-initiated mode, the minimum reporting frequency is throttled to one day. This means the product instance does not send more than one RUM report a day. This resolves the problem of too many RUM reports being generated and sent for certain licenses. It also resolves the memory-related issues and system slow-down caused by an excessive generation of RUM reports.

You can override the throttling restriction by entering the **license smart sync** command in privileged EXEC mode.

RUM report throttling also applies to the Cisco IOS XE Amsterdam 17.3.6 and later releases of the 17.3.x train, and Cisco IOS XE Bengaluru 17.6.4 and later releases of the 17.6.x train. From 17.9.1, RUM report throttling is applicable to *all* subsequent releases.

Where to Go Next:

To implement this topology, see Workflow for Topology: SSM On-Prem Deployment, on page 132.

If you are migrating from an existing version of SSM On-Prem, the sequence in which you perform the various upgrade-related activities is crucial. See Migrating to a Version of SSM On-Prem That Supports Smart Licensing Using Policy, on page 160.

Interactions with Other Features

High Availability

This section explains considerations that apply to a High Availability configuration, when running a software version that supports Smart Licensing Using Policy. The following High Availability setups are within the scope of this document:

A device stack with an active, a standby and one or more members.

A dual-supervisor setup, where two supervisor modules are installed in a chassis, one being the active and the other, the standby.

A dual-chassis setup[7] (could be fixed or modular), with the active in one chassis and a standby in the other chassis.

A dual-chassis *and* dual-supervisor setup[8], on a modular chassis. Two chassis are involved here as well. An active supervisor module is in one chasses and a standby supervisor module in a second chassis. The "dual-supervisor" aspect refers to an additional in-chassis standby supervisor in just one of the chassis, which is the minimum requirement, or an in-chassis standby supervisor in each chassis.

Authorization Code Requirements in a High Availability Setup

The number of SLACs required in a High Availability setup, corresponds with the number of UDIs. Tabled below are the stacking and High Availability setups that are supported when using an export-controlled license (HSECK9 key), and the SLAC requirements in each setup.

Note Each HSECK9 key requires a SLAC. Therefore, the number SLACs will always correspond with the number of HSECK9 keys.

[7] The Cisco StackWise Virtual feature, which is available on certain Cisco Catalyst Access, Core, and Aggregation Switches, is an example of such a setup.
[8] The Quad-Supervisor with Route Processor Redundancy, which is available on certain Cisco Catalyst Access, Core, and Aggregation Switches, is an example of such a setup.

Product Instance Supporting HSECK9 Key	Supported High Availability Setup When Using HSECK9 Key	SLAC Requirements in the Setup
Cisco Catalyst 9300X Series Switches	A device stack with an active, a standby and one or more members.	The SLAC requirement corresponds with the number of UDIs on which you want to configure the cryptographic feature. Each such UDI in the stack requires one SLAC. At a minimum, only the active requires a SLAC. But for uninterrupted use of the cryptographic feature in the event of a switchover, we recommend that you install SLAC on the standby also.
Cisco Catalyst 9500X Series Switches.	None.	Not applicable. High Availability is not supported on Cisco Catalyst 9500X Series Switches.
C9600-LC-40YL4CD line card with supervisor module C9600X-SUP-2	A dual-supervisor setup, where two supervisor modules are installed in a chassis, one being the active and the other, the standby. No other High Availability setup is supported when using an HSECK9 key.	The SLAC requirement corresponds with the number of UDIs. The UDI is tied to the chassis and not the individual supervisor modules. (The UDIs of the active and standby supervisor modules are the same). One SLAC is required for each chassis UDI, regardless of the number of supervisors installed.
Cisco Catalyst 9400 Series Supervisor 2 and 2XL Modules (C9400X-SUP-2 and C9400X-SUP-2XL)	• A dual-supervisor setup, where two supervisor modules are installed in a chassis, one being the active and the other, the standby. • A Cisco StackWise Virtual setup, which involves two chassis. One supervisor module is installed in each chassis, one being the active and the other, the standby.	The SLAC requirement corresponds with the number of UDIs. The UDI is tied to the chassis and not the individual supervisor modules. • In a dual-supervisor setup, one SLAC is required for each chassis UDI, regardless of the number of supervisors installed. • In a Cisco StackWise Virtual setup, at a minimum, you must obtain a SLAC for the chassis with the active supervisor module. But for uninterrupted use of the cryptographic feature in the event of a switchover, we recommend that you obtain an SLAC for the chassis with the standby supervisor module also.

Trust Code Requirements in a High Availability setup

The number of trust codes required depends on the number of UDIs. The active product instance can submit requests for all devices in the High Availability setup and install all the trust codes that are returned in an ACK.

Policy Requirements in a High Availability setup

There are no policy requirements that apply exclusively to a High Availability setup. As in the case of a standalone product instance, only one policy exists in a High Availability setup as well, and this is on the active. The policy on the active applies to the standby or members in the setup.

Product Instance *Functions* in a High Availability setup

This section explains general product instance functions in a High Availability setup, as well as what the product instance does when a new standby or member is added to an existing High Available setup.

For authorization and trust codes: The active product instance can request (if required) and install authorization codes and trust codes for standbys and members.

For policies: The active product instance synchronizes with the standby.

For reporting: Only the active product instance reports usage. The active reports usage information for all devices (standbys or members – as applicable) in the High Availability setup.

In addition to scheduled reporting, the following events trigger reporting:

- The addition or removal of a standby. The RUM report includes information about the standby that was added or removed.
- The addition or removal of a member, including stack merge and stack split events. The RUM report includes information about member that was added or removed.
- A switchover.
- A reload.

When one of the above events occur, the "Next report push" date of the **show license status** privileged EXEC command is updated. But it is the implemented topology and associated reporting method that determine if the report is sent by the product instance or not. For example, if you have implemented a topology where the product instance is disconnected (Transport Type is Off), then the product instance does not send RUM reports even if the "Next report push" date is updated.

For a new member or standby addition:

- A product instance that is connected to CSLU, does not take any further action.
- A product instance that is directly connected to CSSM, performs trust synchronization. Trust synchronization involves the following:

  Installation of trust code on the standby or member if not installed already.

  If a trust code is already installed, the trust synchronization process ensures that the new standby or member is in the same Smart Account and Virtual Account as the active. If it is not, the new standby or member is *moved* to the same Smart Account and Virtual Account as the active.

  Installation of an authorization code, policy, and purchase information, if applicable

  Sending of a RUM report with current usage information.

Upgrades

This section explains the following aspects:

- Migrating from earlier licensing models to Smart Licensing Using Policy.

Earlier licensing models include Smart Licensing, Specific License Reservation (SLR), Right-to-Use Licensing (RTU), and evaluation or expired licenses from earlier licensing models. The Migrating to Smart Licensing Using Policy, on page 137 section provides details and examples for migration scenarios.

Device-led conversion is not supported for migration to Smart Licensing Using Policy.

- Upgrading in the Smart Licensing Using Policy environment - where the software version you are upgrading from and the software version you are upgrading to, both support Smart Licensing Using Policy.

Refer to the corresponding sections:

Identifying the Current Licensing Model Before Upgrade

Before you upgrade to Smart Licensing Using Policy, if you want to know the current licensing model that is effective on the product instance, enter the **show license all** command in privileged EXEC mode. This command displays information about the current licensing model for all except the RTU licensing model. The **show license right-to-use** privileged EXEC command displays license information only if the licensing model is RTU.

How Upgrade Affects Enforcement Types for Existing Licenses

When you upgrade to a software version which supports Smart Licensing Using Policy, the way existing licenses are handled, depends primarily on the license enforcement type.

- An unenforced license that was being used before upgrade, continues to be available after the upgrade. This includes all licenses from all earlier licensing models.
  - Smart Licensing.
  - Specific License Reservation (SLR), which has an accompanying authorization code. The authorization code continues to be valid after upgrade to Smart Licensing Using Policy and authorizes existing license consumption.
  - Right-to-Use (RTU) Licensing.
  - Evaluation or expired licenses from any of the above mentioned licensing models.
- An enforced or export-controlled license that was being used before upgrade, continues to be available after upgrade if the required authorization exists.

  An export-controlled license is supported on certain models and only starting from Cisco IOS XE Bengaluru 17.6.2. No export-controlled or enforced licenses were available on any of the Cisco Catalyst Access, Core, and Aggregation Switches prior to this.

How Upgrade Affects Reporting for Existing Licenses

Existing License	Reporting Requirements After Migration to Smart Licensing Using Policy
Right-to-Use (RTU)	Depends on the license being used. After migration and deployment of a supported topology, in output of the **show license usage** command, refer to the `Next ACK deadline` field to know if and when reporting is required.

Existing License	Reporting Requirements After Migration to Smart Licensing Using Policy
Specific License Reservation (SLR)	Required only if there is a change in license consumption. An existing SLR authorization code authorizes existing license consumption after upgrade to Smart Licensing Using Policy.
Smart Licensing (Registered and Authorized licenses): Reporting for these licenses is based on the reporting requirements in the policy.	Depends on the policy.
Evaluation or expired licenses	Based on the reporting requirements of the Cisco default policy.

How Upgrade Affects Transport Type for Existing Licenses

The transport type, if configured in your existing set-up, is retained after upgrade to Smart Licensing Using Policy.

When compared to the earlier version of Smart Licensing, additional transport types are available with Smart Licensing Using Policy. There is also a change in the default transport mode. The following table clarifies how this may affect upgrades:

Transport type Before Upgrade	License or License State Before Upgrade	Transport Type After Upgrade
Default (callhome)	evaluation	cslu (default in Smart Licensing Using Policy)
	SLR	off
	registered	callhome
smart	evaluation	off
	SLR	off
	registered	smart
Not applicable For example, if the existing licensing model is RTU.	Not applicable For example, if the existing licensing model is RTU.	cslu

How Upgrade Affects the Token Registration Process

In the earlier version of Smart Licensing, a token was used to register and connect to CSSM. ID token registration is not required in Smart Licensing Using Policy. The token generation feature is still available in CSSM, and is used to *establish trust* when a product instance is directly connected to CSSM. See Connected Directly to CSSM.

Upgrades Within the Smart Licensing Using Policy Environment

This section covers any release-specific considerations or actions that apply when you upgrade the product instance from one release where Smart Licensing Using Policy is supported to another release where Smart Licensing Using Policy is supported.

Starting with Cisco IOS XE Cupertino 17.7.1, RUM reports are stored in a format that reduces processing time. In order to ensure that there are no usage reporting inconsistencies resulting from the differences in the old and new formats, we recommend completing one round of usage reporting as a standard practice when upgrading from an earlier release that supports Smart Licensing Using Policy, to Cisco IOS XE Cupertino 17.7.1 or a later release.

Downgrades

This section provides information about downgrades to an earlier licensing model, for new deployments and existing deployments. It also covers information relevant to downgrades within the Smart Licensing Using Policy environment.

New Deployment Downgrade

This section applies if you had a newly purchased product instance with a software version where Smart Licensing Using Policy was already enabled by default and you want to downgrade to a software version where Smart Licensing Using Policy is not supported.

The outcome of the downgrade depends on whether a Trust Code was installed while you were still operating in the Smart Licensing Using Policy environment, and further action may be required depending on the release you downgrade to.

If the topology you implemented while in the Smart Licensing Using Policy environment was "Connected Directly to CSSM", then a trust code installation can be expected or assumed, because it is required as part of topology implementation. For any of the other topologies, trust establishment is not mandatory. Downgrading product instances with one of these other topologies will therefore mean that you have to restore licenses to a registered and authorized state by following the procedures that are applicable in the Smart Licensing environment. See Outcome and Action for New Deployment Downgrade to Smart Licensing below.

Table 9: Outcome and Action for New Deployment Downgrade to Smart Licensing

In the Smart Licensing Using Policy Environment	Downgrade to..	Outcome and Further Action
Standalone product instance, connected directly to CSSM, and trust established.	Cisco IOS XE Amsterdam 17.3.1 OR Cisco IOS XE Gibraltar 16.12.4 and later releases in Cisco IOS XE Gibraltar 16.12.x OR Cisco IOS XE Fuji 16.9.6 and later releases in Cisco IOS XE Fuji 16.9.x	No further action is required. The product instance attempts to renew trust with CSSM after downgrade. After a successful renewal, licenses are in a registered state and the earlier version of Smart Licensing is effective on the product instance.
	Any other release (other than the ones mentioned in the row above) that supports Smart Licensing	Action is required: You must reregister the product instance. Generate an ID token in the CSSM Web UI and on the product instance, configure the **license smart register idtoken** *idtoken* command in global configuration mode.
High Availability set-up, connected directly to CSSM, and trust established.	Any release that supports Smart Licensing	Action is required: You must reregister the product instance. Generate an ID token in the CSSM Web UI and on the product instance, configure the **license smart register idtoken** *idtoken* **all** command in global configuration mode.
Any other topology. (Connected to CSSM Through CSLU, CSLU Disconnected from CSSM, No Connectivity to CSSM and No CSLU)	Any release that supports Smart Licensing	Action is required. Restore licenses to a registered and authorized state by following the procedures that are applicable in the Smart Licensing environment.

Upgrading to Smart Licensing Using Policy and Then Downgrading

Downgrades Within the Smart Licensing Using Policy Environment

This section covers any release-specific considerations or actions that apply when you downgrade the product instance from one release where Smart Licensing Using Policy is supported to another release where Smart Licensing Using Policy is supported.

Starting with Cisco IOS XE Cupertino 17.7.1, RUM reports are stored in a format that reduces processing time. In order to ensure that there are no usage reporting inconsistencies resulting from the differences in the

old and new formats, we recommend completing one round of usage reporting as a standard practice when downgrading from Cisco IOS XE Cupertino 17.7.1 or a later release to an earlier release supporting Smart Licensing Using Policy.

How to Configure Smart Licensing Using Policy: Workflows by Topology

This section provides the simplest and fastest way to implement a topology.

Note These workflows are meant for new deployments only. If you are migrating from an existing licensing model, see Migrating to Smart Licensing Using Policy, on page 137.

Workflow for Topology: Connected to CSSM Through CSLU

Depending on whether you want to implement a product instance-initiated or CSLU-initiated method of communication, complete the corresponding sequence of tasks:

- Tasks for Product Instance-Initiated Communication
- Tasks for CSLU-Initiated Communication

Tasks for Product Instance-Initiated Communication

CSLU Installation → CSLU Preference Settings → Product Instance Configuration

1. ***CSLU Installation***

   Where task is performed: A Windows host (laptop, destop, or a Virtual Machine (VM)

   Download the file from Smart Software Manager > **Smart Licensing Utility**.

   Refer to the Cisco Smart License Utility Quick Start Setup Guide for help with installation and set-up.

2. ***CSLU Preference Settings***

   Where tasks are performed: CSLU

   a. Logging into Cisco (CSLU Interface), on page 162

   b. Configuring a Smart Account and a Virtual Account (CSLU Interface), on page 162

   c. Adding a Product-Initiated Product Instance in CSLU (CSLU Interface), on page 162

3. ***Product Instance Configuration***

   Where tasks are performed: Product Instance

   a. Ensuring Network Reachability for Product Instance-Initiated Communication, on page 163

   b. Ensure that transport type is set to **cslu**.

CSLU is the default transport type. If you have configured a different option, enter the **license smart transport cslu** command in global configuration mode. Save any changes to the configuration file.

```
Device(config)# license smart transport cslu
Device(config)# exit
Device# copy running-config startup-config
```

c. Specify how you want CSLU to be discovered (*choose one*):

- Option 1:

  No action required. Name server configured for Zero-touch DNS discovery of `cslu-local`

  Here, if you have configured DNS (The name server IP address is configured on the product instance), and the DNS server has an entry where hostname `cslu-local` is mapped to the CSLU IP address, then no further action is required. The product instance automatically discovers hostname `cslu-local`.

- Option 2:

  No action required. Name server and domain configured for Zero-touch DNS discovery of `cslu-local.<domain>`

  Here if you have configured DNS, (The name server IP address and domain is configured on the product instance), and the DNS server has an entry where `cslu-local.<domain>` is mapped to the CSLU IP address, then no further action is required. The product instance automatically discovers hostname cslu-local.

- Option 3:

  Configure a specific URL for CSLU.

  Enter the **license smart url cslu** *http://<cslu_ip_or_host>:8182/cslu/v1/pi* command in global configuration mode. For `<cslu_ip_or_host>`, enter the hostname or the IP address of the windows host where you have installed CSLU. 8182 is the port number and it is the only port number that CSLU uses.

  ```
  Device(config)# license smart url cslu http://192.168.0.1:8182/cslu/v1/pi
  Device(config)# exit
  Device# copy running-config startup-config
  ```

Result:

Since the product instance initiates communication, it automatically sends out the first RUM report at the scheduled time, as per the policy. Along with this first report, if applicable and if required, it sends a trust code request. To know when the product instance will be sending this information, enter the **show license all** command in privileged EXEC mode and in the output, check the date for field `Next report push:`. CSLU forwards the information to CSSM and the returning ACK from CSSM, to the product instance.

The following applies only to Cisco IOS XE Amsterdam 17.3.6 and later releases of the 17.3.x train: In the product instance-initiated mode, the product instance does not send more than one RUM report a day. You can override this for an on-demand synchronization between the product instance and CSLU, by entering the **license smart sync** command in privileged EXEC mode.

In case of a change in license usage, see Configuring a Base or Add-On License , on page 215 to know how it affects reporting.

Tasks for CSLU-Initiated Communication

CSLU Installation → CSLU Preference Settings→ Product Instance Configuration

1. ***CSLU Installation***

   Where task is performed: A Windows host (laptop, destop, or a Virtual Machine (VM)

   Download the file from Smart Software Manager > **Smart Licensing Utility**.

   Refer to the Cisco Smart License Utility Quick Start Setup Guide for help with installation and set-up.

2. ***CSLU Preference Settings***

   Where tasks is performed: CSLU

   a. Logging into Cisco (CSLU Interface), on page 162

   b. Configuring a Smart Account and a Virtual Account (CSLU Interface), on page 162

   c. Adding a CSLU-Initiated Product Instance in CSLU (CSLU Interface), on page 165

   d. Collecting Usage Reports: CSLU Initiated (CSLU Interface), on page 165

3. ***Product Instance Configuration***

   Where tasks is performed: Product Instance

   Ensuring Network Reachability for CSLU-Initiated Communication, on page 167

Result:

You can now collect and send a RUM report to CSSM, in CSLU, by navigating to the **Actions for Selected…** menu in CSLU, and selecting **Collect Usage**. The RUM report is sent to CSSM. Along with this first report, if applicable and if required, CSLU sends a trust code request to CSSM. It gets the ACK from CSSM and sends this back to the product instance for installation.

In case of a change in license usage, see Configuring a Base or Add-On License , on page 215 to know how it affects reporting.

Workflow for Topology: Connected Directly to CSSM

Smart Account Set-Up → Product Instance Configuration → Trust Establishment with CSSM→ Authorization Code Installation (Only if Applicable)

1. ***Smart Account Set-Up***

   Where task is performed: CSSM Web UI, https://software.cisco.com/.

   Ensure that you have a user role with proper access rights to a Smart Account and the required Virtual Accounts.

2. ***Product Instance Configuration***

   Where tasks are performed: Product Instance

   a. Set-Up product instance connection to CSSM: Setting Up a Connection to CSSM , on page 172.

   b. Configure a connection method and transport type (choose one)

      • Option 1:

Smart transport: Set transport type to **smart** and configure the corresponding URL.

If the transport mode is set to **license smart transport smart**, and you configure **license smart url default**, the Smart URL (https://smartreceiver.cisco.com/licservice/license) is automatically configured. Save any changes to the configuration file.

```
Device(config)# license smart transport smart
Device(config)# license smart url default
Device(config)# exit
Device# copy running-config startup-config
```

- Option 2:

  Configure Smart transport through an HTTPs proxy. See Configuring Smart Transport Through an HTTPs Proxy, on page 174

- Option 3:

  Configure Call Home service for direct cloud access. See Configuring the Call Home Service for Direct Cloud Access, on page 175.

- Option 4:

  Configure Call Home service for direct cloud access through an HTTPs proxy. See Configuring the Call Home Service for Direct Cloud Access through an HTTPs Proxy Server, on page 178.

3. ***Trust Establishment with CSSM***

   Where task is performed: CSSM Web UI and then the product instance

   a. Generate one token for each *Virtual Account* you have. You can use same token for all the product instances that are part of one Virtual Account: Generating a New Token for a Trust Code from CSSM, on page 207.

   b. Having downloaded the token, you can now install the trust code on the product instance: Establishing Trust with an ID Token., on page 208.

4. ***Authorization Code Installation (Only if Applicable)***

   Where tasks are performed: Product Instance

   An export-controlled license is supported only on certain models of the Cisco Catalyst Access, Core, and Aggregation Switches (See Authorization Code, on page 101). If you want to use an export-controlled license, complete the following task on supported platforms: Manually Requesting and Auto-Installing a SLAC , on page 191.

Result:

After establishing trust, CSSM returns a policy. The policy is automatically installed on all product instances of that Virtual Account. The policy specifies if and how often the product instance reports usage.

The following applies only to Cisco IOS XE Amsterdam 17.3.6 and later releases of the 17.3.x train and Cisco IOS XE Bengaluru 17.6.4 and later releases of the 17.6.x train: the product instance does not send more than one RUM report a day. You can override this for an on-demand synchronization between the product instance and CSLU, by entering the **license smart sync** command in privileged EXEC mode.

If you want to change your reporting interval to report more frequently: on the product instance, configure the **license smart usage interval** command in global configuration mode. For syntax details see the *license smart (privileged EXEC)* command in the Command Reference for the corresponding release.

If you want to change the boot level license, see Configuring a Base or Add-On License , on page 215.

If you want to return an authorization code, see Returning an Authorization Code, on page 201.

Workflow for Topology: Connected to CSSM Through a Controller

To deploy Cisco DNA Center as the controller, complete the following workflow:

Product Instance Configuration → Cisco DNA Center Configuration

1. ***Product Instance Configuration***

   Where task is performed: Product Instance

   Enable NETCONF. Cisco DNA Center uses the NETCONF protocol to provision configuration and retrieve the required information from the product instance - the product instance must therefore have NETCONF enabled, to facilitate this.

   For more information, see the Programmability Configuration Guide, Cisco IOS XE Amsterdam 17.3.x. In the guide, go to *Model-Driven Programmability* > *NETCONF Protocol*.

2. ***Cisco DNA Center Configuration***

   Where tasks is performed: Cisco DNA Center GUI

   An outline of the tasks you must complete and the accompanying documentation reference is provided below. The document provides detailed steps you have to complete in the Cisco DNA Center GUI:

   a. Set-up the Smart Account and Virtual Account.

   Enter the same log in credentials that you use to log in to the CSSM Web UI. This enables Cisco DNA Center to establish a connection with CSSM.

   See the *Cisco DNA Center Administrator Guide* of the required release (Release 2.2.2 onwards) > *Manage Licenses* > *Set Up License Manager*.

   b. Add the required product instances to Cisco DNA Center inventory and assign them to a site.

   This enables Cisco DNA Center to push any necessary configuration, including the required certificates, for Smart Licensing Using Policy to work as expected.

   See the *Cisco DNA Center User Guide* of the required release (Release 2.2.2 onwards) > *Display Your Network Topology* > *Assign Devices to a Site*.

Result:

After you implement the topology, *you* must trigger the very first ad hoc report in Cisco DNA Center, to establish a mapping between the Smart Account and Virtual Account, and product instance. See the *Cisco DNA Center Administrator Guide* of the required release (Release 2.2.2 onwards) > *Manage Licenses* > *Upload Resource Utilization Details to CSSM*. Once this is done, Cisco DNA Center handles subsequent reporting based on the reporting policy.

If multiple policies are available, Cisco DNA Center maintains the narrowest reporting interval. You can change this, but only to report more frequently (a narrower interval). See the *Cisco DNA Center Administrator Guide* of the required release (Release 2.2.2 onwards) > *Manage Licenses* > *Modify License Policy*.

If you want to change the license level after this, see the *Cisco DNA Center Administrator Guide* of the required release (Release 2.2.2 onwards) > *Manage Licenses* > *Change License Level*.

Workflow for Topology: CSLU Disconnected from CSSM

Depending on whether you want to implement a product instance-initiated or CSLU-initiated method of communication. Complete the corresponding table of tasks below.

- Tasks for Product Instance-Initiated Communication
- Tasks for CSLU-Initiated Communication

Tasks for Product Instance-Initiated Communication

CSLU Installation → CSLU Preference Settings → Product Instance Configuration→ Authorization Code Installation (Only if Applicable)→ Usage Synchronization

1. ***CSLU Installation***

   Where task is performed: A Windows host (laptop, destop, or a Virtual Machine (VM)

   Download the file from Smart Software Manager > **Smart Licensing Utility**.

   Refer to the Cisco Smart License Utility Quick Start Setup Guide for help with installation and set-up.

2. ***CSLU Preference Settings***

   Where tasks are performed: CSLU interface

   a. In the CSLU Preferences tab, click the **Cisco Connectivity** toggle switch to **off**. The field switches to "Cisco Is Not Available".

   b. Configuring a Smart Account and a Virtual Account (CSLU Interface), on page 162

   c. Adding a Product-Initiated Product Instance in CSLU (CSLU Interface), on page 162

3. ***Product Instance Configuration***

   Where tasks are performed: Product Instance

   a. Ensuring Network Reachability for Product Instance-Initiated Communication, on page 163

   b. Ensure that transport type is set to **cslu**.

      CSLU is the default transport type. If you have configured a different option, enter the **license smart transport cslu** command in global configuration mode. Save any changes to the configuration file.

      ```
      Device(config)# license smart transport cslu
      Device(config)# exit
      Device# copy running-config startup-config
      ```

   c. Specify how you want CSLU to be discovered (*choose one*)

      - Option 1:

        No action required. Name server configured for Zero-touch DNS discovery of `cslu-local`

        Here, if you have configured DNS (The name server IP address is configured on the product instance), and the DNS server has an entry where hostname `cslu-local` is mapped to the CSLU IP address, then no further action is required. The product instance automatically discovers hostname `cslu-local`.

      - Option 2:

No action required. Name server and domain configured for Zero-touch DNS discovery of `cslu-local.<domain>`

Here if you have configured DNS, (The name server IP address and domain is configured on the product instance), and the DNS server has an entry where `cslu-local.<domain>` is mapped to the CSLU IP address, then no further action is required. The product instance automatically discovers hostname cslu-local.

- Option 3:

  Configure a specific URL for CSLU.

  Enter the **license smart url cslu** *http://<cslu_ip_or_host>:8182/cslu/v1/pi* command in global configuration mode. For `<cslu_ip_or_host>`, enter the hostname or the IP address of the windows host where you have installed CSLU. 8182 is the port number and it is the only port number that CSLU uses.

```
Device(config)# license smart url cslu http://192.168.0.1:8182/cslu/v1/pi
Device(config)# exit
Device# copy running-config startup-config
```

4. ***Authorization Code Installation (Only if Applicable)***

   Where tasks are performed: Product Instance and CSSM Web UI

   An export-controlled license is supported only on certain models of the Cisco Catalyst Access, Core, and Aggregation Switches (See Authorization Code, on page 101). If you want to use an export-controlled license, complete the following tasks on supported platforms:

   a. Manually Requesting and Auto-Installing a SLAC , on page 191

   b. Requesting SLAC for One or More Product Instance (CSLU Interface), on page 171

   c. Generating and Downloading SLAC from CSSM to a File, on page 199

   d. Import from CSSM (CSLU Interface), on page 166

5. ***Usage Synchronization***

   Where tasks are performed: CSLU and CSSM

   Since the product instance initiates communication, it automatically sends out the first RUM report at the scheduled time, as per the policy. You can also enter the **license smart sync** privileged EXEC command to trigger this. Along with this first report, if applicable, it sends a request for a UDI-tied trust code. Since CSLU is disconnected from CSSM, perform the following tasks to send the RUM Reports to CSSM.

   a. Export to CSSM (CSLU Interface), on page 166

   b. Uploading Data or Requests to CSSM and Downloading a File, on page 210

   c. Import from CSSM (CSLU Interface), on page 166

Result:

The ACK you have imported from CSSM contains the trust code if this was requested. The ACK is applied to the product instance the next time the product instance contacts CSLU.

The following applies only to Cisco IOS XE Amsterdam 17.3.6 and later releases of the 17.3.x train and Cisco IOS XE Bengaluru 17.6.4 and later releases of the 17.6.x train: in the product instance-initiated mode, the

product instance does not send more than one RUM report a day. You can override this for an on-demand synchronization between the product instance and CSLU, by entering the **license smart sync** command in privileged EXEC mode.

To know when the product instance will be sending the next RUM report, enter the **show license all** command in privileged EXEC mode and in the output, check the date for the `Next report push` field.

If you want to change the boot level license, see Configuring a Base or Add-On License , on page 215.

If you want to return an authorization code, see Returning an Authorization Code, on page 201.

Tasks for CSLU-Initiated Communication

CSLU Installation → CSLU Preference Settings → Product Instance Configuration → Authorization Code Installation (Only if Applicable) → Usage Synchronization

1. ***CSLU Installation***

   Where task is performed: A Windows host (laptop, destop, or a Virtual Machine (VM)

   Download the file from Smart Software Manager > **Smart Licensing Utility**.

   Refer to the Cisco Smart License Utility Quick Start Setup Guide for help with installation and set-up.

2. ***CSLU Preference Settings***

   Where tasks is performed: CSLU

   a. In the CSLU Preferences tab, click the **Cisco Connectivity** toggle switch to **off**. The field switches to “Cisco Is Not Available”.

   b. Configuring a Smart Account and a Virtual Account (CSLU Interface), on page 162

   c. Adding a CSLU-Initiated Product Instance in CSLU (CSLU Interface), on page 165

   d. Collecting Usage Reports: CSLU Initiated (CSLU Interface), on page 165

3. ***Product Instance Configuration***

   Where task is performed: Product Instance

   Ensuring Network Reachability for CSLU-Initiated Communication, on page 167

4. ***Authorization Code Installation (Only if Applicable)***

   Where tasks are performed: Product Instance

   An export-controlled license is supported only on certain models of the Cisco Catalyst Access, Core, and Aggregation Switches (See Authorization Code, on page 101). If you want to use an export-controlled license, complete the following tasks on supported platforms:

   a. Manually Requesting and Auto-Installing a SLAC , on page 191

   b. Requesting SLAC for One or More Product Instance (CSLU Interface), on page 171

   c. Generating and Downloading SLAC from CSSM to a File, on page 199

   d. Import from CSSM (CSLU Interface), on page 166

5. ***Usage Synchronization***

Where tasks are performed: CSLU and CSSM

Collect usage data from the product instance. Since CSLU is disconnected from CSSM, you then save usage data which CSLU has collected from the product instance to a file. Then, from a workstation that is connected to Cisco, upload it to CSSM. After this, download the ACK from CSSM. In the workstation where CSLU is installed and connected to the product instance, upload the file to CSLU.

a. Export to CSSM (CSLU Interface), on page 166

b. Uploading Data or Requests to CSSM and Downloading a File, on page 210

c. Import from CSSM (CSLU Interface), on page 166

Result:

The uploaded ACK is applied to the product instance the next time CSLU runs an update.

If you want to change the boot level license, see Configuring a Base or Add-On License , on page 215.

If you want to return an authorization code, see Returning an Authorization Code, on page 201.

Workflow for Topology: No Connectivity to CSSM and No CSLU

Since you do not have to configure connectivity to any other component, the list of tasks required to set-up the topology is a small one. See, the ***Results*** section at the end of the workflow to know how you can complete requisite usage reporting after you have implemented this topology.

Product Instance Configuration

Where task is performed: Product Instance

Set transport type to **off**.

Enter the **license smart transport off** command in global configuration mode. Save any changes to the configuration file.

```
Device(config)# license smart transport off
Device(config)# exit
Device# copy running-config startup-config
```

Result:

All communication to and from the product instance is disabled. To report license usage you must save RUM reports to a file (on your product instance) and upload it to CSSM (from a workstation that has connectivity to the internet, and Cisco):

1. Generate and save RUM reports

   Enter the **license smart save usage** command in privileged EXEC mode. In the example below, all RUM reports are saved to the flash memory of the product instance, in file `all_rum.txt`. In the example, the file is first saved to bootflash and then copied to a TFTP location:

   ```
   Device# license smart save usage all file bootflash:all_rum.txt
   Device# copy bootflash:all_rum.txt tftp://10.8.0.6/user01
   ```

2. Upload usage data to CSSM: Uploading Data or Requests to CSSM and Downloading a File, on page 210

3. Install the ACK on the product instance: Installing a File on the Product Instance, on page 211

In case of a change in license usage, see Configuring a Base or Add-On License , on page 215 to know how it affects reporting.

Workflow for Topology: SSM On-Prem Deployment

Depending on whether you want to implement a product instance-initiated method of communicatio (push) or SSM On-Prem-initiated method of communication (pull), complete the corresponding sequence of tasks:

Tasks for Product Instance-Initiated Communication

SSM On-Prem Installation → Addition and Validation of Product Instances (Only if Applicable) → Product Instance Configuration→ Initial Usage Synchronization

1. ***SSM On-Prem Installation***

   Where task is performed: A physical server such as a Cisco UCS C220 M3 Rack Server, or a hardware-based server that meets the necessary requirements.

   Download the file from Smart Software Manager > **Smart Software Manager On-Prem**.

   Refer to the *Cisco Smart Software On-Prem Installation Guide* and the *Cisco Smart Software On-Prem User Guide* for help with installation.

   Installation is complete when you have deployed SSM On-Prem, configured a common name on SSM On-Prem (**Security Widget** > **Certificates**), synchronized the NTP server (**Settings** widget > **Time Settings**), and created, registered, and synchronized (**Synchronization** widget) the SSM On-Prem local account with your Smart Account and Virtual Account in CSSM.

Note Licensing functions in the **On-Prem Licensing Workspace** are greyed-out until you complete the creation, registration, and synchronization of the local account with your Smart Account in CSSM. The *local account* synchronization with CSSM is for the SSM On-Prem instance to be known to CSSM, and is different from usage synchronization which is performed in **4. Initial Usage Synchronization** below.

2. ***Addition and Validation of Product Instances***

   Where tasks are performed: SSM On-Prem UI

   This step ensures that the product instances are validated and mapped to the applicable Smart Account and Virtual account in CSSM. This step is required only in the following cases:

   - If you want your product instances to be added and validated in SSM On-Prem before they are reported in CSSM (for added security).
   - If you want to use a license that requires authorization before use (enforcement type: enforced or export-controlled). Such a product instance must be added to SSM On-Prem before you can request the necessary SLAC in Step 3 d below.
   - If you have created local virtual accounts (in addition to the default local virtual account) in SSM On-Prem. In this case you must provide SSM On-Prem with the Smart Account and Virtual Account information for the product instances in these local virtual accounts, so that SSM On-Prem can report usage to the correct license pool in CSSM.

   a. Assigning a Smart Account and Virtual Account (SSM On-Prem UI), on page 179

b. Validating Devices (SSM On-Prem UI), on page 180

Note If your product instance is in a NAT set-up, also enable support for a NAT Setup when you enable device validation – both toggle switches are in the same window.

3. ***Product Instance Configuration***

   Where tasks are performed: Product Instance and the SSM On-Prem UI

   Remember to save any configuration changes on the product instance, by entering the **copy running-config startup-config** command in privileged EXEC mode.

   a. Ensuring Network Reachability for Product Instance-Initiated Communication, on page 180

   b. Retrieving the Transport URL (SSM On-Prem UI), on page 183

   c. Setting the Transport Type, URL, and Reporting Interval, on page 212

   The transport type configuration for CSLU and SSM On-Prem are the same (**license smart transport cslu** command in global configuration mode), but the URLs are different.

   d. An export-controlled license is supported only on certain models of the Cisco Catalyst Access, Core, and Aggregation Switches (See Authorization Code, on page 101). Complete these sub-steps only if you want to use an export-controlled license on supported platforms: Submitting an Authorization Code Request (SSM On-Prem UI), on page 190 and Manually Requesting and Auto-Installing a SLAC , on page 191.

4. ***Initial Usage Synchronization***

   Where tasks are performed: Product instance, SSM On-Prem UI, CSSM

   a. Synchronize the product instance with SSM On-Prem.

   On the product instance, enter the **license smart sync** {**all** | **local**} command, in privileged EXEC mode. This synchronizes the product instance with SSM On-Prem, to send and receive any pending data.

   ```
   Device(config)# license smart sync local
   ```

   You can verify this in the SSM On-Prem UI. Log in and select the **Smart Licensing** workspace. Navigate to the **Inventory** > **SL Using Policy** tab. In the **Alerts** column of the corresponding product instance, the following message is displayed: Usage report from product instance.

Note If you have not performed Step 2 above (Addition and Validation of Product Instances), completing this sub-step will add the product instance to the SSM On-Prem database.

   b. Synchronize usage information with CSSM (*choose one*):

   - Option 1:

     SSM On-Prem is connected to CSSM: In the SSM On-Prem UI, Smart Licensing workspace, nagivate to **Reports** > **Usage Schedules** > **Synchronize now with Cisco**.

   - Option 2:

SSM On-Prem is not connected to CSSM: See Exporting and Importing Usage Data (SSM On-Prem UI), on page 183.

Result:

You have completed initial usage synchronization. Product instance and license usage information is now displayed in SSM On-Prem.

For subsequent reporting, you have the following options:

- To synchronize data between the product instance and SSM On-Prem:

  Schedule periodic synchronization between the product instance and the SSM On-Prem, by configuring the reporting interval. Enter the **license smart usage interval** *interval_in_days* command in global configuration mode.

  The following applies only to Cisco IOS XE Amsterdam 17.3.6 and later releases of the 17.3.x train and Cisco IOS XE Bengaluru 17.6.4 and later releases of the 17.6.x train: in the product instance-initiated mode, the product instance does not send more than one RUM report a day. You can override this for an on-demand synchronization between the product instance and CSLU, by entering the **license smart sync** command in privileged EXEC mode.

  To know when the product instance will be sending the next RUM report, enter the **show license all** command in privileged EXEC mode and in the output, check the `Next report push:` field.

- To synchronize usage information with CSSM:
  - Schedule periodic synchronization with CSSM. In the SSM On-Prem UI, navigate to **Reports** > **Usage Schedules** > **Synchronization schedule with Cisco**. Enter the following frequency information and save:
    - **Days**: Refers to how *often* synchronization occurs. For example, if you enter 2, synchronization occurs once every two days.
    - **Time of Day:**: Refers to the time at which synchronization occurs, in the 24-hour notation system. For example, if you enter 14 hours and 0 minutes, synchronization occurs at 2 p.m. (1400) in your local time zone.
  - Upload and download the required files for reporting: Exporting and Importing Usage Data (SSM On-Prem UI), on page 183).

If you want to change the boot level license, see Configuring a Base or Add-On License , on page 215.

If you want to return an authorization code, see Returning an Authorization Code, on page 201.

Tasks for SSM On-Prem Instance-Initiated Communication

SSM On-Prem Installation → Product Instance Addition → Product Instance Configuration→ Initial Usage Synchronization

1. ***SSM On-Prem Installation***

   Where task is performed: A physical server such as a Cisco UCS C220 M3 Rack Server, or a hardware-based server that meets the necessary requirements.

   Download the file from Smart Software Manager > **Smart Software Manager On-Prem**.

Refer to the *Cisco Smart Software On-Prem Installation Guide* and the *Cisco Smart Software On-Prem User Guide* for help with installation.

Installation is complete when you have deployed SSM On-Prem, configured a common name on SSM On-Prem (**Security Widget** > **Certificates**), synchronized the NTP server (**Settings** widget > **Time Settings**), and created, registered, and synchronized (**Synchronization** widget) the SSM On-Prem local account with your Smart Account and Virtual Account in CSSM.

Note Licensing functions in the **On-Prem Licensing Workspace** are greyed-out until you complete the creation, registration, and synchronization of the local account with your Smart Account in CSSM. The *local account* synchronization with CSSM is for the SSM On-Prem instance to be known to CSSM, and is different from usage synchronization which is performed in **4. Initial Usage Synchronization** below.

2. ***Product Instance Addition***

   Where task is performed: SSM On-Prem UI

   Depending on whether you want to add a single product instance or multiple product instances, follow the corresponding sub-steps: Adding One or More Product Instances (SSM On-Prem UI), on page 184.

3. ***Product Instance Configuration***

   Where tasks are performed: Product Instance

   Remember to save any configuration changes on the product instance, by entering the **copy running-config startup-config** command in privileged EXEC mode.

   a. Ensuring Network Reachability for SSM On-Prem-Initiated Communication, on page 186

   b. An export-controlled license is supported only on certain models of the Cisco Catalyst Access, Core, and Aggregation Switches (See Authorization Code, on page 101). Complete these sub-steps only if you want to use an export-controlled license on supported platforms: Submitting an Authorization Code Request (SSM On-Prem UI), on page 190.

      The uploaded codes are applied to the product instances the next time SSM On-Prem runs an update. An initial usage synchronization with the product instance is being performed in Step 4 below so this will be completed then.

4. ***Initial Usage Synchronization***

   Where tasks are performed: SSM On-Prem, and CSSM

   a. Retrieve usage information from the product instance.

      In the SSM On-Prem UI, navigate to **Reports** > **Synchronisation pull schedule with the devices** > **Synchronise now with the device**.

      In the **Alerts** column, the following message is displayed: Usage report from product instance.

Tip It takes 60 seconds before synchronization is triggered. To view progress, navigate to the **On-Prem Admin Workspace**, and click the **Support Centre** widget. The system logs here display progress.

   b. Synchronize usage information with CSSM (*choose one*)

- Option 1:

  SSM On-Prem is connected to CSSM: In the SSM On-Prem UI, Smart Licensing workspace, nagivate to **Reports** > **Usage Schedules** > **Synchronize now with Cisco**.

- Option 2:

  SSM On-Prem is not connected to CSSM. See: Exporting and Importing Usage Data (SSM On-Prem UI), on page 183.

Result:

You have completed initial usage synchronization. Product instance and license usage information is now displayed in SSM On-Prem. SSM On-Prem automatically sends the ACK back to the product instance. To verify that the product instance has received the ACK, enter the **show license status** command in privileged EXEC mode, and in the output, check the date for the `Last ACK received` field.

For subsequent reporting, you have the following options:

- To retrieve usage information from the product instance, you can:
  - In the SSM On-Prem UI, Smart Licensing workspace, nagivate to **Reports** > **Usage Schedules** > **Synchronize now with Cisco**.
  - Schedule periodic retrieval of information from the product instance by configuring a frequency. In the SSM On-Prem UI, Smart Licensing workspace, nagivate to **Reports** > **Usage Schedules** > **Synchronisation pull schedule with the devices**. Enter values in the following fields:
    - **Days**: Refers to how *often* synchronization occurs. For example, if you enter 2, synchronization occurs once every two days.
    - **Time of Day:**: Refers to the time at which synchronization occurs, in the 24-hour notation system. For example, if you enter 14 hours and 0 minutes, synchronization occurs at 2 p.m. (1400).
  - Collect usage data from the product instance without being connected to CSSM. In the SSM On-Prem UI, Smart Licensing workspace, nagivate to **Inventory** > **SL Using Policy** tab. Select one or more product instances by enabling the coresponding check box. Click **Actions for Selected...** > **Collect Usage**. On-Prem connects to the selected Product Instance(s) and collects the usage reports. These usage reports are then stored in On-Prem's local library. These reports can then be transferred to Cisco if On-Prem is connected to Cisco, or (if you are not connected to Cisco) you can manually trigger usage collection by selecting **Export/Import All..** > **Export Usage to Cisco**.
- To synchronize usage information with CSSM, you can:
  - Schedule periodic synchronization with CSSM. In the SSM On-Prem UI, navigate to **Reports** > **Usage Schedules** > **Synchronization schedule with Cisco**. Enter the following frequency information and save:
    - **Days**: Refers to how *often* synchronization occurs. For example, if you enter 2, synchronization occurs once every two days.
    - **Time of Day:**: Refers to the time at which synchronization occurs, in the 24-hour notation system. For example, if you enter 14 hours and 0 minutes, synchronization occurs at 2 p.m. (1400).

- Upload and download the required files for reporting: Exporting and Importing Usage Data (SSM On-Prem UI), on page 183).

If you want to change the boot level license, see Configuring a Base or Add-On License , on page 215.

If you want to return an authorization code, see Returning an Authorization Code, on page 201.

Migrating to Smart Licensing Using Policy

To upgrade to Smart Licensing Using Policy, you must upgrade the software version (image) on the product instance to a supported version.

Before you Begin

Ensure that you have read the Upgrades, on page 118 section, to understand how Smart Licensing Using Policy handles various aspects of all earlier licensing models.

Smart Licensing Using Policy is introduced in Cisco IOS XE Amsterdam 17.3.2. This is therefore the minimum required version for Smart Licensing Using Policy.

Note that all the licenses that you are using prior to migration will be available after upgrade. This means that not only registered and authorized licenses (including reserved licenses), but also evaluation licenses will be migrated. The advantage with migrating registered and authorized licenses is that you will have fewer configuration steps to complete after migration, because your configuration is retained after upgrade (transport type configuration and configuration for connection to CSSM, all authorization codes). This ensures a smoother transition to the Smart Licensing Using Policy environment.

Device-led conversion is not supported for migration to Smart Licensing Using Policy.

Upgrading the Switch Software

See the corresponding release note for the upgrade procedure. If there are any general release-specific considerations, these are called-out in the corresponding release notes. For example, to upgrade to Cisco IOS XE Amsterdam 17.3.2, see *Release Notes for Cisco <platform name>, Cisco IOS XE Amsterdam 17.3.x*.

You can use the procedure to upgrade in install mode or with In-Service Software Upgrade (ISSU) (on supported platforms and supported releases).

Release Notes for Cisco Catalyst 9600 Series Switches: https://www.cisco.com/c/en/us/support/switches/catalyst-9600-series-switches/products-release-notes-list.html. See section *Upgrading the Switch Software*. ISSU is supported on this product instance.

After Upgrading the Software Version

- Complete topology implementation.

  If a transport mode is available in your pre-upgrade set-up, this is retained after you upgrade. Only in some cases, like with evaluation licenses or with licensing models where the notion of a transport type does not exist, the default (**cslu**) is applied - in these cases you may have a few more steps to complete before you are set to operate in the Smart Licensing Using Policy environment.

  No matter which licensing model you upgrade from, you can change the topology after upgrade.

- Synchronize license usage with CSSM

No matter which licensing model you are upgrading from and no matter which topology you implement, synchronize your usage information with CSSM. For this you have to follow the reporting method that applies to the topology you implement. This initial synchronization ensures that up-to-date usage information is reflected in CSSM and a custom policy (if available), is applied. The policy that is applicable after this synchronization also indicates subsequent reporting requirements. These rules are also tabled here: How Upgrade Affects Reporting for Existing Licenses, on page 119

Note After initial usage synchronization is completed, reporting is required only if the policy, or, system messages indicate that it is.

Sample Migration Scenarios

Sample migration scenarios have been provided considering the various existing licensing models and licenses. All scenarios provide sample outputs before and after migration, any CSSM Web UI changes to look out for (as an indicator of a successful migration or further action), and how to identify and complete any necessary post-migration steps.

Note For SSM On-Prem, the sequence in which you perform the various upgrade-related activities is crucial. So only for this scenario, the migration sequence has been provided - and not an example.

Example: Smart Licensing to Smart Licensing Using Policy

The following is an example of a Cisco Catalyst 9500 switch migrating from Smart Licensing to Smart Licensing Using Policy. This is a High Availability set-up with an active and standby.

- Table 10: Smart Licensing to Smart Licensing Using Policy: show Commands
- The CSSM Web UI After Migration, on page 142
- Reporting After Migration, on page 145

The **show** command outputs below call-out key fields to check, before and after migration.

Table 10: Smart Licensing to Smart Licensing Using Policy: show Commands

Before Upgrade	After Upgrade
show license summary (Smart Licensing) The `Status` and `License Authorization` fields show that the license is `REGISTERED` and `AUTHORIZED`.	**show license summary** (Smart Licensing Using Policy) The `Status` field shows that the licenses are now `IN USE` instead of registered and authorized.

Before Upgrade	After Upgrade
Device# **show license summary** Smart Licensing is ENABLED Registration: **Status: REGISTERED** Smart Account: SA-Eg-Company-01 Virtual Account: SLE_Test Export-Controlled Functionality: ALLOWED Last Renewal Attempt: None Next Renewal Attempt: Mar 21 11:08:58 2021 PST License Authorization: **Status: AUTHORIZED** Last Communication Attempt: SUCCEEDED Next Communication Attempt: Oct 22 11:09:07 2020 PST License Usage: License Entitlement tag Count Status --- C9500 Network Advantage (C9500 Network Advantage) 2 AUTHORIZED C9500-DNA-16X-A (C9500-16X DNA Advantage) 2 AUTHORIZED	Device# **show license summary** License Usage: License Entitlement tag Count Status --- network-advantage (C9500 Network Advantage) 2 **IN USE** dna-advantage (C9500-16X DNA Advantage) 2 **IN USE**

show license usage (Smart Licensing)	**show license usage** (Smart Licensing Using Policy)
	The license counts remain the same. The `Enforcement Type` field displays NOT ENFORCED, beca licenses that were being used prior to upgrade were unenforce
Device# **show license usage** License Authorization: Status: AUTHORIZED on Sep 22 11:09:07 2020 PST C9500 Network Advantage (C9500 Network Advantage): Description: C9500 Network Advantage **Count: 2** Version: 1.0 **Status: AUTHORIZED** Export status: NOT RESTRICTED C9500-DNA-16X-A (C9500-16X DNA Advantage): Description: C9500-DNA-16X-A **Count: 2** Version: 1.0 **Status: AUTHORIZED** Export status: NOT RESTRICTED	Device# **show license usage** License Authorization: Status: Not Applicable network-advantage (C9500 Network Advantage): Description: network-advantage **Count: 2** Version: 1.0 Status: IN USE Export status: NOT RESTRICTED Feature Name: network-advantage Feature Description: network-advantage **Enforcement type: NOT ENFORCED** **License type: Perpetual** dna-advantage (C9500-16X DNA Advantage): Description: C9500-16X DNA Advantage **Count: 2** Version: 1.0 Status: IN USE Export status: NOT RESTRICTED Feature Name: dna-advantage Feature Description: C9500-16X DNA Advantage **Enforcement type: NOT ENFORCED** **License type: Subscription**

show license status (Smart Licensing)	**show license status** (Smart Licensing Using Policy) The `Transport:` field: A transport type was configured and therefore retained after upgrade. The `Policy:` header and details: A custom policy was available in the Smart Account or Virtual Account – this has also been automatically installed on the product instance. (After establishing trust, CSSM returns a policy. The policy is then automatically installed.) The `Usage Reporting:` header: The `Next report push:` field provides information about when the product instance will send the next RUM report to CSSM. The `Trust Code Installed:` field: The ID token is successfully converted and a trusted connected has been established with CSSM.

```
Device# show license status

Smart Licensing is ENABLED
Utility:
Status: DISABLED
Data Privacy:
Sending Hostname: yes
Callhome hostname privacy: DISABLED
Smart Licensing hostname privacy: DISABLED
Version privacy: DISABLED
Transport:
Type: Callhome
Registration:
Status: REGISTERED
Smart Account: Eg-SA-01
Virtual Account: Eg-VA-01
Export-Controlled Functionality: ALLOWED
Initial Registration: SUCCEEDED on Sep 22 11:08:58 2020 PST
Last Renewal Attempt: None
Next Renewal Attempt: Mar 21 11:08:57 2021 PST
Registration Expires: Sep 22 11:04:23 2021 PST
License Authorization:
Status: AUTHORIZED on Sep 22 11:09:07 2020 PST
Last Communication Attempt: SUCCEEDED on Sep 22 11:09:07 2020
PST
Next Communication Attempt: Oct 22 11:09:06 2020 PST
Communication Deadline: Dec 21 11:04:34 2020 PST
Export Authorization Key:
Features Authorized:
<none>
Miscellaneus:
Custom Id: <empty>
```

```
Device# show license status

Utility:
  Status: DISABLED
Smart Licensing Using Policy:
  Status: ENABLED
Data Privacy:
  Sending Hostname: yes
    Callhome hostname privacy: DISABLED
    Smart Licensing hostname privacy: DISABLED

  Version privacy: DISABLED
Transport:
  Type: Callhome
Policy:
  Policy in use: Merged from multiple sources.

  Reporting ACK required: yes (CISCO default)
  Unenforced/Non-Export Perpetual Attributes:
    First report requirement (days): 365 (CISCO
 default)
    Reporting frequency (days): 0 (CISCO
default)
    Report on change (days): 90 (CISCO default)

  Unenforced/Non-Export Subscription
Attributes:
    First report requirement (days): 90 (CISCO
 default)
    Reporting frequency (days): 90 (CISCO
default)
    Report on change (days): 90 (CISCO default)

  Enforced (Perpetual/Subscription) License
Attributes:
    First report requirement (days): 0 (CISCO
 default)
    Reporting frequency (days): 0 (CISCO
default)
    Report on change (days): 0 (CISCO default)

  Export (Perpetual/Subscription) License
Attributes:
    First report requirement (days): 0 (CISCO
 default)
    Reporting frequency (days): 0 (CISCO
default)
    Report on change (days): 0 (CISCO default)
Miscellaneous:
  Custom Id: <empty>
Usage Reporting:
  Last ACK received: Sep 22 13:49:38 2020 PST
  Next ACK deadline: Dec 21 12:02:21 2020 PST
  Reporting push interval: 30 days
  Next ACK push check: Sep 22 12:20:34 2020
PST
  Next report push: Oct 22 12:05:43 2020 PST
  Last report push: Sep 22 12:05:43 2020 PST
  Last report file write: <none>
Trust Code Installed:
  Active: PID:C9500-16X,SN:FCW2233A5ZV
  INSTALLED on Sep 22 12:02:20 2020 PST
  Standby: PID:C9500-16X,SN:FCW2233A5ZY
  INSTALLED on Sep 22 12:02:20 2020 PST
```

show license udi (Smart Licensing)	**show license udi** (Smart Licensing Using Policy)
	This is a High Availability set-up and the command displays all UDIs in the set-up.

`Device# show license udi`
 	
`UDI: PID:C9500-16X,SN:FCW2233A5ZV`
`HA UDI List:`
`Active:PID:C9500-16X,SN:FCW2233A5ZV`
`Standby:PID:C9500-16X,SN:FCW2233A5ZY`
	`Device# show license udi`
	
	`UDI: PID:C9500-16X,SN:FCW2233A5ZV`
	`HA UDI List:`
	`Active:PID:C9500-16X,SN:FCW2233A5ZV`
	`Standby:PID:C9500-16X,SN:FCW2233A5ZY`

The CSSM Web UI After Migration

Log in to the CSSM Web UI at https://software.cisco.com. Under **Smart Software Licensing**, click the **Manage licenses**link.

Click the **Inventory** tab. From the **Virtual Account** drop-down list, choose the required virtual account. Click the **Product Instances** tab.

Registered licenses in the Smart Licensing environment were displayed with the hostname of the product instance in the Name column. After upgrade to Smart Licensing Using Policy, they are displayed with the UDI of the product instance. All migrated UDIs are displayed. In this example, they are PID:C9500-16X,SN:FCW2233A5ZV and PID:C9500-16X,SN:FCW2233A5ZY.

Only the active product instance reports usage, therefore PID:C9500-16X,SN:FCW2233A5ZV displays license consumption information under **License Usage**. The standby does not report usage and the **License Usage** section for the standby displays No Records Found.

It is always the active that reports usage, so if the active in this High Availabilty set-up changes, the new active product instance will display license consumption information and report usage.

Figure 9: Smart Licensing to Smart Licensing Using Policy: Active and Standby Product Instances After Migration

Figure 10: Smart Licensing to Smart Licensing Using Policy: UDI and License Usage under Active Product Instance

Reporting After Migration

The product instance sends the next RUM report to CSSM, based on the policy.

If you want to change your reporting interval to report more frequently: on the product instance, configure the **license smart usage interval** command. For syntax details see the *license smart (global config)* command in the Command Reference for the corresponding release.

Example: RTU Licensing to Smart Licensing Using Policy

The following is an example of a Cisco Catalyst 9300 switch migrating from Right-to-Use (RTU) Licensing to Smart Licensing Using Policy. This is a set-up with an active and members.

RTU Licensing is available on Cisco Catalyst 9300, 9400, and 9500 Series Switches until Cisco IOS XE Fuji 16.8.x. Smart Licensing was introduced starting from Cisco IOS XE Fuji 16.9.1.

When the software version is upgraded to one that supports Smart Licensing Using Policy, all licenses are displayed as IN USE and the `Cisco default` policy is applied on the product instance. If any add-on licenses are used, the `Cisco default` policy requires usage reporting in 90 days. No export-controlled or enforced licenses were available on Cisco Catalyst Access, Core, and Aggregation Switches when the RTU licensing model was supported, and therefore no functionality is lost.

- Table 11: RTU Licensing to Smart Licensing Using Policy: show Commands
- The CSSM Web UI After Migration, on page 147
- Reporting After Migration, on page 148

The table below calls out key changes or new fields to check for in the **show** command outputs, after upgrade to Smart Licensing Using Policy

Table 11: RTU Licensing to Smart Licensing Using Policy: show Commands

Before Upgrade	After Upgrade

show license right-to-use summary (RTU Licensing)
 	
`Device# show license right-to-use summary`
`License Name Type Period left`
`---`
`network-essentials Permanent Lifetime`
`dna-essentials Subscription CSSM Managed`
`---`
 	
`License Level In Use: network-essentials+dna-essentials Subscription`
`License Level on Reboot:`
`network-essentials+dna-essentials Subscription`
	show license summary (Smart Licensing Using Policy)
	
	All licenses are migrated and `IN USE`.
	
	`Device#show license summary`
	`License Usage:`
	`License Entitlement Tag Count Status`
	`---`
	`network-essentials (C9300-24 Network Essen...) 2 IN USE`
	`dna-essentials (C9300-24 DNA Essentials) 2 IN USE`
	`network-essentials (C9300-48 Network Essen...) 1 IN USE`
	`dna-essentials (C9300-48 DNA Essentials) 1 IN USE`

show license right-to-use usage (Smart Licensing)

```
Device# show license right-to-use usage

Slot# License Name Type usage-duration(y:m:d) In-Use
EULA
-------------------------------------------------------
1 network-essentials Permanent 00:00:00 yes yes
1 network-essentials Evaluation 00:00:00 no no
1 network-essentials Subscription 00:00:00 no no
1 network-advantage Permanent 00:00:00 no no
1 network-advantage Evaluation 00:00:00 no no
1 network-advantage Subscription 00:00:00 no no
1 dna-essentials Evaluation 00:00:00 no no
1 dna-essentials Subscription 00:00:00 yes yes
1 dna-advantage Evaluation 00:00:00 no no
1 dna-advantage Subscription 00:00:00 no no
-------------------------------------------------------
Slot# License Name Type usage-duration(y:m:d) In-Use
EULA
-------------------------------------------------------
2 network-essentials Permanent 00:00:00 yes yes
2 network-essentials Evaluation 00:00:00 no no
2 network-essentials Subscription 00:00:00 no no
2 network-advantage Permanent 00:00:00 no no
2 network-advantage Evaluation 00:00:00 no no
2 network-advantage Subscription 00:00:00 no no
2 dna-essentials Evaluation 00:00:00 no no
2 dna-essentials Subscription 00:00:00 yes yes
2 dna-advantage Evaluation 00:00:00 no no
2 dna-advantage Subscription 00:00:00 no no
-------------------------------------------------------
Slot# License Name Type usage-duration(y:m:d) In-Use
EULA
-------------------------------------------------------
3 network-essentials Permanent 00:00:00 yes yes
3 network-essentials Evaluation 00:00:00 no no
3 network-essentials Subscription 00:00:00 no no
3 network-advantage Permanent 00:00:00 no no
3 network-advantage Evaluation 00:00:00 no no
3 network-advantage Subscription 00:00:00 no no
3 dna-essentials Evaluation 00:00:00 no no
3 dna-essentials Subscription 00:00:00 yes yes
3 dna-advantage Evaluation 00:00:00 no no
3 dna-advantage Subscription 00:00:00 no no
-------------------------------------------------------
```

show license usage (Smart Licensing Using Policy)

All licenses (permanent, subscription) have been migrated and the licenses are now `IN USE` and have types `Perpetual` and `Subscription`.

The `Enforcement Type` field displays NOT ENFORCED, because all the licenses that were being using prior to upgrade, were unenforced licenses.

```
Device# show license usage

License Authorization:
  Status: Not Applicable
network-advantage (C9300-24 Network Advantage):
  Description: C9300-24 Network Advantage
  Count: 2
  Version: 1.0
  Status: IN USE
  Export status: NOT RESTRICTED
  Feature Name: network-advantage
  Feature Description: C9300-24 Network Advantage
  Enforcement type: NOT ENFORCED
  License type: Perpetual
dna-advantage (C9300-24 DNA Advantage):
  Description: C9300-24 DNA Advantage
  Count: 2
  Version: 1.0
  Status: IN USE
  Export status: NOT RESTRICTED
  Feature Name: dna-advantage
  Feature Description: C9300-24 DNA Advantage
  Enforcement type: NOT ENFORCED
  License type: Subscription
network-advantage (C9300-48 Network Advantage):
  Description: C9300-48 Network Advantage
  Count: 1
  Version: 1.0
  Status: IN USE
  Export status: NOT RESTRICTED
  Feature Name: network-advantage
  Feature Description: C9300-48 Network Advantage
  Enforcement type: NOT ENFORCED
  License type: Perpetual
dna-advantage (C9300-48 DNA Advantage):
  Description: C9300-48 DNA Advantage
  Count: 1
  Version: 1.0
  Status: IN USE
  Export status: NOT RESTRICTED
  Feature Name: dna-advantage
  Feature Description: C9300-48 DNA Advantage
  Enforcement type: NOT ENFORCED
  License type: Subscription
```

show license right-to-use (RTU Licensing)

```
Device# show license right-to-use
Slot# License Name Type Period left
--------------------------------------------------
1 network-essentials Permanent Lifetime
1 dna-essentials Subscription CSSM Managed
--------------------------------------------------
License Level on Reboot:
network-essentials+dna-essentials
Subscription

Slot# License Name Type Period left
--------------------------------------------------
2 network-essentials Permanent Lifetime
2 dna-essentials Subscription CSSM Managed
--------------------------------------------------
License Level on Reboot:
network-essentials+dna-essentials
Subscription

Slot# License Name Type Period left
--------------------------------------------------
3 network-essentials Permanent Lifetime
3 dna-essentials Subscription CSSM Managed
--------------------------------------------------
License Level on Reboot:
network-essentials+dna-essentials
Subscription
```

show license status (Smart Licensing Using Policy)

The `Transport:` field displays its off.

The `Trust Code Installed:` field displays that a trust code is not installed.

Under the `Usage Reporting:` header, the `Next report push:` field provides information about when the next RUM report must be sent to CSSM.

```
Device# show license status
Utility:
  Status: DISABLED
Smart Licensing Using Policy:
  Status: ENABLED
Data Privacy:
  Sending Hostname: yes
    Callhome hostname privacy: DISABLED
    Smart Licensing hostname privacy: DISABLED
  Version privacy: DISABLED
Transport:
  Type: Transport Off
Policy:
  Policy in use: Merged from multiple sources.
  Reporting ACK required: yes (CISCO default)
  Unenforced/Non-Export Perpetual Attributes:
    First report requirement (days): 365 (CISCO default)

    Reporting frequency (days): 0 (CISCO default)
    Report on change (days): 90 (CISCO default)
  Unenforced/Non-Export Subscription Attributes:
    First report requirement (days): 90 (CISCO default)

    Reporting frequency (days): 90 (CISCO default)
    Report on change (days): 90 (CISCO default)
  Enforced (Perpetual/Subscription) License Attributes:

    First report requirement (days): 0 (CISCO default)
    Reporting frequency (days): 0 (CISCO default)
    Report on change (days): 0 (CISCO default)
  Export (Perpetual/Subscription) License Attributes:
    First report requirement (days): 0 (CISCO default)
    Reporting frequency (days): 0 (CISCO default)
    Report on change (days): 0 (CISCO default)
Miscellaneous:
  Custom Id: <empty>
Usage Reporting:
  Last ACK received: <none>
  Next ACK deadline: Jan 26 10:27:59 2021 PST
  Reporting push interval: 20  days
  Next ACK push check: <none>
  Next report push: Oct 28 10:29:59 2020 PST
  Last report push: <none>
  Last report file write: <none>
Trust Code Installed: <none>
```

The CSSM Web UI After Migration

No changes in the CSSM Web UI.

Reporting After Migration

Implement any one of the supported topologies, and fulfil reporting requirements. See Supported Topologies, on page 105 and How to Configure Smart Licensing Using Policy: Workflows by Topology , on page 123. The reporting method you can use depends on the topology you implement.

Example: SLR to Smart Licensing Using Policy

The following is an example of a Cisco Catalyst 9500 switch migrating from Specific License Reservation (SLR) to Smart Licensing Using Policy. This is a High Availability set-up with an active and standby.

The license conversion is automatic and authorization codes are migratied. No further action is required to complete migration. After migration the No Connectivity to CSSM and No CSLU, on page 112 topology is effective. For information about the SLR authorization code in the Smart Licensing Using Policy environment, see Authorization Code, on page 101.

- Table 12: SLR to Smart Licensing Using Policy: show Commands
- The CSSM Web UI After Migration, on page 154
- Reporting After Migration, on page 157

The **show** command outputs below call-out key fields to check, before and after migration.

Table 12: SLR to Smart Licensing Using Policy: show Commands

Before Upgrade	After Upgrade

show license summary (SLR)
 	
The `Registration` and `License Authorization` status fields show that the license was `REGISTERED - SPECIFIC LICENSE RESERVATION` and `AUTHORIZED - RESERVED`.
 	
`Device# show license summary`
 	
`Smart Licensing is ENABLED`
`License Reservation is ENABLED`
`Registration:`
`  Status: REGISTERED - SPECIFIC LICENSE RESERVATION`
`Export-Controlled Functionality: ALLOWED`
`License Authorization:`
`  Status: AUTHORIZED - RESERVED`
`License Usage:`
`License  Entitlement tag  Count Status`
`---`
`C9500 Network Advantage(C9500 Network Advantage)  2 AUTHORIZED`
`C9500-DNA-16X-A  (C9500-16X DNA Advantage)  2 AUTHORIZED`
	show license summary (Smart Licensing Using Policy)
	
	The `Status` field shows that the licenses are now `IN USE` instead of registered and authorized.
	
	`Device# show license summary`
	
	`License Reservation is ENABLED`
	`License Usage:`
	`License  Entitlement tag  Count Status`
	`---`
	`network-advantage(C9500 Network Advantage)  2 IN USE`
	`dna-advantage  (C9500-16X DNA Advantage)  2 IN USE`

show license reservation (SLR)	**show license all** (Smart Licensing Using Policy) The `License Authorizations` header: shows that base (C9500 Network Advantage) and add-on (C9500-DNA-16X-A) licenses on the active and standby product instances were authorized with Specific License Reservation. The `Authorization type:` field shows SPECIFIC INSTALLED. The `Last Confirmation code:` field: shows that the SLR authorization code is successfully migrated for the active and standby product instances in the High Availability set-up.

```
Device# show license reservation
License reservation: ENABLED
Overall status:
  Active: PID:C9500-16X,SN:FCW2233A5ZV
       Reservation status: SPECIFIC INSTALLED on Aug 31
 10:15:01 2020 PDT
       Export-Controlled Functionality: ALLOWED
       Last Confirmation code: 4bfbea7f
  Standby: PID:C9500-16X,SN:FCW2233A5ZY
       Reservation status: SPECIFIC INSTALLED on Aug 31
 10:15:01 2020 PDT
       Export-Controlled Functionality: ALLOWED
       Last Confirmation code: 9394f196
Specified license reservations:
  C9500 Network Advantage (C9500 Network Advantage):
    Description: C9500 Network Advantage
    Total reserved count: 2
    Term information:
      Active: PID:C9500-16X,SN:FCW2233A5ZV
        License type: PERPETUAL
          Term Count: 1
      Standby: PID:C9500-16X,SN:FCW2233A5ZY
        License type: PERPETUAL
          Term Count: 1
  C9500-DNA-16X-A (C9500-16X DNA Advantage):
    Description: C9500-DNA-16X-A
    Total reserved count: 2
    Term information:
      Active: PID:C9500-16X,SN:FCW2233A5ZV
        License type: TERM
          Start Date: 2020-MAR-17 UTC
          End Date: 2021-MAR-17 UTC
          Term Count: 1
      Standby: PID:C9500-16X,SN:FCW2233A5ZY
```

```
Device# show license reservation

Smart Licensing Status
======================
Smart Licensing is ENABLED
License Reservation is ENABLED
Export Authorization Key:
  Features Authorized:
    <none>
Utility:
  Status: DISABLED
Smart Licensing Using Policy:
  Status: ENABLED
Data Privacy:
  Sending Hostname: yes
    Callhome hostname privacy: DISABLED
    Smart Licensing hostname privacy: DISABLED
  Version privacy: DISABLED
Transport:
  Type: Transport Off
Miscellaneous:
  Custom Id: <empty>
Policy:
  Policy in use: Merged from multiple sources.
  Reporting ACK required: yes (CISCO default)
  Unenforced/Non-Export Perpetual Attributes:
    First report requirement (days): 365 (CISCO default)

    Reporting frequency (days): 0 (CISCO default)
    Report on change (days): 90 (CISCO default)
  Unenforced/Non-Export Subscription Attributes:
    First report requirement (days): 90 (CISCO default)

    Reporting frequency (days): 90 (CISCO default)
    Report on change (days): 90 (CISCO default)
  Enforced (Perpetual/Subscription) License Attributes:

    First report requirement (days): 0 (CISCO default)
    Reporting frequency (days): 0 (CISCO default)
    Report on change (days): 0 (CISCO default)
  Export (Perpetual/Subscription) License Attributes:
    First report requirement (days): 0 (CISCO default)
    Reporting frequency (days): 0 (CISCO default)
    Report on change (days): 0 (CISCO default)
Usage Reporting:
  Last ACK received: <none>
  Next ACK deadline: Nov 29 10:50:05 2020 PDT
  Reporting Interval: 30
  Next ACK push check: <none>
  Next report push: Aug 31 10:52:05 2020 PDT
  Last report push: <none>
  Last report file write: <none>
Trust Code Installed: <none>
License Usage
=============
network-advantage (C9500 Network Advantage):
  Description: network-advantage
  Count: 2
  Version: 1.0
  Status: IN USE
  Export status: NOT RESTRICTED
  Feature Name: network-advantage
  Feature Description: network-advantage
  Enforcement type: NOT ENFORCED
```

```
  License type: Perpetual
  Reservation:
    Reservation status: SPECIFIC INSTALLED
    Total reserved count: 2
dna-advantage (C9500-16X DNA Advantage):
  Description: C9500-16X DNA Advantage
  Count: 2
  Version: 1.0
  Status: IN USE
  Export status: NOT RESTRICTED
  Feature Name: dna-advantage
  Feature Description: C9500-16X DNA Advantage
  Enforcement type: NOT ENFORCED
  License type: Subscription
  Reservation:
    Reservation status: SPECIFIC INSTALLED
    Total reserved count: 2
Product Information
==================
UDI: PID:C9500-16X,SN:FCW2233A5ZV
HA UDI List:
    Active:PID:C9500-16X,SN:FCW2233A5ZV
    Standby:PID:C9500-16X,SN:FCW2233A5ZY
Agent Version
=============
Smart Agent for Licensing: 5.0.5_rel/42
License Authorizations
======================
Overall status:
  Active: PID:C9500-16X,SN:FCW2233A5ZV
     Status: SPECIFIC INSTALLED on Aug 31 10:15:01 2020
 PDT
      Last Confirmation code: 4bfbea7f
  Standby: PID:C9500-16X,SN:FCW2233A5ZY
     Status: SPECIFIC INSTALLED on Aug 31 10:15:01 2020
 PDT
      Last Confirmation code: 9394f196
Specified license reservations:
  C9500 Network Advantage (C9500 Network Advantage):
    Description: C9500 Network Advantage
    Total reserved count: 2
    Enforcement type: NOT ENFORCED
    Term information:
      Active: PID:C9500-16X,SN:FCW2233A5ZV
        Authorization type: SPECIFIC INSTALLED on Aug
31 10:15:01 2020 PDT
        License type: PERPETUAL
          Term Count: 1
      Standby: PID:C9500-16X,SN:FCW2233A5ZY
        Authorization type: SPECIFIC INSTALLED on Aug
31 10:15:01 2020 PDT
        License type: PERPETUAL
          Term Count: 1
  C9500-DNA-16X-A (C9500-16X DNA Advantage):
    Description: C9500-DNA-16X-A
    Total reserved count: 2
    Enforcement type: NOT ENFORCED
    Term information:
      Active: PID:C9500-16X,SN:FCW2233A5ZV
        Authorization type: SPECIFIC INSTALLED on Aug
31 10:15:01 2020 PDT
        License type: PERPETUAL
          Term Count: 1
      Standby: PID:C9500-16X,SN:FCW2233A5ZY
```

	Authorization type: SPECIFIC INSTALLED on Aug 31 10:15:01 2020 PDT License type: PERPETUAL Term Count: 1 Purchased Licenses: No Purchase Information Available Derived Licenses: Entitlement Tag: regid.2017-03.com.cisco.advantagek9-Nyquist-C9500, 1.0_f1563759-2e03-4a4c-bec5-5feec525a12c Entitlement Tag: regid.2017-07.com.cisco.C9500-DNA-16X-A, 1.0_ef3574d1-156b-486a-864f-9f779ff3ee49
show license status (SLR)	**show license status** (Smart Licensing Using Policy) The `Transport:` header: `Type:` displays that the transport type is set to off. The `Usage Reporting:` header: `Next report push:` field displays if and when the next RUM report must be uploaded to CSSM.

```
Device# show license status

Smart Licensing is ENABLED
Utility:
  Status: DISABLED
License Reservation is ENABLED
Data Privacy:
  Sending Hostname: yes
    Callhome hostname privacy: DISABLED
    Smart Licensing hostname privacy: DISABLED
  Version privacy: DISABLED
Transport:
  Type: Callhome
Registration:
  Status: REGISTERED - SPECIFIC LICENSE RESERVATION
  Export-Controlled Functionality: ALLOWED
  Initial Registration: SUCCEEDED on Aug 31 11:07:39
2020 PDT
License Authorization:
  Status: AUTHORIZED - RESERVED on Aug 31 10:15:01 2020
 PDT
Export Authorization Key:
  Features Authorized:
    <none>
        License type: TERM
          Start Date: 2020-MAR-17 UTC
          End Date: 2021-MAR-17 UTC
          Term Count: 1
```

```
Device# show license status

Utility:
  Status: DISABLED
License Reservation is ENABLED
Data Privacy:
  Sending Hostname: yes
    Callhome hostname privacy: DISABLED
    Smart Licensing hostname privacy: DISABLED
   Version privacy: DISABLED
Transport:
  Type: Transport Off
Policy:
  Policy in use: Merged from multiple sources.
  Reporting ACK required: yes (CISCO default)
  Unenforced/Non-Export Perpetual Attributes:
    First report requirement (days): 365 (CISCO default)

    Reporting frequency (days): 0 (CISCO default)
    Report on change (days): 90 (CISCO default)
  Unenforced/Non-Export Subscription Attributes:
    First report requirement (days): 90 (CISCO default)

    Reporting frequency (days): 90 (CISCO default)
    Report on change (days): 90 (CISCO default)
  Enforced (Perpetual/Subscription) License Attributes:

    First report requirement (days): 0 (CISCO default)
    Reporting frequency (days): 0 (CISCO default)
    Report on change (days): 0 (CISCO default)
  Export (Perpetual/Subscription) License Attributes:
    First report requirement (days): 0 (CISCO default)
    Reporting frequency (days): 0 (CISCO default)
    Report on change (days): 0 (CISCO default)
Miscellaneous:
  Custom Id: <empty>
Usage Reporting:
  Last ACK received: <none>
  Next ACK deadline: Nov 29 10:50:05 2020 PDT
  Reporting Interval: 30
  Next ACK push check: <none>
  Next report push: Aug 31 10:52:05 2020 PDT
  Last report push: <none>
  Last report file write: <none>
Trust Code Installed: <none>
```

The CSSM Web UI After Migration

In CSSM, there are no changes in the **Product Instances** tab. The Last Contact column displays "Reserved Licenses" since there has been no usage reporting yet.

After the requisite RUM report is uploaded and acknowledged "Reserved Licenses" and license usage will only be seen in the Active PID product Instance.

Figure 11: SLR to Smart Licensing Using Policy: Active and Standby Product Instances After Migration, Before Reporting

Figure 12: SLR to Smart Licensing Using Policy: Active and Standby Product Instances After Migration, After Reporting

Reporting After Migration

SLR licenses require reporting only when there is a change in licensing consumption (For example, when using an add-on license which is for specified term). The policy (**show license status**) indicates this, or you will receive syslog messages about this.

Since all communication to and from the product instance is disabled, to report license usage you must save RUM reports to a file and upload it to CSSM (from a workstation that has connectivity to the internet, and Cisco):

1. Generate and save RUM reports.

   Enter the **license smart save usage** command in provileged EXEC mode. In the example below, all RUM reports are saved to the flash memory of the product instance, in file `all_rum.txt`. For syntax details see the *license smart (privileged EXEC)* command in the Command Reference for the corresponding release. In the example, the file is first saved to bootflash and then copied to a TFTP location:

   ```
   Device# license smart save usage all file bootflash:all_rum.txt
   Device# copy bootflash:all_rum.txt tftp://10.8.0.6/all_rum.txt
   ```

2. Upload usage data to CSSM: Uploading Data or Requests to CSSM and Downloading a File, on page 210.
3. Install the ACK on the product instance: Installing a File on the Product Instance, on page 211.

Example: Evaluation or Expired to Smart Licensing Using Policy

The following is an example of a Cisco Catalyst 9500 switch with evaluation licenses (Smart Licensing) that are migrated to Smart Licensing Using Policy.

The notion of evaluation licenses does not apply to Smart Licensing Using Policy. When the software version is upgraded to one that supports Smart Licensing Using Policy, all licenses are displayed as IN USE and the Cisco default policy is applied to the product instance. No export-controlled or enforced licenses were available on Cisco Catalyst Access, Core, and Aggregation Switches when the earlier licensing models were effective, and therefore no functionality is lost.

- Table 13: Evaluation or Expired to Smart Licensing Using Policy: show Commands
- The CSSM Web UI After Migration, on page 159
- Reporting After Migration, on page 159

The table below calls out key changes or new fields to check for in the **show** command outputs, after upgrade to Smart Licensing Using Policy

Table 13: Evaluation or Expired to Smart Licensing Using Policy: show Commands

Before Upgrade	After Upgrade
show license summary (Smart Licensing, Evaluation Mode) Licenses are UNREGISTERED and in `EVAL MODE`.	**show license summary** (Smart Licensing Using Policy) All licenses are migrated and `IN USE`. There are no EVAL MODE licenses.

Before Upgrade	After Upgrade

`Device# show license summary`
 	
`Smart Licensing is ENABLED`
`Registration:`
`Status: UNREGISTERED`
`Export-Controlled Functionality: NOT ALLOWED`
`License Authorization:`
`Status: EVAL MODE`
`Evaluation Period Remaining: 89 days, 21 hours, 37 minutes,`
`30 seconds`
`License Usage:`
`License Entitlement tag Count Status`
`---`
`(C9500 Network Advantage) 2 EVAL MODE`
`(C9500-16X DNA Advantage) 2 EVAL MODE`
	`Device# show license summary`
	
	`License Usage:`
	`License Entitlement tag Count Status`
	`---`
	`network-advantage (C9500 Network Advantage) 2 IN USE`
	`dna-advantage (C9500-16X DNA Advantage) 2 IN USE`

show license usage (Smart Licensing, Evaluation Mode)
 	
`Device# show license usage`
 	
`License Authorization:`
`Status: EVAL MODE`
`Evaluation Period Remaining: 89 days, 21 hours, 37 minutes,`
`21 seconds`
`(C9500 Network Advantage):`
`Description:`
`Count: 2`
`Version: 1.0`
`Status: EVAL MODE`
`Export status: NOT RESTRICTED`
`(C9500-16X DNA Advantage):`
`Description:`
`Count: 2`
`Version: 1.0`
`Status: EVAL MODE`
`Export status: NOT RESTRICTED`
	show license usage (Smart Licensing Using Policy)
	
	The `Enforcement Type` field displays NOT ENFORCED, because all the licenses that were being using prior to upgrade, were unenforced licenses.
	
	`Device# show license usage`
	`License Authorization:`
	`Status: Not Applicable`
	`network-advantage (C9500 Network Advantage):`
	`Description: network-advantage`
	`Count: 2`
	`Version: 1.0`
	`Status: IN USE`
	`Export status: NOT RESTRICTED`
	`Feature Name: network-advantage`
	`Feature Description: network-advantage`
	`Enforcement type: NOT ENFORCED`
	`License type: Perpetual`
	`dna-advantage (C9500-16X DNA Advantage):`
	`Description: C9500-16X DNA Advantage`
	`Count: 2`
	`Version: 1.0`
	`Status: IN USE`
	`Export status: NOT RESTRICTED`
	`Feature Name: dna-advantage`
	`Feature Description: C9500-16X DNA Advantage`
	`Enforcement type: NOT ENFORCED`
	`License type: Subscription`

show license status (Smart Licensing, Evaluation Mode)	**show license status** (Smart Licensing Using Policy) The `Transport:` field displays that its off. The `Policy` field shows that the Cisco default policy is applied The `Trust Code Installed:` field displays that a trust code is not installed. The `Usage Reporting:` header: The `Next report push:` field provides information about when the next RUM report must be sent to CSSM.

```
Switch# show license status

Smart Licensing is ENABLED
Utility:
Status: DISABLED
Data Privacy:
Sending Hostname: yes
Callhome hostname privacy: DISABLED
Smart Licensing hostname privacy: DISABLED
Version privacy: DISABLED
Transport:
Type: Callhome
Registration:
Status: UNREGISTERED
Export-Controlled Functionality: NOT ALLOWED
License Authorization:
Status: EVAL MODE
Evaluation Period Remaining: 89 days, 21 hours, 37
minutes, 15 seconds
Export Authorization Key:
Features Authorized:
<none>
Miscellaneus:
Custom Id: <empty>
```

```
Switch# show license status

Utility:
  Status: DISABLED
Smart Licensing Using Policy:
  Status: ENABLED
Data Privacy:
  Sending Hostname: yes
    Callhome hostname privacy: DISABLED
    Smart Licensing hostname privacy: DISABLED
  Version privacy: DISABLED
Transport:
  Type: Transport Off
Policy:
  Policy in use: Merged from multiple sources.
  Reporting ACK required: yes (CISCO default)
  Unenforced/Non-Export Perpetual Attributes:
    First report requirement (days): 365 (CISCO default)

    Reporting frequency (days): 0 (CISCO default)
    Report on change (days): 90 (CISCO default)
  Unenforced/Non-Export Subscription Attributes:
    First report requirement (days): 90 (CISCO default)

    Reporting frequency (days): 90 (CISCO default)
    Report on change (days): 90 (CISCO default)
  Enforced (Perpetual/Subscription) License Attributes:

    First report requirement (days): 0 (CISCO default)
    Reporting frequency (days): 0 (CISCO default)
    Report on change (days): 0 (CISCO default)
  Export (Perpetual/Subscription) License Attributes:
    First report requirement (days): 0 (CISCO default)
    Reporting frequency (days): 0 (CISCO default)
    Report on change (days): 0 (CISCO default)
Miscellaneous:
  Custom Id: <empty>
Usage Reporting:
  Last ACK received: <none>
  Next ACK deadline: Jan 26 10:27:59 2021 PST
  Reporting push interval: 20  days
  Next ACK push check: <none>
  Next report push: Oct 28 10:29:59 2020 PST
  Last report push: <none>
  Last report file write: <none>
Trust Code Installed: <none>
```

The CSSM Web UI After Migration

No changes in the CSSM Web UI.

Reporting After Migration

Implement any one of the supported topologies, and fulfil reporting requirements. See Supported Topologies, on page 105 and How to Configure Smart Licensing Using Policy: Workflows by Topology , on page 123. The reporting method you can use depends on the topology you implement.

Migrating to a Version of SSM On-Prem That Supports Smart Licensing Using Policy

If you are using a version of SSM On-Prem that is earlier than the minimum required version (See SSM On-Prem, on page 99), you can use this section as an outline of the process and sequence you have to follow to migrate the SSM On-Prem version, the product instance, and any other tasks like SLAC installation, if applicable.

1. Upgrade SSM On-Prem.

   Upgrade to the minimum required Version 8, Release 202102 or a later version.

   Refer to the Cisco Smart Software Manager On-Prem Migration Guide.

2. Upgrade the product instance.

   For information about when Smart Licensing Using Policy was introduced on a supported product instance, see: Supported Products, on page 97.

   For information about the upgrade procedure, see Upgrading the Switch Software, on page 137.

3. Re-Register a local account with CSSM

   Online and Offline options are available. Refer to the Cisco Smart Software Manager On-Prem Migration Guide > *Re-Registering a local Account (Online Mode)* or *Manually Re-Registering a Local Account (Offline Mode)* .

   Once re-registration is complete, the following events occur automatically:

   - SSM On-Prem responds with new transport URL that points to the tenant in SSM On-Prem.
   - The transport type configuration on the product instance changes from from **call-home** or **smart**, to **cslu**. The transport URL is also updated automatically.

4. Save configuration changes on the product instance, by entering the **copy running-config startup-config** command in privileged EXEC mode.

5. Clear older On-Prem Smart Licensing certificates on the product instance and reload the product instance. Do not save configuration changes after this.

   Note This step is required only if the software version running on the product instance is Cisco IOS XE Amsterdam 17.3.x or Cisco IOS XE Bengaluru 17.4.x.

   Enter the **license smart factory reset** and then the **reload** commands in privileged EXEC mode.

   ```
   Device# license smart factory reset
   Device# reload
   ```

6. Perform usage synchronization

   a. On the product instance, enter the **license smart sync** {**all**|**local**} command, in privileged EXEC mode. This synchronizes the product instance with SSM On-Prem, to send and receive any pending data.

      ```
      Device(config)# license smart sync local
      ```

You can verify this in the SSM On-Prem UI. Go to **Inventory** > **SL Using Policy**. In the **Alerts** column, the following message is displayed: Usage report from product instance.

b. Synchronize usage information with CSSM (*choose one*)

- Option 1:

  SSM On-Prem is connected to CSSM: In the SSM On-Prem UI, Smart Licensing workspace, nagivate to **Reports** > **Usage Schedules** > **Synchronize now with Cisco**.

- Option 2:

  SSM On-Prem is not connected to CSSM. See: Exporting and Importing Usage Data (SSM On-Prem UI), on page 183.

Result:

You have completed migration and initial usage synchronization. Product instance and license usage information is now displayed in SSM On-Prem.

For subsequent reporting, you have the following options:

- To synchronize data between the product instance and SSM On-Prem:
  - Schedule periodic synchronization between the product instance and SSM On-Prem, by configuring the reporting interval. Enter the **license smart usage interval** *interval_in_days* command in global configuration mode.

    To know when the product instance will be sending the next RUM report, enter the **show license all** command in privileged EXEC mode and in the output, check the `Next report push:` field.
  - Enter the **license smart sync** privileged EXEC command, for ad hoc or on-demand synchronization between the product instance and SSM On-Prem.
- To synchronize usage information with CSSM:
  - Schedule periodic synchronization with CSSM. In the SSM On-Prem UI, navigate to **Reports** > **Usage Schedules** > **Synchronization schedule with Cisco**. Enter the following frequency information and save:
    - **Days**: Refers to how *often* synchronization occurs. For example, if you enter 2, synchronization occurs once every two days.
    - **Time of Day:**: Refers to the time at which synchronization occurs, in the 24-hour notation system. For example, if you enter 14 hours and 0 minutes, synchronization occurs at 2 p.m. (1400) in your local time zone.
  - Upload and download the required files for reporting: Exporting and Importing Usage Data (SSM On-Prem UI), on page 183).

Task Library for Smart Licensing Using Policy

This section is a grouping of tasks that apply to Smart Licensing Using Policy. It includes tasks performed on a product instance, on the CSLU interface, and on the CSSM Web UI.

To implement a particular topology, refer to the corresponding workflow to know the sequential order of tasks that apply. See How to Configure Smart Licensing Using Policy: Workflows by Topology , on page 123

To perform any additional configuration tasks, for instance, to configure a different license, or use an add-on license, or to configure a narrower reporting interval, refer to the corresponding task here. Check the "Supported Topologies" where provided, before you proceed.

Logging into Cisco (CSLU Interface)

Depending on your needs, when working in CSLU, you can either be in connected or disconnected mode. To work in the connected mode, complete these steps to connect with Cisco.

Procedure

Step 1 From the CSLU Main screen, click **Login to Cisco** (located at the top right corner of the screen).

Step 2 Enter: **CCO User Name** and **CCO Password**.

Step 3 In the CSLU Preferences tab, check that the Cisco connectivity toggle displays “Cisco Is Available”.

Configuring a Smart Account and a Virtual Account (CSLU Interface)

Both the Smart Account and Virtual Account are configured through the Preferences tab. Complete the following steps to configure both Smart and Virtual Accounts for connecting to Cisco.

Procedure

Step 1 Select the **Preferences Tab** from the CSLU home screen.

Step 2 Perform these steps for adding both a Smart Account and Virtual Account:

a) In the Preferences screen navigate to the **Smart Account** field and add the **Smart Account Name**.

b) Next, navigate to the **Virtual Account** field and add the **Virtual Account Name**.

If you are connected to CSSM (In the Preferences tab, **Cisco is Available**), you can select from the list of available SA/VAs.

If you are not connected to CSSM (In the Preferences tab, **Cisco Is Not Available**), enter the SA/VAs manually.

Note SA/VA names are case sensitive.

Step 3 Click **Save**. The SA/VA accounts are saved to the system

Only one SA/VA pair can reside on CSLU at a time. You cannot add multiple accounts. To change to another SA/VA pair, repeat Steps 2a and 2b then Save. A new SA/VA account pair replaces the previous saved pair

Adding a Product-Initiated Product Instance in CSLU (CSLU Interface)

Complete these steps to add a device-created Product Instance using the Preferences tab.

Procedure

Step 1 Click the **Preferences** tab.

Step 2 In the Preferences screen, de-select the **Validate Device** check box.

Step 3 Set the **Default Connect Method** to **Product Instance Initiated** and then click **Save**.

Ensuring Network Reachability for Product Instance-Initiated Communication

This task provides *possible* configurations that may be required to ensure network reachability for product instance-initiated communication. Steps marked as "(Required)" are required for all product instances, all other steps my be required or optional, depending the kind of product instance and network requirements. Configure the applicable commands:

Before you begin

Supported topologies: Connected to CSSM Through CSLU (product instance-initiated communication).

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-type-number* **Example:** `Device (config)# interface gigabitethernet0/0`	Enters interface configuration mode and specifies the Ethernet interface, subinterface, or VLAN to be associated with the VRF.
Step 4	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)# vrf forwarding Mgmt-vrf`	Associates the VRF with the Layer 3 interface. This command activates multiprotocol VRF on an interface
Step 5	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 192.168.0.1 255.255.0.0`	Defines the IP address for the VRF.

	Command or Action	Purpose
Step 6	**negotiation auto** **Example:** `Device(config-if)# negotiation auto`	Enables auto-negotiation operation for the speed and duplex parameters of an interface.
Step 7	**end** **Example:** `Device(config-if)# end`	Exits the interface configuration mode and enters global configuration mode.
Step 8	**ip http client source-interface** *interface-type-number* **Example:** `Device(config)# ip http client source-interface gigabitethernet0/0`	Configures a source interface for the HTTP client.
Step 9	**ip route** *ip-address ip-mask subnet mask* **Example:** `Device(config)# ip route vrf mgmt-vrf 192.168.0.1 255.255.0.0 192.168.255.1`	(Required) Configures a route and gateway on the product instance. You can configure either a static route or a dynamic route.
Step 10	{**ip** | **ipv6**} **name-server** *server-address 1 ...server-address 6*] **Example:** `Device(config)# Device(config)# ip name-server vrf mgmt-vrf 173.37.137.85`	Configures Domain Name System (DNS) on the VRF interface.
Step 11	**ip domain lookup source-interface** *interface-type-number* **Example:** `Device(config)# ip domain lookup source-interface gigabitethernet0/0`	Configures the source interface for the DNS domain lookup. **Note** If you configure this command on a Layer 3 physical interface, it is automatically removed from running configuration in case the port mode is changed or if the device reloads. The only available workaround is to reconfigure the command. Starting with Cisco IOS XE Dublin 17.12.1, this issue is resolved.
Step 12	**ip domain name** *domain-name* **Example:** `Device(config)# ip domain name example.com`	Configure DNS discovery of your domain. In accompanying example, the name-server creates entry `cslu-local.example.com`.

Adding a CSLU-Initiated Product Instance in CSLU (CSLU Interface)

Using the CSLU interface, you can configure the connect method to be CSLU Initiated. This connect method (mode) enables CSLU to retrieve product instance information.

Note The default Connect Method is set in the **Preferences** tab.

Complete these steps to add a Product Instance from the Inventory tab

Procedure

Step 1 Go to the **Inventory** tab and from the Product Instances table, select **Add Single Product**.

Step 2 Enter the **Host** (IP address of the host).

Step 3 Select the **Connect Method** and select an appropriate CSLU Initiated connect method.

Step 4 In the right panel, click **Product Instance Login Credentials**. The left panel of the screen changes to show the User Name and Password fields

Step 5 Enter the product instance **User Name** and **Password**.

Step 6 Click **Save**.

The information is saved to the system and the device is listed in the Product Instances table with the Last Contact listed as never.

Collecting Usage Reports: CSLU Initiated (CSLU Interface)

CSLU also allows you to manually trigger the gathering of usage reports from devices.

After configuring and selecting a product instance (selecting **Add Single Product Instance**, filling in the host name and selecting a CSLU Initiated connect method), select **Actions for Selected** > **Collect Usage**. CSLU connects to the selected product instances and collects usage reports. These usage reports are stored in CSLU's local library. These reports can then be transferred to Cisco if CSLU is connected to Cisco, or (if you are not connected to Cisco) you can manually trigger usage collection by selecting **Data** > **Export to CSSM**.

If you are working in CSLU-initiated mode, complete these steps to configure CSLU to collect RUM reports from Product Instances.

Procedure

Step 1 Click the **Preferences** tab and enter a valid Smart Account and Virtual Account, and then select an appropriate CSLU Initiated collect method. (If there have been any changes in Preferences, make sure you click **Save**.)

Step 2 Click the **Inventory** tab and select one or more product instances.

Step 3 Click **Actions for Selected** > **Collect Usage**

RUM reports are retrieved from each selected device and stored in the CSLU local library. The Last Contact column is updated to show the time the report was received, and the Alerts column shows the status.

If CSLU is currently logged into Cisco the reports will be automatically sent to the associated Smart Account and Virtual Account in Cisco and Cisco will send an acknowledgement to CSLU as well as to the product instance. The acknowledgement will be listed in the alerts column of the Product Instance table.

To manually transfer usage reports Cisco, from the CSLU main screen select **Data** > **Export to CSSM**.

Step 4 From the **Export to CSSM** modal, you can select the local directory where the reports are to be stored. (<CSLU_WORKING_Directory>/data/default/rum/unsent)

At this point, the usage reports are saved in your local directory (library). To upload these usage reports to Cisco, follow the steps described in Uploading Data or Requests to CSSM and Downloading a File, on page 210.

Note The Windows operating system can change the behavior of a usage report file properties by dropping the extension when that file is renamed. The behavior change happens when you rename the downloaded file and the renamed file drops the extension. For example, the downloaded default file named `UD_xxx.tar` is renamed to `UD_yyy`. The file loses its TAR extension and cannot function. To enable the usage file to function normally, after re-naming a usage report file, you must also add the TAR extension back to the file name, for example `UD_yyy.tar`.

Export to CSSM (CSLU Interface)

This option can be used as a part of a manual download procedure when you want the workstation isolated for security purposes.

Procedure

Step 1 Go to the **Preferences** tab, and turn off the **Cisco Connectivity** toggle switch.

The field switches to “Cisco Is Not Available”.

Step 2 From the CSLU home screen, navigate to **Data** > **Export to CSSM**.

Step 3 Select the file from the modal that opens and click **Save**. You now have the file saved.

Note At this point you have a DLC file, RUM file, or both.

Step 4 From a workstation that has connectivity to Cisco, and complete the following: Uploading Data or Requests to CSSM and Downloading a File, on page 210

Once the file is downloaded, you can import it into CSLU. See: Import from CSSM (CSLU Interface), on page 166

Import from CSSM (CSLU Interface)

Once you have received the ACK or other file (such as an authorization code) from Cisco, you are ready to upload that file to your system. This procedure can be used for workstations that are offline. Complete these steps to select and upload files from Cisco.

Procedure

Step 1 Ensure that you have downloaded the file to a location that is accessible to CSLU.

Step 2 From the CSLU home screen, navigate to **Data** > **Import from CSSM**.

Step 3 An Import from CSSM modal open for you to either:

- Drag and Drop a **File** that resides on your local drive, or
- Browse for the appropriate *.xml file, select the file and click **Open**.

If the upload is successful, you will get a message indicating that the file was successfully sent to the server. If the upload is not successful, you will get an import error.

Step 4 When you have finished uploading, click the **x** at the top right corner of the modal to close it.

Ensuring Network Reachability for CSLU-Initiated Communication

This task provides *possible* configurations that may be required to ensure network reachability for CSLU-initiated communication. Steps marked as "(Required)" are required for all product instances, all other steps may be required or optional, depending the kind of product instance and network requirements. Configure the applicable commands:

Before you begin

Supported topologies: Connected to CSSM Through CSLU (CSLU-initiated communication).

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa new model** **Example:** `Device(config)# aaa new model`	(Required) Enable the authentication, authorization, and accounting (AAA) access control model.
Step 4	**aaa authentication login default local** **Example:** `Device(config)# aaa authentication login default local`	(Required) Sets AAA authentication to use the local username database for authentication.

	Command or Action	Purpose
Step 5	**aaa authorization exec default local** **Example:** `Device(config)# aaa authorization exec default local`	Sets the parameters that restrict user access to a network. The user is allowed to run an EXEC shell.
Step 6	ip routing **Example:** `Device(config)# ip routing`	Enables IP routing.
Step 7	{**ip** | **ipv6**} **name-server** *server-address 1 ...server-address 6*] **Example:** `Device(config)# ip name-server vrf Mgmt-vrf 192.168.1.100 192.168.1.200 192.168.1.300`	(Optional) Specifies the address of one or more name servers to use for name and address resolution. You can specify up to six name servers. Separate each server address with a space. The first server specified is the primary server. The device sends DNS queries to the primary server first. If that query fails, the backup servers are queried.
Step 8	**ip domain lookup source-interface** *interface-type-number* **Example:** `Device(config)# ip domain lookup source-interface gigabitethernet0/0`	Enables DNS-based hostname-to-address translation on your device. This feature is enabled by default. If your network devices require connectivity with devices in networks for which you do not control name assignment, you can dynamically assign device names that uniquely identify your devices by using the global Internet naming scheme (DNS). **Note** If you configure this command on a Layer 3 physical interface, it is automatically removed from running configuration in case the port mode is changed or if the device reloads. The only available workaround is to reconfigure the command. Starting with Cisco IOS XE Dublin 17.12.1, this issue is resolved.
Step 9	**ip domain name** *name* **Example:** `Device(config)# ip domain name vrf Mgmt-vrf cisco.com`	Defines a default domain name that the software uses to complete unqualified hostnames (names without a dotted-decimal domain name).

	Command or Action	Purpose
Step 10	**no username** *name* **Example:** `Device(config)# no username admin`	(Required) Clears the specified username, if it exists. For *name* , enter the same username you will create in the next step. This ensures that a duplicate of the username you are going to create in the next step does not exist. If you plan to use REST APIs for CSLU-initiated retrieval of RUM reports, you have to log in to CSLU. Duplicate usernames may cause the feature to work incorrectly if there are duplicate usernames in the system.

Step 11	**username** *name* **privilege** *level* **password** *password*
	Example:
	`Device(config)# username admin privilege 15`
	`password 0 lab`
		(Required) Establishes a username-based authentication system.
		The **privilege** keyword sets the privilege level for the user. A number between 0 and 15 that specifies the privilege level for the user.
		The password allows access to the name argument. A password must be from 1 to 25 characters, can contain embedded spaces, and must be the last option specified in the **username** command.
		This enables CSLU to use the product instance native REST.
		Note Enter this username and password in CSLU (Collecting Usage Reports: CSLU Initiated (CSLU Interface), on page 165 → *Step 4. f.* CSLU can then collect RUM reports from the product instance.

Step 12	**interface** *interface-type-number* **Example:** `Device (config)# interface gigabitethernet0/0`	Enters interface configuration mode and specifies the Ethernet interface, subinterface, or VLAN to be associated with the VRF.
Step 13	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)# vrf forwarding Mgmt-vrf`	Associates the VRF with the Layer 3 interface. This command activates multiprotocol VRF on an interface
Step 14	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 192.168.0.1 255.255.0.0`	Defines the IP address for the VRF.

	Command or Action	Purpose
Step 15	**negotiation auto** **Example:** `Device(config-if)# negotiation auto`	Enables auto-negotiation operation for the speed and duplex parameters of an interface.
Step 16	**no shutdown** **Example:** `Device(config-if)# no shutdown`	Restarts a disabled interface.
Step 17	**end** **Example:** `Device(config-if)# end`	Exits the interface configuration mode and enters global configuration mode.
Step 18	**ip http server** **Example:** `Device(config)# ip http server`	(Required) Enables the HTTP server on your IP or IPv6 system, including a Cisco web browser user interface. The HTTP server uses the standard port 80, by default.

Step 19	**ip http authentication local**
	Example:
	`ip http authentication local`
	`Device(config)#`
		(Required) Specifies a particular authentication method for HTTP server users.
		The **local** keyword means that the login user name, password and privilege level access combination specified in the local system configuration (by the username global configuration command) should be used for authentication and authorization.

Step 20	**ip http secure-server** **Example:** `Device(config)# ip http server`	(Required) Enables a secure HTTP (HTTPS) server. The HTTPS server uses the Secure Sockets Layer (SSL) version 3.0 protocol.
Step 21	**ip http max-connections** **Example:** `Device(config)# ip http max-connections 16`	(Required) Configures the maximum number of concurrent connections allowed for the HTTP server. Enter an integer in the range from 1 to 16. The default is 5.
Step 22	**ip tftp source-interface** *interface-type-number* **Example:** `Device(config)# ip tftp source-interface GigabitEthernet0/0`	Specifies the IP address of an interface as the source address for TFTP connections.
Step 23	**ip route** *ip-address ip-mask subnet mask* **Example:** `Device(config)# ip route vrf mgmt-vrf 192.168.0.1 255.255.0.0 192.168.255.1`	Configures a route and gateway on the product instance. You can configure either a static route or a dynamic route.

	Command or Action	Purpose
Step 24	**logging host** **Example:** `Device(config)# logging host 172.25.33.20 vrf Mgmt-vrf`	Logs system messages and debug output to a remote host.
Step 25	**end** **Example:** `Device(config)# end`	Exits the global configuration mode and enters priveleged EXEC mode.
Step 26	**show ip http server session-module** **Example:** `Device# show ip http server session-module`	(Required) Verifies HTTP connectivity. In the output, check that `SL_HTTP` is active. Additionally, you can also perform the following checks : • From device where CSLU is installed, verify that you can ping the product instance. A successful ping confirms that the product instance is reachable. • From a Web browser on the device where CSLU is installed verify `https://<product-instance-ip>/`. This ensures that the REST API from CSLU to the product instance works as expected.

Requesting SLAC for One or More Product Instance (CSLU Interface)

This task shows you how to manually request SLAC for one or more product instances in CSLU.

Before you begin

Supported topologies:

- Connected to CSSM Through CSLU (Product instance-initiated and CSLU-initiated)
- CSLU Disconnected from CSSM (Product instance-initiated and CSLU-initiated)

Procedure

Step 1 Navigate to the **Inventory** tab. From the Product Instance table, select the one or more product instances for authorization code request.

Step 2 From the **Actions for Selected** menu, select the **Authorization Code Request** option.

The **Authorization Request Information** modal pops up.

Step 3 Click **Accept**.

Another modal opens to select a local .csv file for uploading.

Step 4 Upload the file to CSSM, generate authorization codes and download the file containing the codes. See Generating and Downloading SLAC from CSSM to a File, on page 199.

Step 5 Return to the CSLU interface.

Step 6 Apply the authorization codes by selecting **Data** > **Import from CSSM**. See Import from CSSM (CSLU Interface), on page 166

If CSLU is in the product instance-initiated mode: The uploaded codes are applied to the product instance the next time the product instance contacts CSLU.

If CSLU is in the CSLU-initiated mode: The uploaded codes are now applied to the product instance the next time the CSLU runs an update.

Setting Up a Connection to CSSM

The following steps show how to set up a Layer 3 connection to CSSM to verify network reachability. Steps marked as "(Required)" are required for all product instances, all other steps may be required or optional, depending the kind of product instance and network requirements. Configure the applicable commands:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	{**ip** | **ipv6**} **name-server** *server-address 1 ...server-address 6*] **Example:** `Device(config)# ip name-server 209.165.201.1 209.165.200.225 209.165.201.14 209.165.200.230`	Specifies the address of one or more name servers to use for name and address resolution. You can specify up to six name servers. Separate each server address with a space. The first server specified is the primary server. The device sends DNS queries to the primary server first. If that query fails, the backup servers are queried.
Step 4	**ip name-server vrf Mgmt-vrf** *server-address 1...server-address 6* **Example:** `Device(config)# ip name-server vrf Mgmt-vrf 209.165.201.1 209.165.200.225 209.165.201.14 209.165.200.230`	(Optional) Configures DNS on the VRF interface. You can specify up to six name servers. Separate each server address with a space. **Note** This command is an alternative to the **ip name-server** command.

	Command or Action	Purpose
Step 5	**ip domain lookup source-interface** *interface-type interface-number* **Example:** `Device(config)# ip domain lookup source-interface Vlan100`	Configures the source interface for the DNS domain lookup.
Step 6	**ip domain name** *domain-name* **Example:** `Device(config)# ip domain name example.com`	Configures the domain name.
Step 7	**ip host tools.cisco.com** *ip-address* **Example:** `Device(config)# ip host tools.cisco.com 209.165.201.30`	Configures static hostname-to-address mappings in the DNS hostname cache if automatic DNS mapping is not available.

Step 8	**interface** *interface-type-number*
	Example:
	`Device(config)# interface Vlan100`
	`Device(config-if)# ip address 192.0.2.10 255.255.255.0`
	`Device(config-if)# exit`
		Configures a Layer 3 interface. Enter an interface type and number or a VLAN.

Step 9	**ntp server** *ip-address* [**version** *number*] [**key** *key-id*] [**prefer**] **Example:** `Device(config)# ntp server 198.51.100.100 version 2 prefer`	(Required) Activates the NTP service (if it has not already been activated) and enables the system to synchronize the system software clock with the specified NTP server. This ensures that the device time is synchronized with CSSM. Use the **prefer** keyword if you need to use this command multiple times and you want to set a preferred server. Using this keyword reduces switching between servers.

Step 10	**switchport access vlan** *vlan_id*
	Example:
	`Device(config)# interface GigabitEthernet1/0/1`
	`Device(config-if)# switchport access vlan 100`
	`Device(config-if)# switchport mode access`
	`Device(config-if)# exit`
	`OR`
	`Device(config)#`
		Enables the VLAN for which this access port carries traffic and sets the interface as a nontrunking nontagged single-VLAN Ethernet interface.
		Note This step is to be configured only if the switchport access mode is required. The **switchport access vlan** command may apply to Catalyst switching product instances, for example, and for routing product instances you may want to configure the **ip address** *ip-address mask* command instead.

	Command or Action	Purpose
Step 11	**ip route** *ip-address ip-mask subnet mask* **Example:** `Device(config)# ip route 192.0.2.0 255.255.255.255 192.0.2.1`	Configures a route on the device. You can configure either a static route or a dynamic route.
Step 12	**ip http client source-interface** *interface-type-number* **Example:** `Device(config)# ip http client source-interface Vlan100`	(Required) Configures a source interface for the HTTP client. Enter an interface type and number or a VLAN.
Step 13	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 14	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	Saves your entries in the configuration file.

Configuring Smart Transport Through an HTTPs Proxy

To use a proxy server to communicate with CSSM when using the Smart transport mode, complete the following steps:

Note *Authenticated* HTTPs proxy configurations are not supported.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**license smart transport smart** **Example:** `Device(config)# license smart transport smart`	Enables Smart transport mode.

	Command or Action	Purpose
Step 4	**license smart url default** **Example:** `Device(config)# license smart transport default`	Automatically configures the Smart URL (https://smartreceiver.cisco.com/licservice/license). For this option to work as expected, the transport mode in the previous step must be configured as **smart**.

Step 5	**license smart proxy** {**address** *address_hostname* | **port** *port_num*}
	Example:
	`Device(config)# license smart proxy address 192.168.0.1`
	`Device(config)# license smart proxy port 3128`
		Configures a proxy for the Smart transport mode. When a proxy is configured, licensing messages are sent to the proxy along with the final destination URL (CSSM). The proxy sends the message on to CSSM. Configure the proxy IP address and port information separately:
		• **address** *address_hostname*: Specifies the proxy address. Enter the IP address or hostname of the proxy server.
		• **port** *port_num*: Specifies the proxy port. Enter the proxy port number.

Step 6	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	Saves your entries in the configuration file.

Configuring the Call Home Service for Direct Cloud Access

The Call Home service provides email-based and web-based notification of critical system events to CSSM. To configure the transport mode, enable the Call Home service, and configure a destination profile (A destination profile contains the required delivery information for an alert notification. At least one destination profile is required.), complete the following steps:

Note All steps are required unless specifically called-out as "(Optional)".

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**license smart transport callhome** **Example:** `Device(config)# license smart transport callhome`	Enables Call Home as the transport mode.
Step 4	**license smart url** *url* **Example:** `Device(config)# license smart url https://tools.cisco.com/its/service/oddce/services/DDCEService`	For the **callhome** transport mode, configure the CSSM URL exactly as shown in the example.
Step 5	**service call-home** **Example:** `Device(config)# service call-home`	Enables the Call Home feature.
Step 6	**call-home** **Example:** `Device(config)# call-home`	Enters Call Home configuration mode.
Step 7	**contact-email-address** *email-address* **Example:** `Device(config-call-home)# contact-email-addr username@example.com`	Assigns customer's email address and enables Smart Call Home service full reporting capability and sends a full inventory message from Call-Home TAC profile to Smart Call Home server to start full registration process. You can enter up to 200 characters in email address format with no spaces.

Step 8	**profile** *name*
	Example:
	`Device(config-call-home)# profile CiscoTAC-1`
	`Device(config-call-home-profile)#`
		Enters the Call Home destination profile configuration submode for the specified destination profile.
		By default:
		• The CiscoTAC-1 profile is inactive. To use this profile with the Call Home service, you must enable the profile.
		• The CiscoTAC-1 profile sends a full report of all types of events subscribed in the profile. The alternative is to additionally configure `Device(cfg-call-home-profile)# anonymous-reporting-only` anonymous-reporting-only. When this is set, only crash, inventory, and test messages will be sent.

	Command or Action	Purpose
		Use the **show call-home profile all** command to check the profile status.
Step 9	**active** **Example:** `Device(config-call-home-profile)# active`	Enables the destination profile.

Step 10	**destination transport-method http** {**email** |**http**}
	Example:
	`Device(config-call-home-profile)# destination transport-method http`
	AND
	`Device(config-call-home-profile)# no destination transport-method email`
		Enables the message transport method. In the example, Call Home service is enabled via HTTP and transport via email is disabled.
		The **no** form of the command disables the method.

Step 11	**destination address** { **email** *email_address* |**http** *url*}
	Example:
	`Device(config-call-home-profile)# destination address http https://tools.cisco.com/its/service/oddce/services/DDCEService`
	AND
	`Device(config-call-home-profile)# no destination address http https://tools.cisco.com/its/service/oddce/services/DDCEService`
		Configures the destination e-mail address or URL to which Call Home messages are sent. When entering a destination URL, include either **http://** (default) or **https://**, depending on whether the server is a secure server.
		In the example provided here, a **http://** destination URL is configured; and the **no** form of the command is configured for **https://**.

Step 12	**exit** **Example:** `Device(config-call-home-profile)# exit`	Exits Call Home destination profile configuration mode and returns to Call Home configuration mode.
Step 13	**exit** **Example:** `Device(config-call-home)# end`	Exits Call Home configuration mode and returns to privileged EXEC mode.
Step 14	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	Saves your entries in the configuration file.
Step 15	**show call-home profile** {*name* |**all**}	Displays the destination profile configuration for the specified profile or all configured profiles.

Configuring the Call Home Service for Direct Cloud Access through an HTTPs Proxy Server

The Call Home service can be configured through an HTTPs proxy server. This configuration requires no user authentication to connect to CSSM.

Note Authenticated HTTPs proxy configurations are not supported.

To configure and enable the Call Home service through an HTTPs proxy, complete the following steps:

Note All steps are required unless specifically called-out as "(Optional)".

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**license smart transport callhome** **Example:** `Device(config)# license smart transport callhome`	Enables Call Home as the transport mode.
Step 4	**service call-home** **Example:** `Device(config)# service call-home`	Enables the Call Home feature.
Step 5	**call-home** **Example:** `Device(config)# call-home`	Enters Call Home configuration mode.
Step 6	**http-proxy** *proxy-address* **proxy-port** *port-number* **Example:** `Device(config-call-home)# http-proxy 198.51.100.10 port 5000`	Configures the proxy server information to the Call Home service.

	Command or Action	Purpose
Step 7	**exit** **Example:** `Device(config-call-home)# exit`	Exits Call Home configuration mode and enters global configuration mode. Note the change in the criteria for the acceptance of proxy servers, starting with Cisco IOS XE Bengaluru 17.6.1: only the status code of the proxy server response is verified by the system and not the reason phrase. The RFC format is `status-line = HTTP-version SP status-code SP reason-phrase CRLF`. For more information about the status line, see section 3.1.2 of RFC 7230.
Step 8	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and enters privileged EXEC mode.
Step 9	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	Saves your entries in the configuration file.

Assigning a Smart Account and Virtual Account (SSM On-Prem UI)

You can use this procedure to import one or more product instances along with corresponding Smart Account and Virtual Account information, into the SSM On-Prem database. This enables SSM On-Prem to map product instances that are part of local virtual accounts (other than the default local virtual account), to the correct license pool in CSSM:

Before you begin

Supported topologies: SSM On-Prem Deployment (product instance-initiated communication).

Procedure

Step 1 Log into the SSM On-Prem and select the **Smart Licensing** workspace.

Step 2 Navigate to **Inventory** > **SL Using Policy** > **Export/Import All** > **Import Product Instances List**

The **Upload Product Instances** window is displayed.

Step 3 Click **Download** to download the .csv template file and enter the required information for all the product instances in the template.

Step 4 Once you have filled-out the template, click **Inventory** > **SL Using Policy** > **Export/Import All** > **Import Product Instances List**.

The **Upload Product Instances** window is displayed.

Step 5 Now, click **Browse** and upload the filled-out .csv template.

Smart Account and Virtual Account information for all uploaded product instances is now available in SSM On-Prem.

Validating Devices (SSM On-Prem UI)

When device validation is enabled, RUM reports from an unknown product instance (not in the SSM On-Prem database) are rejected.

By default, devices are not validated. Complete the following steps to enable the function:

Before you begin

Supported topologies: SSM On-Prem Deployment (product instance-initiated communication).

Procedure

Step 1 In the **On-Prem License Workspace** window, click **Admin Workspace** and log in, if prompted.

The **On-Prem Admin Workspace** window is displayed.

Step 2 Click the **Settings** widget.

The **Settings** window is displayed.

Step 3 Navigate to the **CSLU** tab and turn-on the **Validate Device** toggle switch.

RUM reports from an unknown product instance will now be rejected. If you haven't already, you must now add the required product instances to the SSM On-Prem database before sending RUM reports. See Assigning a Smart Account and Virtual Account (SSM On-Prem UI), on page 179.

Ensuring Network Reachability for Product Instance-Initiated Communication

This task provides *possible* configurations that may be required to ensure network reachability for product instance-initiated communication. Steps marked as "(Required)" are required for all product instances, all other steps my be required or optional, depending the kind of product instance and network requirements. Configure the applicable commands:

Note Ensure that you configure steps 13, 14, and 15 exactly as shown below. These commands must be configured to ensure that the correct trustpoint is used and that the necessary certificates are accepted for network reachability.

Before you begin

Supported topologies: SSM On-Prem Deployment (product instance-initiated communication).

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *interface-type-number* **Example:** `Device (config)# interface gigabitethernet0/0`	Enters interface configuration mode and specifies the Ethernet interface, subinterface, or VLAN to be associated with the VRF.
Step 4	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)# vrf forwarding Mgmt-vrf`	Associates the VRF with the Layer 3 interface. This command activates multiprotocol VRF on an interface
Step 5	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 192.168.0.1 255.255.0.0`	Defines the IP address for the VRF.
Step 6	**negotiation auto** **Example:** `Device(config-if)# negotiation auto`	Enables auto-negotiation operation for the speed and duplex parameters of an interface.
Step 7	**end** **Example:** `Device(config-if)# end`	Exits the interface configuration mode and enters global configuration mode.
Step 8	**ip http client source-interface** *interface-type-number* **Example:** `Device(config)# ip http client source-interface gigabitethernet0/0`	Configures a source interface for the HTTP client.
Step 9	**ip route** *ip-address ip-mask subnet mask* **Example:** `Device(config)# ip route vrf mgmt-vrf 192.168.0.1 255.255.0.0 192.168.255.1`	(Required) Configures a route and gateway on the product instance. You can configure either a static route or a dynamic route.

	Command or Action	Purpose
Step 10	{**ip** | **ipv6**} **name-server** *server-address 1 ...server-address 6*] **Example:** `Device(config)# Device(config)# ip name-server vrf mgmt-vrf 198.51.100.1`	Configures Domain Name System (DNS) on the VRF interface.
Step 11	**ip domain lookup source-interface** *interface-type-number* **Example:** `Device(config)# ip domain lookup source-interface gigabitethernet0/0`	Configures the source interface for the DNS domain lookup. **Note** If you configure this command on a Layer 3 physical interface, it is automatically removed from running configuration in case the port mode is changed or if the device reloads. The only available workaround is to reconfigure the command. Starting with Cisco IOS XE Dublin 17.12.1, this issue is resolved.
Step 12	**ip domain name** *domain-name* **Example:** `Device(config)# ip domain name example.com`	Configure DNS discovery of your domain. In the accompanying example, the name-server creates entry `cslu-local.example.com`.

Step 13	**crypto pki trustpoint SLA-TrustPoint**
	Example:
	`Device(config)# crypto pki trustpoint SLA-TrustPoint`
	`Device(ca-trustpoint)#`
		(Required) Declares that the product instance should use trustpoint "SLA-TrustPoint" and enters the ca-trustpoint configuration mode. The product instance does not recognize any trustpoints until you declare a trustpoint using this command.

Step 14	enrollment terminal **Example:** `Device(ca-trustpoint)# enrollment terminal`	(Required) Specifies the certificate enrollment method.
Step 15	**revocation-check none** **Example:** `Device(ca-trustpoint)# revocation-check none`	(Required) Specifes a method that is to be used to ensure that the certificate of a peer is not revoked. For the SSM On-Prem Deployment topology, enter the **none** keyword. This means that a revocation check will not be performed and the certificate will always be accepted.

	Command or Action	Purpose

Step 16	**exit**
	Example:
	`Device(ca-trustpoint)# exit`
	`Device(config)# exit`
		Exits the ca-trustpoint configuration mode and then the global configuration mode and returns to privileged EXEC mode.

Step 17	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	Saves your entries in the configuration file.

Retrieving the Transport URL (SSM On-Prem UI)

You must configure the transport URL on the product instance when you deploy product instance-initiated communication in an SSM On-Prem deployment. This task shows you how to easily copy the complete URL including the tenant ID from SSM On-Prem.

Before you begin

Supported topologies: SSM On-Prem Deployment (product instance-initiated communication).

Procedure

Step 1 Log into SSM On-Prem and select the **Smart Licensing** workspace.

Step 2 Navigate to the **Inventory** tab and from the dropdown list of local virtual accounts (top right corner), select the *default local virtual account*. When you do, the area under the **Inventory** tab displays **Local Virtual Account: Default**.

Step 3 Navigate to the **General** tab.

The **Product Instance Registration Tokens** area is displayed.

Step 4 In the **Product Instance Registration Tokens** area click **CSLU Transport URL**.

The **Product Registration URL** pop-window is displayed.

Step 5 Copy the entire URL and save it in an accessible place.

You will require the URL when you configure the transport type and URL on the product instance.

Step 6 Configure the transport type and URL. See: Setting the Transport Type, URL, and Reporting Interval, on page 212.

Exporting and Importing Usage Data (SSM On-Prem UI)

You can use this procedure to complete usage synchronization between SSM On-Prem and CSSM when SSM On-Prem is disconnected from CSSM.

Before you begin

Supported topologies:

- SSM On-Prem Deployment (SSM On-Prem-initiated communication)
- SSM On-Prem Deployment (product instance-initiated communication).

Reporting data must be available in SSM On-Prem. You must have either pushed the nessary reporting data from the product instance to SSM On-Prem (product instance-initiated communication) or retrieved the necessary reporting data from the product instance (SSM On-Prem-initiated communication).

Procedure

Step 1 Log into SSM On-Prem and select **Smart Licensing**.

Step 2 Navigate to **Inventory** > **SL Using Policy** tab.

Step 3 In the **SL Using Policy** tab area, click **Export/Import All...** > **Export Usage to Cisco**.

This generates one .tar file with *all* the usage reports available in the SSM On-Prem server.

Step 4 Complete this task in CSSM: Uploading Data or Requests to CSSM and Downloading a File, on page 210.

At the end of this task you will have an ACK file to import into SSM On-Prem.

Step 5 Again navigate to the **Inventory** > **SL Using Policy** tab.

Step 6 In the **SL Using Policy** tab area, click **Export/Import All...** > **Import From Cisco** . Upload the .tar ACK file.

To verify ACK import, in the **SL Using Policy** tab area check the **Alerts** column of the corresponding product instance. The following message is displayed: Acknowledgement received from CSSM.

Adding One or More Product Instances (SSM On-Prem UI)

You can use this procedure to add one product instance or to import and add multiple product instances. It enables SSM On-Prem to retrieve information from the product instance.

Before you begin

Supported topologies: SSM On-Prem Deployment (SSM On-Prem-initiated communication).

Procedure

Step 1 Log into the SSM On-Prem UI and click **Smart Licensing**.

Step 2 Navigate to **Inventory** tab. Select a local virtual account from the drop-down list in the top right corner.

Step 3 Navigate to the **SL Using Policy** tab.

Step 4 Add a single product or import multiple product instances (*choose one*).

- **To add a single product instance**:

a. In the **SL Using Policy** tab area, click **Add Single Product**.

b. In the **Host** field, enter the IP address of the host (product instance).

c. From the **Connect Method** dropdown list, select an appropriate SSM On-Prem-initiated connect method.

   The available connect methods for SSM On-Prem-initiated communication are: NETCONF, RESTCONF, and REST API.

d. In the right panel, click **Product Instance Login Credentials**.

   The **Product Instance Login Credentials** window is displayed

   Note You need the login credentials only if a product instance requires a SLAC.

e. Enter the **User ID** and **Password**, and click **Save**.

   This is the same user ID and password that you configured as part of commands required to establish network reachability (Ensuring Network Reachability for SSM On-Prem-Initiated Communication, on page 186).

   Once validated, the product instance is displayed in the listing in the **SL Using Policy** tab area.

- **To import multiple product instances:**

a. In **SL Using Policy** tab, click **Export/Import All...** > **Import Product Instances List**.

   The **Upload Product Instances** window is displayed.

b. Click **Download** to download the predefined .csv template.

c. Enter the required information for all the product instances in the .csv template.

   In the template, ensure that you provide **Host**, **Connect Method** and **Login Credentials** for all product instances.

   The available connect methods for SSM On-Prem-initiated communication are: NETCONF, RESTCONF, and REST API.

   Login credentials refer to the user ID and password that you configured as part of commands required to establish network reachability (Ensuring Network Reachability for SSM On-Prem-Initiated Communication, on page 186).

d. Again navigate to **Inventory** > **SL Using Policy** tab. Click **Export/Import All....** > **Import Product Instances List**.

   The **Upload Product Instances** window is displayed.

e. Now upload the filled-out .csv template.

   Once validated, the product instances are displayed in the listing in the **SL Using Policy** tab.

Ensuring Network Reachability for SSM On-Prem-Initiated Communication

This task provides *possible* configurations that may be required to ensure network reachability for SSM On-Prem-initiated communication. Steps marked as "(Required)" are required for all product instances, all other steps may be required or optional, depending the kind of product instance and network requirements. Configure the applicable commands:

Note Ensure that you configure steps 25, 26, and 27 exactly as shown below. These commands must be configured to ensure that the correct trustpoint is used and that the necessary certificates are accepted for network reachability.

Before you begin

Supported topologies: SSM On-Prem Deployment (SSM On-Prem-initiated communication).

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa new model** **Example:** `Device(config)# aaa new model`	(Required) Enable the authentication, authorization, and accounting (AAA) access control model.
Step 4	**aaa authentication login default local** **Example:** `Device(config)# aaa authentication login default local`	(Required) Sets AAA authentication to use the local username database for authentication.
Step 5	**aaa authorization exec default local** **Example:** `Device(config)# aaa authorization exec default local`	Sets the parameters that restrict user access to a network. The user is allowed to run an EXEC shell.
Step 6	ip routing **Example:** `Device(config)# ip routing`	Enables IP routing.

	Command or Action	Purpose
Step 7	{**ip** | **ipv6**} **name-server** *server-address 1 ...server-address 6*] **Example:** `Device(config)# ip name-server vrf Mgmt-vrf 192.168.1.100 192.168.1.200 192.168.1.300`	(Optional) Specifies the address of one or more name servers to use for name and address resolution. You can specify up to six name servers. Separate each server address with a space. The first server specified is the primary server. The device sends DNS queries to the primary server first. If that query fails, the backup servers are queried.
Step 8	**ip domain lookup source-interface** *interface-type-number* **Example:** `Device(config)# ip domain lookup source-interface gigabitethernet0/0`	Enables DNS-based hostname-to-address translation on your device. This feature is enabled by default. If your network devices require connectivity with devices in networks for which you do not control name assignment, you can dynamically assign device names that uniquely identify your devices by using the global Internet naming scheme (DNS). **Note** If you configure this command on a Layer 3 physical interface, it is automatically removed from running configuration in case the port mode is changed or if the device reloads. The only available workaround is to reconfigure the command. Starting with Cisco IOS XE Dublin 17.12.1, this issue is resolved.
Step 9	**ip domain name** *name* **Example:** d `Device(config)# ip domain name vrf Mgmt-vrf cisco.com`	Defines a default domain name that the software uses to complete unqualified hostnames (names without a dotted-decimal domain name).
Step 10	**no username** *name* **Example:** `Device(config)# no username admin`	(Required) Clears the specified username, if it exists. For *name* , enter the same username you will create in the next step. This ensures that a duplicate of the username you are going to create in the next step does not exist. If you plan to use REST APIs for SSM On-Prem-initiated retrieval of RUM reports, you have to log in to SSM On-Prem. Duplicate usernames may cause the feature to work incorrectly if there are present in the system.

	Command or Action	Purpose

Step 11	**username** *name* **privilege** *level* **password** *password*
	Example:
	`Device(config)# username admin privilege 15`
	`password 0 lab`
		(Required) Establishes a username-based authentication system.
		The **privilege** keyword sets the privilege level for the user. A number between 0 and 15 that specifies the privilege level for the user.
		The password allows access to the name argument. A password must be from 1 to 25 characters, can contain embedded spaces, and must be the last option specified in the **username** command.
		This enables SSM On-Prem to use the product instance native REST.
		Note Enter this username and password in SSM On-Prem (Adding One or More Product Instances (SSM On-Prem UI), on page 184). This enables SSM On-Prem to collect RUM reports from the product instance.

Step 12	**interface** *interface-type-number* **Example:** `Device (config)# interface gigabitethernet0/0`	Enters interface configuration mode and specifies the Ethernet interface, subinterface, or VLAN to be associated with the VRF.
Step 13	**vrf forwarding** *vrf-name* **Example:** `Device(config-if)# vrf forwarding Mgmt-vrf`	Associates the VRF with the Layer 3 interface. This command activates multiprotocol VRF on an interface
Step 14	**ip address** *ip-address mask* **Example:** `Device(config-if)# ip address 192.168.0.1 255.255.0.0`	Defines the IP address for the VRF.
Step 15	**negotiation auto** **Example:** `Device(config-if)# negotiation auto`	Enables auto-negotiation operation for the speed and duplex parameters of an interface.
Step 16	**no shutdown** **Example:** `Device(config-if)# no shutdown`	Restarts a disabled interface.
Step 17	**end** **Example:**	Exits the interface configuration mode and enters global configuration mode.

	Command or Action	Purpose
	`Device(config-if)# end`	
Step 18	**ip http server** **Example:** `Device(config)# ip http server`	(Required) Enables the HTTP server on your IP or IPv6 system, including a Cisco web browser user interface. The HTTP server uses the standard port 80, by default.

Step 19	**ip http authentication local**
	Example:
	`ip http authentication local`
	`Device(config)#`
		(Required) Specifies a particular authentication method for HTTP server users.
		The **local** keyword means that the login user name, password and privilege level access combination specified in the local system configuration (by the username global configuration command) should be used for authentication and authorization.

Step 20	**ip http secure-server** **Example:** `Device(config)# ip http server`	(Required) Enables a secure HTTP (HTTPS) server. The HTTPS server uses the Secure Sockets Layer (SSL) version 3.0 protocol.
Step 21	**ip http max-connections** **Example:** `Device(config)# ip http max-connections 16`	(Required) Configures the maximum number of concurrent connections allowed for the HTTP server. Enter an integer in the range from 1 to 16. The default is 5.
Step 22	**ip tftp source-interface** *interface-type-number* **Example:** `Device(config)# ip tftp source-interface GigabitEthernet0/0`	Specifies the IP address of an interface as the source address for TFTP connections.
Step 23	**ip route** *ip-address ip-mask subnet mask* **Example:** `Device(config)# ip route vrf mgmt-vrf 192.168.0.1 255.255.0.0 192.168.255.1`	Configures a route and gateway on the product instance. You can configure either a static route or a dynamic route.

Step 24	**logging host**
	Example:
	`Device(config)# logging host 172.25.33.20`
	`vrf Mgmt-vrf`
		Logs system messages and debug output to a remote host.

Step 25	**crypto pki trustpoint SLA-TrustPoint**
	Example:
	`Device(config)# crypto pki trustpoint SLA-TrustPoint`
	`Device(ca-trustpoint)#`
		(Required) Declares that the product instance should use trustpoint "SLA-TrustPoint" and enters the ca-trustpoint configuration mode. The product instance does not recognize any trustpoints until you declare a trustpoint using this command.

	Command or Action	Purpose
Step 26	enrollment terminal **Example:** `Device(ca-trustpoint)# enrollment terminal`	(Required) Specifies the certificate enrollment method.
Step 27	**revocation-check none** **Example:** `Device(ca-trustpoint)# revocation-check none`	(Required) Specifes a method that is to be used to ensure that the certificate of a peer is not revoked. For the SSM On-Prem Deployment topology, enter the **none** keyword. This means that a revocation check will not be performed and the certificate will always be accepted.

Step 28	**end**
	Example:
	`Device(ca-trustpoint)# exit`
	`Device(config)# end`
		Exits the ca-trustpoint configuration mode and then the global configuration mode and returns to privileged EXEC mode.

Step 29	**show ip http server session-module** **Example:** `Device# show ip http server session-module`	(Required) Verifies HTTP connectivity. In the output, check that `SL_HTTP` is active. Additionally, you can also perform the following checks : • From device where SSM On-Prem is installed, verify that you can ping the product instance. A successful ping confirms that the product instance is reachable. • From a Web browser on the device where SSM On-Prem is installed verify `https://<product-instance-ip>/`. This ensures that the REST API from SSM On-Prem to the product instance works as expected.
Step 30	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	Saves your entries in the configuration file.

Submitting an Authorization Code Request (SSM On-Prem UI)

With the SSM On-Prem Deployment topology, the authorization codes required for export-controlled and enfored licenses must be generated in CSSM and imported into SSM On-Prem before the product instance can request the same. This procedure shows you the steps you have to complete in SSM On-Prem (to submit the request and then import SLAC), points you to the procedure you have to complete in CSSM (to generate and download SLAC), and to the procedure you have to complete on the product instance (to finally request and install SLAC).

Before you begin

Supported topologies:

- SSM On-Prem Deployment (SSM On-Prem-initiated communication)
- SSM On-Prem Deployment (product instance-initiated communication).

Ensure that you have an adequate positive balace of the necessary export-controlled or enforced licenses in your Smart Account and Virtual Account in CSSM.

Procedure

Step 1 Log into SSM On-Prem and select **Smart Licensing**.

Step 2 Navigate to **Inventory** > **SL Using Policy**. Select all the product instances for which you want to request SLAC.

Step 3 Click **Actions for Selected…** > **Authorization Code Request**.

The **Authorization Request Information** pop-up window is displayed.

Step 4 Click **Accept** and save the .csv file when prompted.

The generated .csv file contains the list of selected product instances along with required device information, in the required format, to generate the SLAC in CSSM. Save this file in a location that is accessible when you are working on the CSSM Web UI (in the next step).

Step 5 Complete this task in CSSM: Generating and Downloading SLAC from CSSM to a File, on page 199.

You can use the above procedure to generate SLAC for a single product instance and for multiple product instances. For the SSM On-Prem Deployment topology, follow the steps to generate SLAC for multiple product instances.

Step 6 Again navigate to **Inventory** > **SL Using Policy**.

Step 7 Click **Export/Import All…** > **Import From Cisco**.

Import the .csv file download at the end of the procedure in Step 4 above.

To verify import, under **Inventory** > **SL Using Policy**, see the Alerts column. The following message is displayed: Authorization message received from CSSM.

Step 8 Complete the final step depending on whether the product instance or SSM On-Prem initiates communication.

- For product instance-initiated communication, configure the product instance to request and install SLAC from SSM On-Prem. See: Manually Requesting and Auto-Installing a SLAC , on page 191
- For SSM On-Prem-initiated communication, the uploaded codes are applied to the product instances the next time SSM On-Prem runs an update.

Manually Requesting and Auto-Installing a SLAC

To request CSSM or CSLU or SSM On-Prem for a SLAC and have it automatically installed on the product instance, perform the following steps on the product instance:

Before you begin

Supported topologies:

- Connected to CSSM Through CSLU (product instance-initiated and CSLU-initiated communication)
- Connected Directly to CSSM
- CSLU Disconnected from CSSM (product instance-initiated and CSLU-initiated communication)
- SSM On-Prem Deployment (product instance-initiated communication)

Before you proceed, check the following as well:

- You have the required number of HSECK9 keys in the applicable Smart Account and Virtual Account in CSSM.

  Each UDI where you want to use a cryptographic feature requires one HSECK9 key. Each HSECK9 key requires a SLAC. When you follow this task to request and install SLAC on the product instance, the usage count of the HSECK9 key is updated accordingly in CSSM.

Note The following restriction applies only to Cisco Catalyst 9400 Series Supervisor Modules supporting the HSECK9 key: In a Cisco StackWise Virtual set-up, when requesting SLAC for a product instance that is connected to CSLU or SSM On-Prem, even if you use the option to request SLAC only for the active (the **local** keyword), SLAC is requested and installed for the active *and* standby. You must therefore ensure that you have two available HSECK9 keys - one for each chassis UDI - in the Smart Account and Virtual Account in CSSM. A corresponding SLAC is then installed for each chassis UDI.

This restriction does not affect a single or a dual-supervisor setup, because only one HSECK9 and one corresponding SLAC is required in these setups.

- The product instance on which you are requesting the SLAC is connected CSSM, or CSLU, or SSM On-Prem.
- The transport type and URL are configured accordingly. In the **show license all** command in privileged EXEC mode. In the output, check field `Transport: .`
- You have installed a trust code by generating a token, if you are directly connected to CSSM. Enter the **show license all** command in privileged EXEC mode. In the output check field `Trust Code Installed:`
- In case of an SSM On-Prem Deployment, the product instance requests SSM On-Prem for SLAC, so ensure that you have made the required number of SLACs available in the SSM On-Prem server before you can begin with this task.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.

	Command or Action	Purpose

Step 2	**license smart authorization request** {**add** | **replace**} *feature_name* {**all** | **local**}
	Example:
	`Device# license smart authorization request add hseck9 all`
		Requests a SLAC from CSSM or CSLU or SSM On-Prem.
		• Specify if you want to add to or replace an existing SLAC:
		  • **add**: This adds the requested key to an existing SLAC. The new SLAC will contain all the keys of the existing SLAC, and the requested key.
		  • **replace**: This replaces the existing SLAC. The new SLAC will contain only the requested key. All HSECK9 keys in the existing SLAC are returned. When you enter this keyword, the product instance checks if these existing keys are in-use. If they are, an error message is displayed, telling you to first disable the corresponding cryptographic feature.
		Note On Cisco Catalyst 9300X Series Switches in a stacking setup: If you have added a device (where SLAC is not installed) to an existing stack where SLAC is already installed, use the **replace** and **all** keywords. This returns all HSECK9 keys in the existing SLAC and requests SLAC for all the devices in the stack. You cannot request SLAC for a particular member. Your only options are: either the active, or the entire stack.

	Command or Action	Purpose
		Note This keyword is not supported on Cisco Catalyst 9400 Series Supervisor Modules in a Cisco StackWise Virtual set-up. If SLAC is installed only on the active and you want to install it on the standby as well, return the SLAC which is on the active and then request and install SLAC on the active and standby again. • *feature_name*: Enter the name of the export-controlled license for which you want to request an addition or a replacement of the SLAC. Enter "hseck9" to request and install SLAC for the HSECK9 key. • Specify the device by entering one of these options: • **all**: Gets the authorization code for *all* devices in a High Availability and stacking set-up. In case of a stacking setup or a Cisco StackWise Virtual setup, we recommend that you use this option and install SLAC for the active and the standby. This ensures uninterrupted use of the cryptographic feature in the event of a switchover. • **local**: Gets the authorization code for the *active* device in a High Availability and stacking set-up. This is the default option.
Step 3	(Optional) **license smart sync** {**all** | **local**} **Example:** `Device# license smart sync all`	Triggers the product instance to synchronize with CSSM, or CSLU, or SSM On-Prem, to send and receive any pending data. This step applies only to topologies where the product instance is connected to CSSM, or CSLU or SSM On-Prem, and where the product instance initiates communication. The

	Command or Action	Purpose
		topologies are: *Connected Directly to CSSM*, *Connected to CSSM Through CSLU* (product instance-initiated), and SSM On-Prem Deployment (product instance-initiated). By triggering an on-demand synchronization, you can ensure that the SLAC installation process is completed soon after you request SLAC. Otherwise, SLAC is applied to the product instance only the next time the product instance is *scheduled* to contact CSSM, or CSLU or SSM On-Prem.
Step 4	Complete remaining steps for applicable topologies.	• For *Connected to CSSM Through CSLU* (CSLU-initiated communication), see Tasks for CSLU-Initiated Communication, on page 125. • For *CSLU Disconnected from CSSM* (product instance-initiated and CSLU-initiated communication), see Workflow for Topology: CSLU Disconnected from CSSM, on page 128. • For *SSM On-Prem Deployment* (product instance-initiated communication), see Workflow for Topology: SSM On-Prem Deployment, on page 132

Step 5	**show license authorization**
	
	Example:
	`Device# show license authorization`
	`Overall status:`
	`  Active: PID:C9300X-24HX,SN:FOC2519L8R7`
	
	`      Status: SMART AUTHORIZATION INSTALLED on`
	`Oct 29 17:45:28 2021 UTC`
	`      Last Confirmation code: 6746c5b5`
	`  Standby:`
	`PID:C9300X-48HXN,SN:FOC2524L39P`
	`      Status: NOT INSTALLED`
	`  Member: PID:C9300X-48HX,SN:FOC2516LC92`
	
	`      Status: NOT INSTALLED`
	
	`Authorizations:`
	`  C9K HSEC (Cat9K HSEC):`
	`    Description: HSEC Key for Export Compliance on`
	`Cat9K Series Switches`
	`    Total available count: 1`
	`    Enforcement type: EXPORT RESTRICTED`
	`    Term information:`
	`      Active:`
		Displays the SLAC that is installed on the product instance.

	Command or Action	Purpose
	PID:C9300X-24HX,SN:FOC2519L8R7 Authorization type: SMART AUTHORIZATION INSTALLED License type: PERPETUAL Term Count: 1 Purchased Licenses: No Purchase Information Available	
Step 6	Configure the cryptographic feature. **Example:** Device# **show license summary** License Usage: License Tag Entitlement Count Status --- network-advantage (C9300-24 Network Advan...) 1 IN USE dna-advantage (C9300-24 DNA Advantage) 1 IN USE network-advantage (C9300-48 Network Advan...) 2 IN USE dna-advantage (C9300-48 DNA Advantage) 2 IN USE hseck9 (Cat9K HSEC) 1 IN USE	After you configure the cryptographic feature, the usage count and status of HSECK9 key in the output of the **show license summary** privileged EXEC command changes to `1` and `IN USE`, respectively Depending on the cryptopgraphic feature and the product instance, refer to the correspoding document: For information about disabling the IPsec feature on Cisco Catalyst 9300X Series Switches, see the *Configuring IPsec* chapter of the *Security Configuration Guide, Cisco IOS XE <applicable release number> (Catalyst 9300 Switches)*. For information about disabling the IPsec feature on Cisco Catalyst 9400 Series Supervisor 2 and 2XL Modules, see the *Configuring IPsec* chapter of the *Security Configuration Guide, Cisco IOS XE <applicable release number> (Catalyst 9400 Switches)*. For information about disabling the WANMACsec feature on Cisco Catalyst 9500X Series Switches, see the *MACsec Encryption* chapter of the *Security Configuration Guide, Cisco IOS XE <applicable release number> (Catalyst 9500 Switches)* For information about disabling the WANMACsec feature on Cisco Catalyst 9600 Series 40-Port 50G, 2-Port 200G, 2-Port 400G Line Card, see the *MACsec Encryption* chapter of the *Security Configuration Guide, Cisco IOS XE <applicable release number> (Catalyst 9600 Switches)*

Generating and Saving a SLAC Request on the Product Instance

To generate and then save a SLAC request for an HSECK9 key to a file on the product instance, complete the following task:

Note This method of requesting a SLAC is supported starting with Cisco IOS XE Cupertino 17.7.1 only.

Before you begin

Supported topologies: No Connectivity to CSSM and No CSLU

Also ensure that you have the required number of HSECK9 keys in the applicable Smart Account and Virtual Account in CSSM. Each UDI where you want to use a cryptographic feature requires one HSECK9 key. Each HSECK9 key requires a SLAC. After you complete this task you have to upload the SLAC request file in CSSM. Once this is processed in CSSM, the usage count of the HSECK9 key is updated accordingly in CSSM.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**license smart authorization request** {**add** | **replace**} *feature_name* {**all**| **local**} **Example:** `Device# license smart authorization request add hseck9 all`	Generates a SLAC request with all the required information. Specify if you want to add to or replace an existing SLAC: • **add**: Adds the requested key to an existing SLAC. The new authorization code will contain all the keys of the existing SLAC, and the requested license. • **replace**: Replaces the existing SLAC. The new SLAC will contain only the requested HSECK9 key. All keys in the existing SLAC are returned. When you enter this keyword, the product instance checks if these existing keys are in-use. If they are, an error message is displayed, telling you to first disable the corresponding feature.

	Command or Action	Purpose
		Note For a stacking scenario (Cisco Catalyst 9300X Series Switches): If you have added a device (where SLAC is not installed) to an existing stack where SLAC is already installed, use the **replace** and **all** keywords. This returns all HSECK9 keys in the existing SLAC and requests SLAC for all the devices in the stack. You cannot request SLAC for a particular member. Your only options are: either the active, or the entire stack. **Note** This keyword is not supported on Cisco Catalyst 9400 Series Supervisor Modules in a Cisco StackWise Virtual set-up. If SLAC is installed only on the active and you want to install it on the standby as well, return the SLAC which is on the active and then request and install SLAC on the active and standby again. For *feature_name*, enter the name of the export-controlled license for which you want to request an addition or a replacement of the SLAC. Enter "hseck9" to request and install SLAC for the HSECK9 key. Specify the device by entering one of these options: • **all**: Gets the SLAC for *all* devices in a High Availability set-up In case of a stacking setup or a Cisco StackWise Virtual setup, we recommend that you use this option and install SLAC for the active and the standby. This ensures uninterrupted use of the cryptographic feature in the event of a switchover.

	Command or Action	Purpose
		• **local**: Gets the SLAC for the *active* device in a High Availability set-up. This is the default option.
Step 3	**license smart authorization request save***path* **Example:** `Device# license smart authorization request save bootflash:slac.txt`	Saves the required UDI information for the SLAC request in a .txt file, in the specified location.
Step 4	Upload the file to CSSM, and then download the file containing the SLAC code.	Complete this task: Uploading Data or Requests to CSSM and Downloading a File, on page 210.
Step 5	Install the file on the product instance.	Complete this task: Installing a File on the Product Instance, on page 211.

Generating and Downloading SLAC from CSSM to a File

You can use this procedure to generate SLAC for a single product instance and for multiple product instances.

If it is for a single product instance, you will require the PID and serial number to complete this task. On the product instance, enter the **show license udi** command in privileged EXEC mode and keep this information handy.

If it is for multiple product instances, have the .csv file containing the PIDs and serial numbers of all applicable product instances saved in an accessible location.

Before you begin

Supported topologies:

- Connected to CSSM Through CSLU (Product instance-initiated and CSLU-initiated)
- CSLU Disconnected from CSSM (Product instance-initiated and CSLU-initiated)
- No Connectivity to CSSM and No CSLU
- SSM On-Prem Deployment (product instance-initiated and SSM On-Prem-initiated communication)

Procedure

Step 1 Log in to the CSSM Web UI at https://software.cisco.com. Under **Smart Software Licensing**, click the **Manage licenses** link.

Log in using the username and password provided by Cisco.

Step 2 Click the **Inventory** tab.

Step 3 From the **Virtual Account** drop-down list, choose the applicable virtual account.

Step 4 Click the **Product Instances** tab.

Step 5 Click the **Authorize License Enforced Features** tab.

Step 6 Generate SLAC for a single product instance or for multiple product instances (*choose one*).

- **To generate SLAC for a single product instance:**
  a. Enter the **PID** and **Serial Number**.

     Note Do not populate any of the other fields.

  b. Choose the license, and in the corresponding **Reserve** column, and enter `1`.

     Ensure that you choose the correct license for a PID. For Cisco Catalyst Access, Core, and Aggregation Switches where the HSECK9 is supported, select "C9K HSEC".

  c. Click **Next**

  d. Click **Generate Authorization Code**.

  e. Download the authorization code and save as a .csv file.

  f. Install the file on the product instance. See Installing a File on the Product Instance, on page 211.

- **To generate SLAC for multiple product instances (you should have a .csv file to upload in this case)**:
  a. From the dropdown list that says “Single Device” (by default), change the selection to “Multiple Devices”.

     At this point, a "Download a template" link is displayed. If you don't already have the required template or file, you can download it. Only the serial number PID are mandatory.

  b. Click **Choose File** and navigate to the .csv file, which contains the list of product instances that require SLAC.

  c. Once uploaded, the list of devices is displayed in CSSM. All the devices will have the checkbox enabled (implying that you want to request a SLAC for all of them), and click **Next**.

  d. Specify the license quantity required for each product instance, and click **Next**.

     Note For the "C9K HSEC" license, one SLAC is required for each UDI.

  e. Click **Reserve Licenses**.

  f. Download accordingly to topology:

     - For the *Connected to CSSM Through CSLU*, *CSLU Disconnected from CSSM*, *SSM On-Prem Deployment* topologies, click **Download Authorization Codes** to download a.csv file containing all the authorization codes. Click **Close**.

       You can now import this .csv file to CSLU or SSM On-Prem. Return to the CSLU or SSM On-Prem interface to complete the remaining steps to import this file.

     - For the *No Connectivity to CSSM and No CSLU* topology (in an air-gapped network), where you have to import the code into the product instance, download the authorization code for each product instance to a separate .txt file. Do not download the .csv file which has all the codes.

       In the CSSM Web UI, return to the **Inventory**> **Product Instances** tab. Locate each product instance by its PID or serial number. Click on the UDI to display the **Overview** tab. The **Last Contact** field displays a link called *Download Reservation Authorization Code*. Click on the link to download the authorization code of only the selected product instance, in .txt format.

Import each SLAC into the product instance, see Installing a File on the Product Instance, on page 211.

Returning an Authorization Code

This task shows you how to return an authorization code for a license and to then return the license to your license pool in CSSM. You can use this procedure for all authorization codes - SLAC and SLR.

Before you begin

Supported topologies: all

Procedure

	Command or Action	Purpose
Step 1	Disable or unconfigure the cryptographic feature for which you used the HSECK9 key.	Depending on the cryptopgraphic feature and the product instance, refer to the correspoding document: For information about disabling the IPsec feature on Cisco Catalyst 9300X Series Switches, see the *Configuring IPsec* chapter of the *Security Configuration Guide, Cisco IOS XE <applicable release number> (Catalyst 9300 Switches)*. For information about disabling the IPsec feature on Cisco Catalyst 9400 Series Supervisor 2 and 2XL Modules, see the *Configuring IPsec* chapter of the *Security Configuration Guide, Cisco IOS XE <applicable release number> (Catalyst 9400 Switches)*. For information about disabling the WANMACsec feature on Cisco Catalyst 9500X Series Switches, see the *MACsec Encryption* chapter of the *Security Configuration Guide, Cisco IOS XE <applicable release number> (Catalyst 9500 Switches)* For information about disabling the WANMACsec feature on Cisco Catalyst 9600 Series 40-Port 50G, 2-Port 200G, 2-Port 400G Line Card, see the *MACsec Encryption* chapter of the *Security Configuration Guide, Cisco IOS XE <applicable release number> (Catalyst 9600 Switches)*

	Command or Action	Purpose
		If the cryptographic feature you are disabling is the WAN MACsec feature, also note the following: Even after disabling the cryptographic feature, the output of the **show license summary** command displays the usage count and status for the HSECK9 key as `1` and `IN USE`. This is as expected. The steps in this task show you how to *release* the key, which changes the count and status to `0` and `NOT IN USE`. But you must disable the WAN MACsec feature before you try to release the HSECK9 key.
Step 2	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.

Step 3	**show license summary**
	Example:
	`Device# show license summary`
	`License Usage:`
	`License Entitlement Tag Count Status`
	`-----------------------------------------------------`
	`network-advantage (C9300-24 Network Advan...) 1 IN USE`
	`dna-advantage (C9300-24 DNA Advantage) 1 IN USE`
	`network-advantage (C9300-48 Network Advan...) 2 IN USE`
	`dna-advantage (C9300-48 DNA Advantage) 2 IN USE`
	`C9K HSEC (Cat9K HSEC) 1 IN USE`
		(Optional) Displays license usage summary. This step applies only if you are returning a SLAC.
		If the status of the HSECK9 key is displayed as `NOT IN USE` skip to Step 5.
		If the status of the HSECK9 key is displayed as `IN USE` even after the cryptographic feature is disabled, then perform the next step. This is the case in the accompanying example.

Step 4	Depending on the cryptographic feature you were using, enter the applicable command to release the HSECK9 key.
	• For IPSec: **platform hsec-license-release**
	• For WAN MACsec: **platform wanmacsec hsec-license-release**
	Example:
	`Device# configure terminal`
	`Device(config)# platform hsec-license-release`
	`HSEC license is released`
	`Device(config)# exit`
		(Optional) Enters the global configuration mode, releases the HSECK9 key, and returns to privileged EXEC mode. This step applies only if you are returning a SLAC.
		If the cryptographic feature using the HSECK9 key has been disabled or unconfigured, and the license is still displayed as `IN USE`, this command forces the HSECK9 key to be marked as `NOT IN USE`. If the status of the HSECK9 key is still displayed as `IN USE`, repeat Step 1.

	Command or Action	Purpose

Step 5	**show license summary**
	Example:
	`Device# show license summary`
	`License Usage:`
	`License Entitlement Tag Count Status`
	`---`
	`network-advantage (C9300-24 Network Advan...) 1 IN USE`
	`dna-advantage (C9300-24 DNA Advantage) 1 IN USE`
	`network-advantage (C9300-48 Network Advan...) 2 IN USE`
	`dna-advantage (C9300-48 DNA Advantage) 2 IN USE`
	`C9K HSEC (Cat9K HSEC) 0 NOT IN USE`
		(Optional) Displays license usage summary. This step applies only if you are returning a SLAC.
		Ensure that the status of the license that you want to return is `NOT IN USE`.

Step 6	**license smart authorization return** {**all** |**local**} {**offline**[*path*] |**online**}
	Example:
	`Device# license smart authorization return all online`
	`OR`
	`Device# license smart authorization return all offline`
	`Enter this return code in Cisco Smart Software Manager portal:`
	`UDI: PID:C9300X-24HX,SN:FOC2519L8R7`
	`Return code: Cr9JHx-L1x5Rj-ftwzg1-h9QZAU-LE5DT1-babWeL-FABPt9-Wr1Dn7-Rp7`
	`OR`
	`Device# license smart authorization return all offline bootflash:return-code.txt`
		Returns an authorization code back to the license pool in CSSM. A return code is displayed after you enter this command.
		Specify the product instance:
		• **all**: Performs the action for all connected product instances in a High Availability or stacking set-up.
		• **local**: Performs the action for the active product instance. This is the default option.
		Specify if you are connected to CSSM or not:
		• If the product instance is directly connected to CSSM, or it is connected to CSSM through CSLU or SSM On-Prem and the product instance-initiates communication, enter **online**. The code is automatically returned to CSSM and a confirmation is returned and installed on the product instance. If you choose this option, the return code is automatically submitted to CSSM.
		• If the product instance is not connected to CSSM, or if you have implemented a topology with CSLU-initiated or SSM On-Prem initiated communication, enter **offline** [*filepath_filename*].
		If you choose the offline option, you must complete the additional step of submitting this to CSSM.

	Command or Action	Purpose
		• For software versions Cisco IOS XE Cupertino 17.7.1 and later only: Specify a path to save the SLAC return request in a file and upload the file to CSSM: Uploading Data or Requests to CSSM and Downloading a File, on page 210. The file format can be any readable format. For example: `Device# license smart authorization return local offline bootflash:return-code.txt`. • For software versions prior to 17.7.1: If you are returning a SLAC, copy the return code that is displayed on the CLI and complete this task to enter the return code in CSSM: Entering a SLAC Return Code in CSSM and Removing a Product Instance, on page 205. • For all software versions, if you are returning an SLR authorization code, copy the return code that is displayed on the CLI and complete this task to enter the return code in CSSM: Entering an SLR Return Code in CSSM and Removing the Product Instance, on page 206. Proceed with the next step only after you complete this step.

Step 7	**no license smart reservation**
	
	Example:
	`Device# configure terminal`
	`Device(config)# no license smart reservation`
	`Device(config)# exit`
		Enter the global configuration mode, disables SLR configuration on the product instance, and returns to privileged EXEC mode.
		
		This step is required only if the authorization code you are returning is an SLR authorization code. Skip this step if the code you are returning is a SLAC for an HSECK9 key.

	Command or Action	Purpose
		Note You must complete the authorization code return process (**license smart authorization return**), online or offline, before you enter the **no license smart reservation** command in this step. Otherwise, the return may not be reflected in CSSM or in the **show** command, and you will have to contact your Cisco technical support representative to rectify the problem.

Step 8	**show license authorization**
	Example:
	`Device# show license authorization`
	`Overall status:`
	`  Active: PID:C9300X-24HX,SN:FOC2519L8R7`
	`      Status: NOT INSTALLED`
	`      Last return code: Cr9JHx-L1x5Rj-ftwzg1-h9QZAU-LE5DT1-babWeL-FABPt9-Wr1Dn7-Rp7`
	`  Standby: PID:C9300X-48HXN,SN:FOC2524L39P`
	`      Status: NOT INSTALLED`
	`  Member: PID:C9300X-48HX,SN:FOC2516LC92`
	`      Status: NOT INSTALLED`
	`<output truncated>`
		Displays licensing information. If the return process is completed correctly, the `Last return code:` field displays the return code.

Entering a SLAC Return Code in CSSM and Removing a Product Instance

You can use this task to complete the return procedure for a SLAC when the product instance is not connected to CSSM. This returns the HSECK9 keys to the license pool. Additionally, you also have the option of removing the product instance from CSSM.

Before you begin

Supported topologies: all

Follow this procedure only if you are returning a SLAC.

Ensure that you have generated a return code as shown in Returning an Authorization Code, on page 201. (Enter it in Step 7 in this task).

Procedure

Step 1 Log in to the CSSM Web UI at https://software.cisco.com. Under **Smart Software Licensing**, click the **Manage licenses**link.

Log in using the username and password provided by Cisco.

Step 2 Click the **Inventory** tab.

Step 3 From the **Virtual Account** drop-down list, choose your Virtual Account.

Step 4 Click the **Product Instances** tab.

The list of product instances that are available is displayed.

Step 5 Locate the required product instance from the product instances list. You can enter the PID or serial number in the search tab to locate it.

Step 6 In the Actions column of the product instance, from the **Actions** dropdown list, select **Remove**.

The **Remove Reservation** window is displayed.

Step 7 In the **Reservation Return Code** field, enter the SLAC return code you generated.

Step 8 Click **Remove Reservation**.

The HSECK9 key is returned to the license pool. The Remove Reservation window is automatically closed and you return to the **Product Instances** tab.

Note If you want to only return the SLAC, your task ends here. If you also want to remove the product instance from CSSM, continue to the next step.

Step 9 In the Actions column of the product instance, from the **Actions** dropdown list, *again* select **Remove**.

The **Confirm Remove Product Instance** window is displayed.

Step 10 Click **Remove Product Instance**.

The product instance is removed from CSSM and no longer consumes any licenses.

Entering an SLR Return Code in CSSM and Removing the Product Instance

You can use this task to complete the return prcedure for an SLR authorization code. This returns the licenses to the license pool and removes the product instance.

Before you begin

Supported topologies: all

Follow this procedure only if you are returning an SLR authorization code.

Ensure that you have generated a return code as shown in Returning an Authorization Code, on page 201. (Enter it in Step 7 in this task).

Procedure

Step 1 Log in to the CSSM Web UI at https://software.cisco.com. Under **Smart Software Licensing**, click the **Manage licenses**link.

Log in using the username and password provided by Cisco.

Step 2 Click the **Inventory** tab.

Step 3 From the **Virtual Account** drop-down list, choose your Virtual Account.

Step 4 Click the **Product Instances** tab.

The list of product instances that are available is displayed.

Step 5 Locate the required product instance from the product instances list. You can enter the PID or serial number in the search tab to locate it.

Step 6 In the Actions column of the product instance, from the **Actions** dropdown list, select **Remove**.

- If the product instance is *not* using a license with an SLR authorization code then the **Confirm Remove Product Instance** window is displayed.
- If the product instance *is* using a license with an SLR authorization code, then the **Remove Product Instance** window, with a field for return code entry is displayed.

Step 7 In the **Reservation Return Code** field, enter the return code you generated.

Note This step applies only if the product instance is using a license with an SLR authorization code.

Step 8 Click **Remove Product Instance**.

The license is returned to the license pool and the product instance is removed.

Generating a New Token for a Trust Code from CSSM

To generate a token to request a trust code, complete the following steps.

Generate one token for each *Virtual Account* you have. You can use same token for all the product instances that are part of one Virtual Account.

Before you begin

Supported topologies: Connected Directly to CSSM

Procedure

Step 1 Log in to the CSSM Web UI at https://software.cisco.com and click **Smart Software Licensing**.

Log in using the username and password provided by Cisco.

Step 2 Click the **Inventory** tab.

Step 3 From the **Virtual Account** drop-down list, choose the required virtual account

Step 4 Click the **General** tab.

Step 5 Click **New Token**. The **Create Registration Token** window is displayed.

Step 6 In the **Description** field, enter the token description

Step 7 In the **Expire After** field, enter the number of days the token must be active.

Step 8 (Optional) **In the Max. Number of Uses** field, enter the maximum number of uses allowed after which the token expires.

Note If you enter a value here, ensure that you stagger the installation of the trust code during the next part of the process. If you want to simultaneously install the trust code on a large number of product instances, we recommend that you leave this field blank. Entering a limit here and simultaneously installing it on a large number of devices causes a bottleneck in the processing of these requests in CSSM and installation on some devices may fail, with the following error:

```
Failure Reason: Server error occurred: LS_LICENGINE_FAIL_TO_CONNECT.
```

Step 9 Click **Create Token**.

Step 10 You will see your new token in the list. Click **Actions** and download the token as a `.txt` file.

Establishing Trust with an ID Token.

This task shows you how to establish trust. Here, you use the ID token downloaded from CSSM and submit a trust request. CSSM responds with the trust code, which is automatically installed on the product instance.

Before you begin

Supported topologies: Connected Directly to CSSM

You must have already generated and downloaded an ID token file from CSSM: Generating a New Token for a Trust Code from CSSM, on page 207.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted
Step 2	**license smart trust idtoken** *id_token_value* {**local** | **all**} [**force**] **Example:** `Device# license smart trust idtoken NGMwMjk5mYtNZaxMS00NzMZmtgWm all force`	Establishes a trusted connection with CSSM. For *id_token_value*, enter the token you generated in CSSM. Enter one of following options: • **local**: Submits the trust request only for the active device in a High Availability set-up. This is the default option. • **all**: Submits the trust request for all devices in a High Availability set-up.

	Command or Action	Purpose
		Enter the **force** keyword to submit the trust code request in spite of an existing trust code on the product instance. Trust codes are node-locked to the UDI of the product instance. If a UDI is already registered, CSSM does not allow a new registration for the same UDI. Entering the **force** keyword sets a force flag in the message sent to CSSM to create a new trust code even if one already exists. You may for example need to use the **force** keyword if there is already a factory-installed trust code on the product instance. A trust code is factory-installed starting with Cisco IOS XE Cupertino 17.7.1. Since a factory-installed trust code cannot be used for secure communication with CSSM, you must use the **force** keyword to overwrite it with the trust code obtained using the ID token. Also see: Trust Code, on page 104.

Step 3	**show license status**
	
	Example:
	`<output truncated>`
	`Trust Code Installed:`
	`  Active: PID:C9500-24Y4C,SN:CAT2344L4GH`
	
	`    INSTALLED on Sep 04 01:01:46 2020 EDT`
	`  Standby: PID:C9500-24Y4C,SN:CAT2344L4GJ`
	
	`    INSTALLED on Sep 04 01:01:46 2020 EDT`
		Displays date and time if trust code is installed. Date and time are in the local time zone. See field `Trust Code Installed:`.

Downloading a Policy File from CSSM

If you have requested a custom policy or if you want to apply a policy that is different from the default that is applied to the product instance, complete the following task:

Before you begin

Supported topologies:

- No Connectivity to CSSM and No CSLU
- CSLU Disconnected from CSSM

Procedure

Step 1 Log in to the CSSM Web UI at https://software.cisco.com. Under **Smart Software Licensing**, click the **Manage licenses**link.

Log in using the username and password provided by Cisco.

Step 2 Follow this directory path: **Reports** > **Reporting Policy**.

Step 3 Click **Download**, to save the `.xml` policy file.

You can now install the file on the product instance. See Installing a File on the Product Instance, on page 211.

Uploading Data or Requests to CSSM and Downloading a File

You can use this task to:

- To upload a RUM report to CSSM and download an ACK.
- To upload a SLAC request file and download a SLAC code file.

  This applies only to the *No Connectivity to CSSM and No CSLU* topology and is supported starting with Cisco IOS XE Cupertino 17.7.1.
- To upload a SLAC or SLR authorization code return request.

  This applies only to the *No Connectivity to CSSM and No CSLU* topology and is supported starting with Cisco IOS XE Cupertino 17.7.1.

To upload a file to CSSM and download file when the product instance is not connected to CSSM or CSLU, or when SSM On-Prem is not connect to CSSM, complete the following task:

Before you begin

Supported topologies:

- No Connectivity to CSSM and No CSLU
- CSLU Disconnected from CSSM
- SSM On-Prem Deployment (Product instance-initiated and SSM On-Prem-initiated communication)

Procedure

Step 1 Log in to the CSSM Web UI at https://software.cisco.com. Under **Smart Software Licensing**, click the **Manage licenses** link.

Log in using the username and password provided by Cisco.

Step 2 Select the **Smart Account** that will receive the report.

Step 3 Select **Smart Software Licensing** → **Reports** → **Usage Data Files**.

Step 4 Click **Upload Usage Data**. Browse to the file location (RUM report in tar format), select, and click **Upload Data**.

Upload a RUM report (`.tar` format), or a SLAC request file (`.txt` format), or a SLAC return request file (`.txt` format).

You cannot delete a file after it has been uploaded. You can however upload another file, if required.

Step 5 From the Select Virtual Accounts pop-up, select the Virtual Account that will receive the uploaded file. The file is uploaded to Cisco and is listed in the Usage Data Files table in the Reports screen showing the File Name, time is was Reported, which Virtual Account it was uploaded to, the Reporting Status, Number of Product Instances reported, and the Acknowledgement status.

Step 6 In the Acknowledgement column, click Download to save the ACK or SLAC file for the report or request you uploaded.

You may have to wait for the file to appear in the Acknowledgement column. If there many RUM reports or requests to process, CSSM may take a few minutes.

After you download the file, import and install the file on the product instance, or transfer it to CSLU or SSM On-Prem.

Installing a File on the Product Instance

To import and install a policy, or ACK, or SLAC, on the product instance, complete the following task:

Before you begin

Supported topologies: No Connectivity to CSSM and No CSLU

You have saved the corresponding file in a location that is accessible to the product instance.

- For a policy, see Downloading a Policy File from CSSM, on page 209.
- For an ACK, see Uploading Data or Requests to CSSM and Downloading a File, on page 210.
- For a SLAC, see Uploading Data or Requests to CSSM and Downloading a File, on page 210 or Generating and Downloading SLAC from CSSM to a File, on page 199 (There are multiple ways to obtain a SLAC).

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**copy** *source filename* **bootflash:** **Example:** `Device# copy tftp://10.8.0.6/user01/example.txt bootflash:`	(Optional) Copies the file from its source location or directory to the flash memory of the product instance. You can also import the file *directly* from a remote location and install it on the product instance (next step).

	Command or Action	Purpose
		• *source*: This is the source location of file. The source can be either local or remote. • **bootflash:**: This is the destination for boot flash memory.
Step 3	**license smart import** *filepath_filename* **Example:** `Device# license smart import bootflash:example.txt`	Imports and installs the file on the product instance. For *filepath_filename*, specify the location, including the filename. After installation, a system message displays the type of file you installed. **Note** If you generated SLAC for multiple product instances (as in a stacking set-up) in the CSSM Web UI, that is, you followed the method described here: Generating and Downloading SLAC from CSSM to a File, on page 199, ensure that you download a separate .txt SLAC file for each UDI. Import and install one file at a time.
Step 4	**show license all** **Example:** `Device# show license all`	Displays license authorization, policy, and reporting information for the product instance.

Setting the Transport Type, URL, and Reporting Interval

To configure the mode of transport for a product instance, complete the following task:

Before you begin

Supported topologies: all

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	

	Command or Action	Purpose
Step 3	**license smart transport** {**automatic** | **callhome** | **cslu** | **off** | **smart**} **Example:** `Device(config)# license smart transport cslu`	Configures a mode of transport for the product instance to use. Choose from the following options: • **automatic**: Sets the transport mode **cslu**. • **callhome**: Enables Call Home as the transport mode. • **cslu**: This is the default transport mode. Enter this keyword if you are using CSLU *or* SSM On-Prem, with product instance-initiated communication. While the transport mode keyword is the same for CSLU and SSM On-Prem, the transport URLs are different. See **license smart url cslu** *cslu_or_on-prem_url* in the next step. • **off**: Disables all communication from the product instance. • **smart**: Enables Smart transport.

Step 4	**license smart url** {*url* | **cslu** *cslu_url* | **default** | **smart** *smart_url* | **utility** *smart_url*}
	Example:
	`Device(config)# license smart url cslu http://192.168.0.1:8182/cslu/v1/pi`
		Sets a URL for the configured transport mode. Depending on the transort mode you have chosen to configure in the previous step, configure the corresponding URL here:
		• *url*: If you have configured the transport mode as **callhome**, configure this option. Enter the CSSM URL exactly as follows:
		`https://tools.cisco.com/its/service/oddce/services/DDCEService`
		The **no license smart url** *url* command reverts to the default URL.
		• **cslu** *cslu_or_on-prem_url*: If you have configured the transport mode as **cslu**, configure this option with the URL for CSLU or SSM On-Prem, as applicable.
		  • If you are using CSLU, enter the URL as follows:
		`http://<cslu_ip_or_host>:8182/cslu/v1/pi`
		For `<cslu_ip_or_host>`, enter the hostname or the IP address of the windows host where you have installed CSLU. 8182 is the port number and it is the only port number that CSLU uses.

	Command or Action	Purpose

		The **no license smart url cslu** *cslu_url* command reverts to `http://cslu-local:8182/cslu/v1/pi`
		• If you are using SSM On-Prem, enter the URL as follows:
		`http://<ip>/cslu/v1/pi/<tenant ID>`
		For <ip>, enter the hostname or the IP address of the server where you have installed SSM On-Prem. The <tenantID> must be the default local virtual account ID.
		Tip You can retrieve the entire URL from SSM On-Prem. See Retrieving the Transport URL (SSM On-Prem UI), on page 183
		The **no license smart url cslu** *cslu_url* command reverts to `http://cslu-local:8182/cslu/v1/pi`
		• **default**: Depends on the configured transport mode. Only the **smart** and **cslu** transport modes are supported with this option.
		If the transport mode is set to **cslu**, and you configure **license smart url default**, the CSLU URL is configured automatically (`https://cslu-local:8182/cslu/v1/pi`).
		If the transport mode is set to **smart**, and you configure **license smart url default**, the Smart URL is configured automatically (`https://smartreceiver.cisco.com/licservice/license`).
		• **smart** *smart_url*: If you have configured the transport type as **smart**, configure this option. Enter the URL exactly as follows:
		`https://smartreceiver.cisco.com/licservice/license`
		When you configure this option, the system automatically creates a duplicate of the URL in **license smart url** *url*. You can ignore the duplicate entry, no further action is required.

	Command or Action	Purpose
		The **no license smart url smart***smart_url* command reverts to the default URL. • **utility** *smart_url*: Although available on the CLI, this option is not supported.
Step 5	**license smart usage interval** *interval_in_days* **Example:** `Device(config)# license smart usage interval 40`	(Optional) Sets the reporting interval in days. By default the RUM report is sent every 30 days. The valid value range is 1 to 3650. If you set the value to zero, RUM reports are not sent, regardless of what the applied policy specifies - this applies to topologies where CSLU or CSSM may be on the receiving end. If you set a value that is greater than zero and the transport type is set to **off**, then, between the *interval_in_days* and the policy value for `Ongoing reporting frequency(days):`, the lower of the two values is applied. For example, if *interval_in_days* is set to 100, and the value in the in the policy says `Ongoing reporting frequency (days):90`, RUM reports are sent every 90 days. If you do not set an interval, and the default is effective, the reporting interval is determined entirely by the policy value. For example, if the default value is effective and only unenforced licenses are in use, if the policy states that reporting is not required, then RUM reports are not sent.
Step 6	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 7	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	Saves your entries in the configuration file.

Configuring a Base or Add-On License

After you order and purchase a base or add-on license, you must configure the license on the device before you can use it.

This task sets a license level and requires a reload before the configured changes are effective. You can use this task to:

- Change the current license.
- Add another license. For example, if you are currently using Network Advantage and you also want to use features available with the corresponding Digital Networking Architecture (DNA) Advantage license.
- Remove a license.

Before you begin

Supported topologies: all

For information about the available base and add-on licenses, see Base and Add-On Licenses, on page 65.

Information about the licenses that you have purchased can be found in the Smart Account and Virtual Account of the product instance in the Cisco Smart Software Manager (CSSM) Web UI.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**license boot level** { **network-advantage** [**addon dna-advantage**] | **network-essentials** [**addon dna-essentials**] } **Example:** `Device(config)# license boot level network-advantage add-on dna-advantage`	Activates the configured license on the product instance. • **network-advantage** [**addon dna-advantage**]: Configures the Network Advantage license. Optionally, you can also configure the Digital Networking Architecture (DNA) Advantage license. • **network-advantage** [**addon dna-advantage**]: Configures the Network Essentials license. Optionally, you can also configure the Digital Networking Architecture (DNA) Essentials license. In the accompanying example, the DNA Advantage license will be activated on the product instance after reload.
Step 4	**exit** **Example:** `Device(config)# exit`	Returns to the privileged EXEC mode.

	Command or Action	Purpose
Step 5	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	Saves changes in the configuration file.

Step 6	**show version**
	Example:
	`Device# show version`
	
	`<output truncated>`
	`Technology Package License Information:`
	
	`------------------------------------------------`
	`Technology-package`
	`Technology-package`
	`Current Type`
	`Next reboot`
	`------------------------------------------------`
	`network-advantage Smart License`
	`network-advantage`
	`Subscription Smart License dna-advantage`
	
	`<output truncated>`
		Shows currently configured license information and the license that is applicable after reload.
		The "`Technology-package Next reboot`" column displays the change in the configured license that is effective after reload, only if you save the configuration change.
		In the accompanying example, the current license level is Network Advantage. Because the configuration change was saved, the "`Technology-package Next reboot`" column shows that the DNA Advantage license will be activated after reload.

Step 7	**reload** **Example:** `Device# reload`	Reloads the device.

Step 8	**show version**
	Example:
	`Device# show version`
	
	`<output truncated>`
	`Technology Package License Information:`
	
	`------------------------------------------------`
	`Technology-package`
	`Technology-package`
	`Current Type`
	`Next reboot`
	`------------------------------------------------`
	`network-advantage Smart License`
	`network-advantage`
	`dna-advantage Subscription Smart License dna-advantage`
	
	`<output truncated>`
		Shows currently configured license information and the license that is applicable after reload.

What to do next

After you configure a license level, the change is effective after a reload. To know if reporting is required, refer to the output of the **show license status** privileged EXEC command and check the `Next ACK deadline:` and `Next report push:` fields.

Note The change in license usage is recorded on the product instance. The next steps relating to reporting - if required - depend on your current topology.

- Connected to CSSM Through CSLU
  - Product Instance-initiated communication: No action required. Since the product instance initiates communication, it automatically sends out the RUM report at the scheduled time, as per the policy (**show license status** —> `Next report push`), to CSLU. (To manually trigger this on the product instance, enter the **license smart sync** {**all**| **local**} privileged EXEC command. This synchronizes the product instance with CSLU, to send and receive any pending data.) CSLU forwards the RUM report to CSSM and retrieves the ACK. The ACK is applied to the product instance the next time the product instance contacts CSLU.
  - CSLU-initiated communication: In the CSLU interface, collect usage from the product instance: Collecting Usage Reports: CSLU Initiated (CSLU Interface), on page 165. CSLU sends the RUM report to CSSM and retrieves the ACK from CSSM. The ACK is applied to the product instance the next time CSLU runs an update.
- Connected Directly to CSSM: No action required. Since the product instance initiates communication, it automatically sends out the RUM report at the scheduled time, as per the policy (**show license status** —> `Next report push`), to CSSM. (To manually trigger this on the product instance, enter the **license smart sync** {**all**| **local**} privileged EXEC command. This synchronizes the product instance with CSSM, to send and receive any pending data.) Once the ACK is available, CSSM sends this back to the product instance.
- CSLU Disconnected from CSSM
  - Product Instance-initiated communication: No action required. Since the product instance initiates communication, it automatically sends out the RUM report at the scheduled time, as per the policy (**show license status** —> `Next report push`), to CSLU. (To manually trigger this on the product instance, enter the **license smart sync** {**all**| **local**} privileged EXEC command. This synchronizes the product instance with CSLU, to send and receive any pending data.)

    Since CSLU is disconnected from CSSM, in the CSLU interface and then the CSSM Web UI, complete these tasks Export to CSSM (CSLU Interface), on page 166 > Uploading Data or Requests to CSSM and Downloading a File, on page 210 > Import from CSSM (CSLU Interface), on page 166. The ACK is applied to the product instance the next time the product instance contacts CSLU.
  - CSLU-initiated communication: In the CSLU interface, collect usage from the product instance: Collecting Usage Reports: CSLU Initiated (CSLU Interface), on page 165.

    Since CSLU is disconnected from CSSM, in the CSLU interface and then the CSSM Web UI, complete these tasks Export to CSSM (CSLU Interface), on page 166 > Uploading Data or Requests to CSSM and Downloading a File, on page 210 > Import from CSSM (CSLU Interface), on page 166. The ACK is applied to the product instance the next time CSLU runs an update.
- Connected to CSSM Through a Controller: No action is required (if you have already completed the first ad hoc report in the Cisco DNA Center GUI). Cisco DNA Center handles all subsequent reporting and returns the ACK to the product instance.
- No Connectivity to CSSM and No CSLU: Save RUM reports to a file (on your product instance) and upload it to CSSM (from a workstation that has connectivity to the Internet, and Cisco). Enter the **license**

smart save usage command in privileged EXEC mode, to save RUM reports to a file. Then to upload the file to CSSM and download the ACK, complete this task: Uploading Data or Requests to CSSM and Downloading a File, on page 210. Lastly, to install the ACK on the product instance, complete this task: Installing a File on the Product Instance, on page 211.

- SSM On-Prem Deployment:
  - Product Instance-initiated communication: No action required. Since the product instance initiates communication, it automatically sends out the RUM report at the scheduled time, as per the policy (**show license status** —> `Next report push`), to SSM On-Prem. (To manually trigger this on the product instance, enter the **license smart sync** {**all**| **local**} privileged EXEC command. This synchronizes the product instance with SSM On-Prem, to send and receive any pending data.)
    - If SSM On-Prem is connected to CSSM, in the SSM On-Prem interface, navigate to **Reports** > **Usage Schedules** > **Synchronization schedule with Cisco**.
    - If SSM On-Prem is disconnected from CSSM, upload and download the required files for reporting: Exporting and Importing Usage Data (SSM On-Prem UI), on page 183.
  - SSM On-Prem initiated communication: In the SSM On-Prem interface, collect usage information from the product instance. Navigate to **Reports** > **Synchronisation pull schedule with the devices** > **Synchronise now with the device**.
    - If SSM On-Prem is connected to CSSM, in the SSM On-Prem interface, navigate to **Reports** > **Usage Schedules** > **Synchronization schedule with Cisco**.
    - If SSM On-Prem is disconnected from CSSM, upload and download the required files for reporting: Exporting and Importing Usage Data (SSM On-Prem UI), on page 183.

Sample Resource Utilization Measurement Report

The following is a sample Resource Utilization Measurement (RUM) report, in XML format (See RUM Report and Report Acknowledgement, on page 103). Several such reports may be concatenated to form one report.

```
<?xml version="1.0" encoding="UTF-8"?>
  <smartLicense>
```

```
  </smartLicense>
```

Troubleshooting Smart Licensing Using Policy

This section provides the list of Smart Licensing Using Policy-related system messages you may encounter, possible reasons for failure, and recommended action.

System Message Overview

The system software sends system messages to the console (and, optionally, to a logging server on another system). Not all system messages mean problems with your system. Some messages are informational, and others can help diagnose problems with communications lines, internal hardware, or the system software.

How to Read System Messages

System log messages can contain up to 80 characters. Each system message begins with a percent sign (%) and is structured as follows:

```
%FACILITY-SEVERITY-MNEMONIC: Message-text
```

%FACILITY

Two or more uppercase letters that show the facility to which the message refers. A facility can be a hardware device, a protocol, or a module of the system software

SEVERITY

A single-digit code from 0 to 7 that reflects the severity of the condition. The lower the number, the more serious the situation.

Table 14: Message Severity Levels

Severity Level	Description
0 - emergency	System is unusable.
1 - alert	Immediate action required.
2 - critical	Critical condition.
3 - error	Error condition.
4 - warning	Warning condition.
5 - notification	Normal but significant condition.
6 - informational	Informational message only.
7 - debugging	Message that appears during debugging only.

MNEMONIC

A code that uniquely identifies the message.

Message-text

Message-text is a text string describing the condition. This portion of the message sometimes contains detailed information about the event, including terminal port numbers, network addresses, or addresses that correspond to locations in the system memory address space. Because the information in these variable fields changes from message to message, it is represented here by short strings enclosed in square brackets ([ ]). A decimal number, for example, is represented as [dec].

Table 15: Variable Fields in Messages

Severity Level	Description
[char]	Single character
[chars]	Character string
[dec]	Decimal number
[enet]	Ethernet address (for example, 0000.FEED.00C0)
[hex]	Hexadecimal number
[inet]	Internet address (for example, 10.0.2.16)
[int]	Integer
[node]	Address or node name
[t-line]	Terminal line number in octal (or in decimal if the decimal-TTY service is enabled)
[clock]	Clock (for example, 01:20:08 UTC Tue Mar 2 1993

System Messages

This section provides the list of Smart Licensing Using Policy-related system messages you may encounter, possible reasons for failure (incase it is a failure message), and recommended action (if action is required).

For all error messages, if you are not able to solve the problem, contact your Cisco technical support representative with the following information:

- The message, exactly as it appears on the console or in the system log.
- The output from the **show license tech support**, **show license history message**, and the **show platform software sl-infra** privileged EXEC commands.

Smart Licensing Using Policy-related system messages:

- %SMART_LIC-3-POLICY_INSTALL_FAILED
- %SMART_LIC-3-AUTHORIZATION_INSTALL_FAILED
- %SMART_LIC-3-COMM_FAILED
- %SMART_LIC-3-COMM_RESTORED
- %SMART_LIC-3-POLICY_REMOVED
- %SMART_LIC-3-TRUST_CODE_INSTALL_FAILED
- %SMART_LIC-4-REPORTING_NOT_SUPPORTED
- %SMART_LIC-6-POLICY_INSTALL_SUCCESS
- %SMART_LIC-6-AUTHORIZATION_INSTALL_SUCCESS

- %SMART_LIC-6-AUTHORIZATION_REMOVED
- %SMART_LIC-6-REPORTING_REQUIRED
- %SMART_LIC-6-TRUST_CODE_INSTALL_SUCCESS

```
Error Message %SMART_LIC-3-POLICY_INSTALL_FAILED: The installation of a new
licensing policy has failed: [chars].
```

Explanation: A policy was installed, but an error was detected while parsing the policy code, and installation failed. [chars] is the error string with details of the failure.

Possible reasons for failure include:

- A signature mismatch: This means that the system clock is not accurate.
- A timestamp mismatch: This means the system clock on the product instance is not synchronized with CSSM.

Recommended Action:

For both possible failure reasons, ensure that the system clock is accurate and synchronized with CSSM. Configure the **ntp server** command in global configuration mode. For example:

```
Device(config)# ntp server 198.51.100.100 version 2 prefer
```

If the above does not work and policy installation still fails, contact your Cisco technical support representative.


```
Error Message %SMART_LIC-3-AUTHORIZATION_INSTALL_FAILED: The install of a new
licensing authorization code has failed on [chars]: [chars].
```

Explanation: Authorization code installation was attempted, but installation failed. The first [chars] is the UDI for which the authorization code installation failed, and the second [chars] is the error string with details of the failure.

Possible reasons for failure include:

- Not enough licenses with authorization for currently configured features: This means that you have not provided the requisite number of authorization codes.
- UDI mismatch: One or more UDIs in the authorization code file do not match with the product instance where you are installing the authorization code file. If you have generated authorization codes for multiple UDIs, for a High Availability or stacking set-up, all the UDIs listed in the authorization code file must match with all the UDIs in the High Availability or stacking set-up. If this is not the case, installation fails.

  Cross-check all UDIs in the authorizatin code file against the UDIs of the product instance (standalone or High Availability).

  ```
  Excerpt of UDI information in a SLAC file:
  <smartLicenseAuthorization>
  <udi>P:C9300X-24HX,SN:FOC2519L8R7</udi>

  <output truncated>
  </smartLicenseAuthorization>

  Sample output of UDI information on a product instance:
  ```

```
Device# show license udi
UDI: PID:C9300X-24HX,SN:FOC2519L8R7
```

- A signature mismatch: This means that the system clock is not accurate. If the clock is not synchronized, your *attempts* at requesting SLAC are not reflected in the **show license tech** output.

```
Authorization Confirmation:
  Attempts: Total=0, Success=0, Fail=0  Ongoing Failure: Overall=0 Communication=0
```

Recommended Action

- In the output of the **show license tech support** command, check the `Failure Reason:` field to understand what may have gone wrong.

```
Device# show license tech support
<output truncated>

Communication Statistics:
=========================
Authorization Confirmation:
  Attempts: Total=2, Success=2, Fail=0  Ongoing Failure: Overall=0 Communication=0
  Last Response: OK on Sep 23 17:51:52 2020 UTC
    Failure Reason: <none>
  Last Success Time: Sep 23 17:51:52 2020 UTC
  Last Failure Time: <none>
```

- Not enough licenses in authorization for currently configured features and UDI mismatch:
- Use the **show license udi** command to verify that you have the correct and complete list of UDIs. This command displays all product instances in case of High Availability and stacking set-up. Then install SLAC again.
- Signature mismatch: Ensure that the system clock is accurate and synchronized with CSSM. To do this, configure the **ntp server** command in global configuration mode. For example:

```
Device(config)# ntp server 198.51.100.100 version 2 prefer
```

After you complete this configuration, again use the **show license tech** to verify if the clock has actually synchronized. If successfully synchronized, the `Clock sync-ed with NTP` field is set to `True`. If not synchronized, this field is set to `False`.

```
Error Message %SMART_LIC-3-COMM_FAILED: Communications failure with the [chars] :
[chars]
```

Explanation: Smart Licensing communication either with CSSM, CSLU, or SSM On-Prem failed. The first [chars] is the currently configured transport type, and the second [chars] is the error string with details of the failure. This message appears for every communication attempt that fails.

Possible reasons for failure include:

- CSSM, CSLU, SSM On-Prem is not reachable: This means that there is a network reachability problem.
- 404 host not found: This means the CSSM server is down.
- A TLS or SSL handshake failure caused by a missing client certificate. The certificate is required for TLS authentication of the two communicating sides. A recent server upgrade may have cause the certificate

to be removed. This reason applies only to a topology where the product instance is directly connected to CSSM.

Note If the error message is displayed for this reason, there is no actual configuration error or disruption in the communication with CSSM.

For topologies where the product instance initiates the sending of RUM reports (Connected to CSSM Through CSLU: Product Instance-Initiated Communication, Connected Directly to CSSM, CSLU Disconnected from CSSM: Product Instance-Initiated Communication, and SSM On-Prem Deployment: Product Instance-Initiated Communication) if this communication failure message coincides with scheduled reporting (**license smart usage interval** *interval_in_days* global configuration command), the product instance attempts to send out the RUM report for up to four hours after the scheduled time has expired. If it is still unable to send out the report (because the communication failure persists), the system resets the interval to 15 minutes. Once the communication failure is resolved, the system reverts the reporting interval to last configured value.

Recommended Action:

Troubleshooting steps are provided for when CSSM is not reachable or there is a missing client certificate, when CSLU is not reachable, and when SSM On-Prem is not reachable.

- If a client certificate is missing and there is no actual configuration error or disruption in the communication with CSSM:

  To resolve the error, configure the **ip http client secure-trustpoint** *trustpoint-name* command in global configuration mode. For *trustpoint-name*, enter only `SLA-TrustPoint`. This command specifies that the secure HTTP client should use the certificate associated with the trustpoint indicated by the trustpoint-name argument.

- If CSSM is not reachable and the configured transport type is **smart**:

  1. Check if the smart URL is configured correctly. Use the **show license status** command in privileged EXEC mode, to check if the URL is exactly as follows: https://smartreceiver.cisco.com/licservice/license. If it is not, reconfigure the **license smart url smart** *smar_URL* command in global configuration mode.

  2. Check DNS resolution. Verify that the product instance can ping `smartreceiver.cisco.com` or the nslookup translated IP. The following example shows how to ping the translated IP

     ```
     Device# ping 171.70.168.183
     Type escape sequence to abort.
     Sending 5, 100-byte ICMP Echos to 171.70.168.183, timeout is 2 seconds:
     !!!!!
     Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/2 ms
     ```

- If CSSM is not reachable and the configured transport type is **callhome**:

  1. Check if the URL is entered correctly. Use the **show license status** command in privileged EXEC mode, to check if the URL is exactly as follows: https://tools.cisco.com/its/service/oddce/services/DDCEService.

  2. Check if Call Home profile `CiscoTAC-1` is active and destination URL is correct. Use the **show call-home profile all** command in privileged EXEC mode:

     ```
     Current smart-licensing transport settings:
      Smart-license messages: enabled
      Profile: CiscoTAC-1 (status: ACTIVE)
     ```

```
Destination URL(s): https://tools.cisco.com/its/service/oddce/services/DDCEService
```

3. Check DNS Resolution. Verify that the product instance can ping `tools.cisco.com`, or the nslookup translated IP.

   ```
   Device# ping tools.cisco.com
   Type escape sequence to abort.
   Sending 5, 100-byte ICMP Echos to 173.37.145.8, timeout is 2 seconds:
   !!!!!
   Success rate is 100 percent (5/5), round-trip min/avg/max = 41/41/42 ms
   ```

   If the above does not work check the following: if the product instance is set, if the product instance IP network is up. To ensure that the network is up, configure the **no shutdown** command in interface configuration mode.

   Check if the device is subnet masked with a subnet IP, and if the DNS IP is confgured.

4. Verify that the HTTPs client source interface is correct.

   Use the **show ip http client** command in privileged EXEC mode to display current configuration. Use **ip http client source-interface** command in global configuration mode to reconfigure it.

   In case the above does not work, double-check your routing rules, and firewall settings.

- If CSLU is not reachable:

1. Check if CSLU discovery works.

   - Zero-touch DNS discovery of `cslu-local` or DNS discovery of your domain..

     In the **show license all** command output, check if the `Last ACK received:` field. If this has a recent timestamp it means that the product instance has connectivity with CSLU. If it is not, proceed with the following checks:

     Check if the product instance is able to ping `cslu-local`. A successful ping confirms that the product instance is reachable.

     If the above does not work, configure the name server with an entry where hostname `cslu-local` is mapped to the CSLU IP address (the windows host where you installed CSLU). Configure the **ip domain name** *domain-name* and **ip name-server** *server-address* commands in global configuration mode. Here the CSLU IP is 192.168.0.1 and name-server creates entry `cslu-local.example.com`:

     ```
     Device(config)# ip domain name example.com
     Device(config)# ip name-server 192.168.0.1
     ```

   - CSLU URL is configured.

     In the **show license all** command output, under the `Transport:` header check the following: The `Type:` must be `cslu`and `Cslu address:` must have the hostname or the IP address of the windows host where you have installed CSLU. Check if the rest of the address is configured as shown below and check if the port number is 8182.

     ```
     Transport:
       Type: cslu
       Cslu address: http://192.168.0.1:8182/cslu/v1/pi
     ```

     If it is not, configure the **license smart transport cslu** and **license smart url cslu** *http://<cslu_ip_or_host>:8182/cslu/v1/pi* commands in global configuration mode

2. For CSLU-initiated communication, in addition to the CSLU discovery checks listed above, check the following:

   Verify HTTP connectivity. Use the **show ip http server session-module** command in privileged EXEC mode. In the output, under header `HTTP server current connections:`, check that `SL_HTTP` is active. If it is not re-configure the **ip http** commands as mentioned in Ensuring Network Reachability for CSLU-Initiated Communication, on page 167

   From a Web browser on the device where CSLU is installed, verify `https://<product-instance-ip>/`. This ensures that the REST API from CSLU to the product instance works as expected.

- If SSM On-Prem is not reachable:

1. For product instance-initiated communication, check if the SSM On-Prem transport type and URL are configured correctly.

   In the **show license all** command output, under the `Transport:` header check the following: The `Type:` must be `cslu`and `Cslu address:` must have the hostname or the IP address of the server where you have installed SSM On-Prem and <tenantID> of the *default* local virtual account. See the example below:

   ```
   Transport:
     Type: cslu
     Cslu address: https://192.168.0.1/cslu/v1/pi/on-prem-default
   ```

   Check if you have the correct URL from SSM On-Prem (See Retrieving the Transport URL (SSM On-Prem UI), on page 183) and then configure **license smart transport cslu** and **license smart url cslu** *http://<ip>/cslu/v1/pi/<tenant ID>* commands in global configuration mode.

   Check that you have configured any other required commands for your network, as mentioned in Ensuring Network Reachability for Product Instance-Initiated Communication, on page 180

2. For SSM On-Prem-initiated communication, check HTTPs connectivity.

   Use the **show ip http server session-module** command in privileged EXEC mode. In the output, under header `HTTP server current connections:`, check that `SL_HTTP` is active. If it is not re-configure the **ip http** commands as mentioned in Ensuring Network Reachability for SSM On-Prem-Initiated Communication, on page 186.

3. Check trustpoint and that certificates are accepted.

   For both forms of communication in an SSM On-Prem Deployment, ensure that the correct trustpoint is used and that the necessary certificates are accepted:

   ```
   Device(config)# crypto pki trustpoint SLA-TrustPoint
   Device(ca-trustpoint)#
   Device(ca-trustpoint)# enrollment terminal
   Device(ca-trustpoint)# revocation-check none
   Device(ca-trustpoint)# end
   Device# copy running-config startup-config
   ```

If the above does not work and the communication failure persists, contact your Cisco technical support representative.


```
Error Message  %SMART_LIC-3-COMM_RESTORED: Communications with the [chars] restored.
[chars] - depends on the transport type
        - Cisco Smart Software Manager (CSSM)
        - Cisco Smart License utility (CSLU)
Smart Agent communication with either the Cisco Smart Software Manager (CSSM) or the Cisco
 Smart License
utility (CSLU) has been restored. No action required.
```

Explanation: Product instance communication with either the CSSM, CSLU, or SSM On-Prem is restored.

Recommended Action: No action required.

```
Error Message %SMART_LIC-3-POLICY_REMOVED: The licensing policy has been removed.
```

Explanation: A previously installed *custom* licensing policy has been removed. The `Cisco default` policy is then automatically effective. This may cause a change in the behavior of smart licensing.

Possible reasons for failure include:

If you have entered the **license smart factory reset** command in privileged EXEC mode all licensing information including the policy is removed.

Recommended Action:

If the policy was removed intentionally, then no further action is required.

If the policy was removed inadvertantly, you can reapply the policy. Depending on the topology you have implemented, follow the corresponding method to retrieve the policy:

- Connected Directly to CSSM:

  Enter **show license status**, and check field `Trust Code Installed:`. If trust is established, then CSSM will automatically return the policy again. The policy is automatically re-installed on product instances of the corresponding Virtual Account.

  If trust has not been established, complete these tasks: Generating a New Token for a Trust Code from CSSM, on page 207 and Establishing Trust with an ID Token., on page 208. When you have completed these tasks, CSSM will automatically return the policy again. The policy is then automatically installed on all product instances of that Virtual Account.

- Connected to CSSM Through CSLU:

  - For product instance-initiatied communication), enter the **license smart sync** command in privileged EXEC mode. The synchronization request causes CSLU to push the missing information (a policy or authurization code) to the product instance.

  - For CSLU-initiated communication, complete this task: Collecting Usage Reports: CSLU Initiated (CSLU Interface), on page 165. This causes CSLU to detect and re-furnish the missing policy in an ACK response.

- CSLU Disconnected from CSSM:

  - For product instance-initiatied communication), enter the **license smart sync** command in privileged EXEC mode. The synchronization request causes CSLU to push the missing information (a policy or authurization code) to the product instance. Then complete these tasks in the given order:Export

to CSSM (CSLU Interface), on page 166 > Uploading Data or Requests to CSSM and Downloading a File, on page 210 > Import from CSSM (CSLU Interface), on page 166.

- For CSLU-initiated communication, complete this task: Collecting Usage Reports: CSLU Initiated (CSLU Interface), on page 165. This causes CSLU to detect and re-furnish the missing policy in an ACK response. Then complete these tasks in the given order: Export to CSSM (CSLU Interface), on page 166 > Uploading Data or Requests to CSSM and Downloading a File, on page 210 > Import from CSSM (CSLU Interface), on page 166.

- No Connectivity to CSSM and No CSLU

  If you are in an entirely air-gapped network, from a workstation that has connectivity to the internet and CSSM complete these tasks: Downloading a Policy File from CSSM, on page 209 and Installing a File on the Product Instance, on page 211

- SSM On-Prem Deployment

  - For product instance-initiatied communication), enter the **license smart sync** command in privileged EXEC mode. The causes the product instance to synchronize with SSM On-Prem and restore any required or missing information. Then synchronize SSM On-Prem with CSSM if required:

  - For SSM On-Prem-initiated communication: In the SSM On-Prem UI, navigate to **Reports** > **Synchronisation pull schedule with the devices** > **Synchronise now with the device**.

  For both forms of communication in an SSM On-Prem Deployment, synchronize with CSSM using either option:

  - SSM On-Prem is connected to CSSM: In the SSM On-Prem UI, Smart Licensing workspace, nagivate to **Reports** > **Usage Schedules** > **Synchronize now with Cisco**.

  - SSM On-Prem is not connected to CSSM: See Exporting and Importing Usage Data (SSM On-Prem UI), on page 183.

```
Error Message %SMART_LIC-3-TRUST_CODE_INSTALL_FAILED: The install of a new licensing
trust code has failed on [chars]: [chars].
```

Explanation: Trust code installation has failed. The first [chars] is the UDI where trust code installation was attempted. The second [chars] is the error string with details of the failure.

Possible reasons for failure include:

- A trust code is already installed: Trust codes are node-locked to the UDI of the product instance. If the UDI is already registered, and you try to install another one, installation fails.

- Smart Account-Virtual Account mismatch: This means the Smart Account or Virtual Account (for which the token ID was generated) does not include the product instance on which you installed the trust code. The token generated in CSSM, applies at the Smart Account or Virtual Account level and applies only to all product instances in that account.

- A signature mismatch: This means that the system clock is not accurate.

- Timestamp mismatch: This means the product instance time is not synchronized with CSSM, and can cause installation to fail.

Recommended Action:

- A trust code is already installed: If you want to install a trust code inspite of an existing trust code on the product instance, re-configure the **license smart trust idtoken** *id_token_value* {**local** | **all**} [**force**] command in privileged EXEC mode, and be sure to include the **force** keyword this time. Entering the **force** keyword sets a force flag in the message sent to CSSM to create a new trust code even if one already exists.
- Smart Account-Virtual Account mismatch:

  Log in to the CSSM Web UI at https://software.cisco.com. Under **Smart Software Licensing**, click the **Manage licenses**link. Click the **Inventory** tab. From the **Virtual Account** drop-down list, choose the required virtual account. Click the **Product Instances** tab.

  Check if the product instance on which you want to generate the token is listed in the selected Virtual Account. If it is, proceed to the next step: Generating a New Token for a Trust Code from CSSM, on page 207 and Establishing Trust with an ID Token., on page 208. If not, check and select the correct Smart Account and Virtual Account. Then complete the next steps.
- Timestamp mismatch and signature mismatch: Configure the **ntp server** command in global configuration mode. For example:

  ```
  Device(config)# ntp server 198.51.100.100 version 2 prefer
  ```

```
Error Message    %SMART_LIC-4-REPORTING_NOT_SUPPORTED: The CSSM OnPrem that this
product instance is connected to is down rev and does not support the enhanced policy and
usage
reporting mode.
```

Explanation: Cisco Smart Software Manager On-Prem (formerly known as Cisco Smart Software Manager satellite) is supported in the Smart Licensing Using Policy environment starting with Cisco IOS XE Amsterdam 17.3.3 only (See SSM On-Prem, on page 99). In *unsupported* releases, the product instance will behave as follows:

- Stop sending registration renewals and authorization renewals.
- Start recording usage and saving RUM reports locally.

Recommended Action:

You have the following options:

- Refer to and implement one of the supported topologies instead. See: Supported Topologies, on page 105.
- Upgrade to a release where SSM On-Prem is supported with Smart Licensing Using Policy. See Migrating to a Version of SSM On-Prem That Supports Smart Licensing Using Policy, on page 160.

```
Error Message %SMART_LIC-6-POLICY_INSTALL_SUCCESS: A new licensing policy
was successfully installed.
```

Explanation: A policy was installed in one of the following ways:

- Using Cisco IOS commands.
- CSLU-initiated communication.
- As part of an ACK response.

Recommended Action: No action is required. If you want to know which policy is applied (the policy in-use) and its reporting requirements, enter the **show license all** command in privileged EXEC mode.

```
Error Message %SMART_LIC-6-AUTHORIZATION_INSTALL_SUCCESS: A new licensing
authorization code was successfully installed on: [chars].
```

Explanation: [chars] is the UDI where the authorization code was installed successfully.

Recommended Action: No action is required. If you want to know the details of the authorization code that was installed, enter the **show license authorization** command in privileged EXEC mode.

```
Error Message %SMART_LIC-6-AUTHORIZATION_REMOVED: A licensing authorization code has
 been removed from [chars]
```

Explanation: [chars] is the UDI where the authorization code was installed. The authorization code has been removed. This removes the licenses from the product instance and may cause a change in the behavior of smart licensing and the features using licenses.

Recommended Action: No action is required. If you want to see the current state of the license, enter the **show license all** command in privileged EXEC mode.

```
Error Message %SMART_LIC-6-REPORTING_REQUIRED: A Usage report acknowledgement
will be required in [dec] days.
```

Explanation: This is an alert which means that RUM reporting to Cisco is required. [dec] is the amount of time (in days) left to meet this reporting requirements.

Recommended Action: Ensure that RUM reports are sent within the requested time. The topology you have implemented determines the reporting method.

- Connected to CSSM Through CSLU
  - For product instance-initiatied communication: Enter the **license smart sync** command in privileged EXEC mode. If CSLU is currently logged into CSSM the reports will be automatically sent to the associated Smart Account and Virtual Account in CSSM.
  - For CSLU-initiated communication, complete this task: Collecting Usage Reports: CSLU Initiated (CSLU Interface), on page 165.
- Connected Directly to CSSM: Enter the **license smart sync** command in privileged EXEC mode.

- Connected to CSSM Through a Controller: If the product instance is managed by a controller, the controller will send the RUM report at the scheduled time.

  If you are using Cisco DNA Center as the controller, you have the option of ad-hoc reporting. See the *Cisco DNA Center Administrator Guide* of the required release (Release 2.2.2 onwards) > *Manage Licenses* > *Upload Resource Utilization Details to CSSM*.

- CSLU Disconnected from CSSM: If the product instance is connected to CSLU, synchronize with the product instance as shown for "Connected to CSSM Through CSLU"above, then complete these tasks: Export to CSSM (CSLU Interface), on page 166, Uploading Data or Requests to CSSM and Downloading a File, on page 210, and Import from CSSM (CSLU Interface), on page 166.

- No Connectivity to CSSM and No CSLU: Enter the **license smart save usage** command in privileged EXEC mode, to save the required usage information in a file. Then, from a workstation where you have connectivity to CSSM, complete these tasks: Uploading Data or Requests to CSSM and Downloading a File, on page 210 > Installing a File on the Product Instance, on page 211.

- SSM On-Prem Deployment:

  Synchronize the product instance with SSM On-Prem:

  - For product instance-initiatied communication: Enter the **license smart sync** command in privileged EXEC mode. If CSLU is currently logged into CSSM the reports will be automatically sent to the associated Smart Account and Virtual Account in CSSM.
  - For SSM On-Prem-initiated communication, complete this task: In the SSM On-Prem UI, navigate to **Reports** > **Synchronisation pull schedule with the devices** > **Synchronise now with the device**.

  Synchronize usage information with CSSM (*choose one*)

  - SSM On-Prem is connected to CSSM: In the SSM On-Prem UI, Smart Licensing workspace, nagivate to **Reports** > **Usage Schedules** > **Synchronize now with Cisco**.
  - SSM On-Prem is not connected to CSSM: See Exporting and Importing Usage Data (SSM On-Prem UI), on page 183.

```
Error Message %SMART_LIC-6-TRUST_CODE_INSTALL_SUCCESS: A new licensing trust code
 was successfully installed on [chars].
```

Explanation:[chars] is the UDI where the trust code was successfully installed.

Recommended Action: No action is required. If you want to verify that the trust code is installed, enter the **show license status** command in privileged EXEC mode. Look for the updated timestamp under header `Trust Code Installed:` in the output.

Additional References for Smart Licensing Using Policy

Topic	Document Title
For complete syntax and usage information for the commands used in this chapter, see *System Mangement* > *System Mangement Commands* in the Command Reference of the required release.	Command Reference (Catalyst 9600 Series Switches)
Cisco Smart Software Manager Help	Smart Software Manager Help
Cisco Smart License Utility (CSLU) installation and user guides	Cisco Smart Licensing Utility Quick Start Setup Guide Cisco Smart Licensing Utility User Guide
General information about Smart Licensing	Smart Software Licensing
Troubleshooting TechNotes	Smart Licensing using Policy on Catalyst Switching Platforms Migrate Catalyst License to Smart Licensing Using Policy
Cisco DNA for Switching	Cisco DNA Software Subscription Matrix for Switching

Feature History for Smart Licensing Using Policy

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.11.1	Smart Licensing	A cloud-based, software license management solution that allows you to manage and track the status of your license, hardware, and software usage trends. Smart Licensing is the default and the only available method to manage licenses.

Release	Feature	Feature Information
Cisco IOS XE Amsterdam 17.3.2a	Smart Licensing Using Policy	An enhanced version of Smart Licensing, with the overarching objective of providing a licensing solution that does not interrupt the operations of your network, rather, one that enables a compliance relationship to account for the hardware and software licenses you purchase and use. Starting with this release, Smart Licensing Using Policy is automatically enabled on the device. This is also the case when you upgrade to this release. By default, your Smart Account and Virtual Account in CSSM is enabled for Smart Licensing Using Policy.
	Cisco DNA Center support for Smart Licensing Using Policy	Cisco DNA Center supports Smart Licensing Using Policy functionality starting with Cisco DNA Center Release 2.2.2. When you use Cisco DNA Center to manage a product instance, Cisco DNA Center connects to CSSM, and is the interface for all communication to and from CSSM. For information about the comptabile controller and product instance versions, see Controller, on page 98. For information about this topology, see Connected to CSSM Through a Controller, on page 109 and Workflow for Topology: Connected to CSSM Through a Controller, on page 127.
Cisco IOS XE Amsterdam 17.3.3	Smart Software Manager On-Prem (SSM On-Prem) Support for Smart Licensing Using Policy	SSM On-Prem is an asset manager, which works in conjunction with CSSM. It enables you to administer products and licenses on your premises instead of having to directly connect to CSSM. For information about the comptabile SSM On-Prem and product instance versions, see: SSM On-Prem, on page 99. For an overview of this topology, and to know how to implement it, see SSM On-Prem Deployment, on page 113 and Workflow for Topology: SSM On-Prem Deployment, on page 132. For information about migrating from an exisiting version of SSM On-Prem, to one that supports Smart Licensing Using Policy, see Migrating to a Version of SSM On-Prem That Supports Smart Licensing Using Policy, on page 160.

Release	Feature	Feature Information
Cisco IOS XE Bengaluru 17.6.2	Export Control Key for High Security (HSECK9 key)	The HSECK9 key was introduced on the Cisco Catalyst 9300X Series Switches. The HSECK9 key is an export-controlled license, which authorizes the use of cryptographic features that are restricted by U.S. export control laws. If you want to use a restricted cryptographic feature, an HSECK9 key is required. See Authorization Code, on page 101. On product instances where the HSECK9 key is supported, you can obtain and install SLAC by implementing one of these topologies: • Workflow for Topology: Connected to CSSM Through CSLU, on page 123 • Workflow for Topology: Connected Directly to CSSM, on page 125 • Workflow for Topology: CSLU Disconnected from CSSM, on page 128 • Workflow for Topology: No Connectivity to CSSM and No CSLU, on page 131 • Workflow for Topology: SSM On-Prem Deployment, on page 132

Release	Feature	Feature Information
Cisco IOS XE Cupertino 17.7.1	CSLU support for Linux	Support for CSLU deployment on a machine (laptop or desktop) running Linux. See CSLU, on page 97, Workflow for Topology: Connected to CSSM Through CSLU, on page 123 and Workflow for Topology: CSLU Disconnected from CSSM, on page 128.
	Factory-installed trust code	For new hardware orders, Cisco installs a trust code at the time of manufacturing. See: Overview, on page 96 and Trust Code, on page 104.
	Trust code request and installation in additional topologies	A trust code is automatically obtained in topologies where the product instance initiates the sending of data to *CSLU* and in topologies where the product instance is in an air-gapped network. See: • Trust Code, on page 104 • Connected to CSSM Through CSLU, on page 105 and Tasks for Product Instance-Initiated Communication, on page 123 • CSLU Disconnected from CSSM, on page 110 and Tasks for Product Instance-Initiated Communication, on page 128 • No Connectivity to CSSM and No CSLU, on page 112 and Workflow for Topology: No Connectivity to CSSM and No CSLU, on page 131 • In the command reference of the corresponding release, see the **license smart** privileged EXEC command.
	Ability to save SLAC request and return in a file in an air-gapped network	

Release	Feature	Feature Information
		Option to save a SLAC request file on the product instance. The SLAC request file must be uploaded to CSSM and the file containing the SLAC code can then be downloaded and installed it on the product instance - the same as a RUM report and ACK. With this method you do not have to gather and enter the required details on the CSSM Web UI to generate a SLAC Similarly, an authorization code that is saved to a file can also be uploaded the same way as a RUM report. See: No Connectivity to CSSM and No CSLU, on page 112 and Workflow for Topology: No Connectivity to CSSM and No CSLU, on page 131. In the command reference of the corresponding release, see the **license smart** privileged EXEC command.
	Support to collect software version in a RUM report	If version privacy is disabled (**no license smart privacy version** global configuration command), the Cisco IOS-XE software version running on the product instance and the Smart Agent version information is *included* in the RUM report. In the command reference of the corresponding release, see the **license smart** global configuration command.
	RUM Report optimization and availability of statistics	RUM report generation and related processes have been optimized. This includes a reduction in the time it takes to process RUM reports, better memory and disk space utilization, and visibility into the RUM reports on the product instance (how many there are, the processing state each one is in, if there are errors in any of them, and so on). See RUM Report and Report Acknowledgement, on page 103, Upgrades Within the Smart Licensing Using Policy Environment, on page 121, and Downgrades Within the Smart Licensing Using Policy Environment, on page 122. In the command reference of the corresponding release, see the **show license rum**, **show license all**, and **show license tech** privileged EXEC commands.
	Account information included in **show** command outputs	

Release	Feature	Feature Information
		A RUM acknowledgement (ACK) includes the Smart Account and Virtual Account that was reported to, in CSSM. You can then display account information using various **show** commands. The account information that is displayed is always as per the latest available ACK on the product instance. In the command reference of the corresponding release, see the **show license summary**, **show license status**, **show license all**, and **show license tech** privileged EXEC commands.
Cisco IOS XE Cupertino 17.7.1	Smart Licensing Using Policy	Smart Licensing Using Policy was implemented on the following product instances: • C9500X-28C8D, which was introduced in this release. C9500X-28C8D is part of the new Cisco Catalyst 9500X Series Switches, which is still part of the overall Cisco Catalyst 9500 Series Switches. • Catalyst 9600 Series Supervisor Engine 2 (C9600X-SUP-2), which was introduced this release • Cisco Catalyst 9400 Series Supervisor Modules 2 and 2XL (C9400X-SUP-2 and C9400X-SUP-2XL), which were introduced in this release

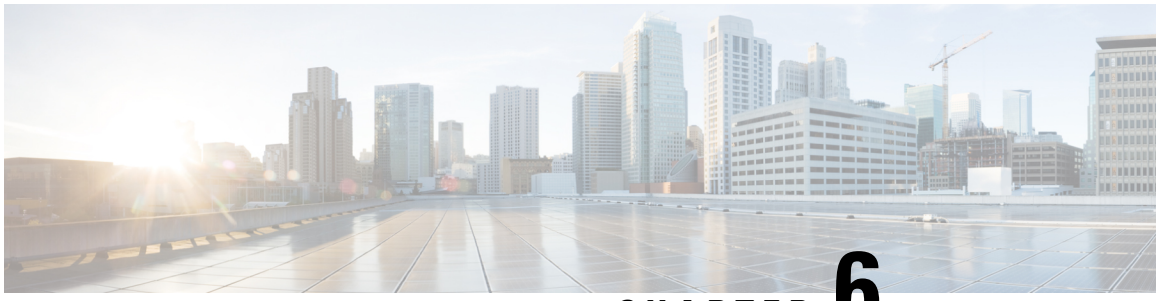
Release	Feature	Feature Information
Cisco IOS XE Cupertino 17.8.1	Export Control Key for High Security (HSECK9 key)	This feature was implemented on the following product instances: • Cisco Catalyst 9500X Series Switches • Catalyst 9600 Series Supervisor Engine 2 with associated line cards. See Authorization Code, on page 101. On product instances where the HSECK9 key is supported, you can obtain and install Smart Licensing Authorization Code (SLAC) for the HSECK9 key, by implementing one of these topologies: • Workflow for Topology: Connected to CSSM Through CSLU, on page 123 • Workflow for Topology: Connected Directly to CSSM, on page 125 • Workflow for Topology: CSLU Disconnected from CSSM, on page 128 • Workflow for Topology: No Connectivity to CSSM and No CSLU, on page 131 • Workflow for Topology: SSM On-Prem Deployment, on page 132

Release	Feature	Feature Information
Cisco IOS XE Cupertino 17.9.1	New mechanism to send data privacy related information	A new mechanism to send all data privacy related information was introduced. This information is no longer included in a RUM report. If data privacy is disabled (**no license smart privacy** {**all** | **hostname** | **version**}} global configuration command), data privacy related information is sent in a separate sync message or offline file. Depending on the topology you have implemented, the product instance initiates the sending of this information in a separate message, or CSLU and SSM On-Prem initiates the retrieval of this information from the product instance, or this information is saved in the offline file that is generated when you enter the license smart save usage privileged EXEC command In the command reference of the corresponding release, see the **license smart** global configuration command.
	Hostname support	If you configure a hostname on the product instance and disable the corresponding privacy setting (**no license smart privacy hostname** global configuration command), hostname information is sent from the product instance. Depending on the topology you have implemented, the hostname information is received by CSSM, and CSLU or SSM On-Prem. It is then displayed on the corresponding user interface. In the command reference of the corresponding release, see the **license smart** global configuration command.
	Trust code request and installation	From this release, trust code request and installation is supported in the CSLU-initiated mode as well. See Trust Code, on page 104, Workflow for Topology: Connected to CSSM Through CSLU, on page 123, and Workflow for Topology: CSLU Disconnected from CSSM, on page 128.
	RUM Report Throttling	

Release	Feature	Feature Information
		For all topologies where the product instance initiates communication, the minimum reporting frequency is throttled to one day. This means the product instance does not send more than one RUM report a day. The affected topologies are: *Connected Directly to CSSM*, *Connected to CSSM Through CSLU* (product instance-initiated communication), *CSLU Disconnected from CSSM* (product instance-initiated communication), and *SSM On-Prem Deployment* (product instance-initiated communication). You can override the reporting frequency throttling, by entering the **license smart sync** command in privileged EXEC mode. This triggers an on-demand synchronization with CSSM or CSLU, or SSM On-Prem, to send and receive any pending data. RUM report throttling also applies to the Cisco IOS XE Amsterdam 17.3.6 and later releases of the 17.3.x train, and Cisco IOS XE Bengaluru 17.6.4 and later releases of the 17.6.x train. From Cisco IOS XE Cupertino 17.9.1, RUM report throttling is applicable to *all* subsequent releases. See: Connected to CSSM Through CSLU, on page 105, Connected Directly to CSSM, on page 107, CSLU Disconnected from CSSM, on page 110, and SSM On-Prem Deployment, on page 113.
	Smart Licensing Using Policy	This feature was implemented on C9200CX-12P-2X2G, C9200CX-8P-2X2G, and C9200CX-12T-2X2G models of the Cisco Catalyst 9200CX Series Switches.

Release	Feature	Feature Information
Cisco IOS XE Dublin 17.11.1	Export Control Key for High Security (HSECK9 key)	This feature was implemented on Cisco Catalyst 9400 Series Supervisor 2 and 2XL Modules (C9400X-SUP-2 and C9400X-SUP-2XL). See Authorization Code, on page 101. On product instances where the HSECK9 key is supported, you can obtain and install Smart Licensing Authorization Code (SLAC) for the HSECK9 key, by implementing one of these topologies: • Workflow for Topology: Connected to CSSM Through CSLU, on page 123 • Workflow for Topology: Connected Directly to CSSM, on page 125 • Workflow for Topology: CSLU Disconnected from CSSM, on page 128 • Workflow for Topology: No Connectivity to CSSM and No CSLU, on page 131 • Workflow for Topology: SSM On-Prem Deployment, on page 132

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn



CHAPTER 6

Environmental Monitoring and Power Management

About Environmental Monitoring

Environmental monitoring of chassis components provides early warning indications of possible component failure. This warning helps you to ensure the safe and reliable operation of your system and avoid network interruptions.

This section describes how to monitor critical system components so that you can identify and rapidly correct hardware-related problems.

Using CLI Commands to Monitor your Environment

Enter the **show environment** [**all** | **counters** | **history** | **location** | **sensor** | **status** | **summary** | **table**] command to display system status information. Keyword descriptions are listed in the following table.

Table 16: Keyword Descriptions

Keyword	Purpose
all	Displays a detailed listing of all the environmental monitor parameters (for example, the power supplies, temperature readings, voltage readings, and so on). This is the default.
counters	Displays operational counters.
history	Displays the sensor state change history.
location	Displays sensors by location.
sensor	Displays the sensor summary.

Keyword	Purpose
status	Displays field-replaceable unit (FRU) operational status and power and power supply fan sensor information.
summary	Displays the summary of all the environment monitoring sensors.
table	Displays a sensor state table.

Displaying Environment Conditions

Supervisor modules and their associated line cards support multiple temperature sensors per card. The environment condition output includes the temperature reading from each sensor and the temperature thresholds for each sensor. These line cards support three thresholds: warning, critical, and shutdown.

The following example illustrates how to display the environment condition on a supervisor module. The thresholds appear within parentheses.

```
Device# show environment

Number of Critical alarms:  0
Number of Major alarms:     0
Number of Minor alarms:     0

 Slot        Sensor          Current State   Reading
Threshold(Minor,Major,Critical,Shutdown)
 ----------  --------------  --------------- ------------
-------------------------------------
 R0          Temp: InltFrnt  Normal          27   Celsius (45 ,50 ,55 ,60 )(Celsius)
 R0          Temp: InltRear  Normal          28   Celsius (45 ,50 ,55 ,60 )(Celsius)
 R0          Temp: OtltFrnt  Normal          35   Celsius (75 ,80 ,85 ,90 )(Celsius)
 R0          Temp: OtltRear  Normal          43   Celsius (75 ,80 ,85 ,90 )(Celsius)
 R0          Temp: UADP_0_0  Normal          54   Celsius (105,110,120,124)(Celsius)
 R0          Temp: UADP_0_1  Normal          53   Celsius (105,110,120,124)(Celsius)
 R0          Temp: UADP_0_2  Normal          53   Celsius (105,110,120,124)(Celsius)
 R0          Temp: UADP_0_3  Normal          55   Celsius (105,110,120,124)(Celsius)
 R0          Temp: UADP_0_4  Normal          54   Celsius (105,110,120,124)(Celsius)
 R0          Temp: UADP_0_5  Normal          55   Celsius (105,110,120,124)(Celsius)
 R0          Temp: UADP_0_6  Normal          64   Celsius (105,110,120,124)(Celsius)
 R0          Temp: UADP_0_7  Normal          59   Celsius (105,110,120,124)(Celsius)
 R0          Temp: UADP_0_8  Normal          55   Celsius (105,110,120,124)(Celsius)
 <output truncated>
```

The following example illustrates how to display the LED status on a supervisor module.

```
Device# show hardware led

SWITCH: 1
SYSTEM: GREEN

Line Card : 1
PORT STATUS: (48) Fo1/0/1:BLACK Fo1/0/2:BLACK Fo1/0/3:BLACK Fo1/0/4:BLACK Fo1/0/5:BLACK
Fo1/0/6:BLACK Fo1/0/7:BLACK Fo1/0/8:BLACK Fo1/0/9:BLACK Fo1/0/10:BLACK Fo1/0/11:BLACK
Fo1/0/12:BLACK Fo1/0/13:BLACK Fo1/0/14:BLACK Fo1/0/15:BLACK Fo1/0/16:BLACK Fo1/0/17:BLACK
Fo1/0/18:BLACK Fo1/0/19:BLACK Fo1/0/20:BLACK Fo1/0/21:GREEN Fo1/0/22:BLACK Fo1/0/23:BLACK
Fo1/0/24:BLACK Hu1/0/25:GREEN Hu1/0/26:BLACK Hu1/0/27:BLACK Hu1/0/28:BLACK Hu1/0/29:BLACK
Hu1/0/30:BLACK Hu1/0/31:BLACK Hu1/0/32:BLACK Hu1/0/33:BLACK Hu1/0/34:BLACK Hu1/0/35:BLACK
Hu1/0/36:BLACK Hu1/0/37:BLACK Hu1/0/38:BLACK Hu1/0/39:BLACK Hu1/0/40:BLACK Hu1/0/41:BLACK
```

```
Hu1/0/42:BLACK Hu1/0/43:BLACK Hu1/0/44:BLACK Hu1/0/45:BLACK Hu1/0/46:BLACK Hu1/0/47:BLACK
Hu1/0/48:BLACK
BEACON: BLACK
STATUS: GREEN

Line Card : 2
PORT STATUS: (48) Fo2/0/1:BLACK Fo2/0/2:GREEN Fo2/0/3:GREEN Fo2/0/4:GREEN Fo2/0/5:GREEN
Fo2/0/6:GREEN Fo2/0/7:GREEN Fo2/0/8:GREEN Fo2/0/9:GREEN Fo2/0/10:GREEN Fo2/0/11:GREEN
Fo2/0/12:GREEN Fo2/0/13:GREEN Fo2/0/14:GREEN Fo2/0/15:GREEN Fo2/0/16:GREEN Fo2/0/17:GREEN
Fo2/0/18:GREEN Fo2/0/19:GREEN Fo2/0/20:GREEN Fo2/0/21:GREEN Fo2/0/22:GREEN Fo2/0/23:GREEN
Fo2/0/24:BLACK Hu2/0/25:BLACK Hu2/0/26:BLACK Hu2/0/27:BLACK Hu2/0/28:BLACK Hu2/0/29:BLACK
Hu2/0/30:BLACK Hu2/0/31:BLACK Hu2/0/32:BLACK Hu2/0/33:BLACK Hu2/0/34:BLACK Hu2/0/35:BLACK
Hu2/0/36:BLACK Hu2/0/37:BLACK Hu2/0/38:BLACK Hu2/0/39:BLACK Hu2/0/40:BLACK Hu2/0/41:BLACK
Hu2/0/42:BLACK Hu2/0/43:BLACK Hu2/0/44:BLACK Hu2/0/45:BLACK Hu2/0/46:BLACK Hu2/0/47:BLACK
Hu2/0/48:BLACK
BEACON: BLACK
STATUS: GREEN

MODULE: slot 3
SUPERVISOR: ACTIVE
PORT STATUS: (0)
BEACON: BLACK
STATUS: GREEN
SYSTEM: GREEN
ACTIVE: GREEN

MODULE: slot 4
SUPERVISOR: STANDBY
PORT STATUS: (0)
BEACON: BLACK
STATUS: GREEN
SYSTEM: GREEN
ACTIVE: AMBER

Line Card : 5
PORT STATUS: (48) Twe5/0/1:BLACK Twe5/0/2:GREEN Twe5/0/3:GREEN Twe5/0/4:GREEN Twe5/0/5:GREEN
 Twe5/0/6:GREEN Twe5/0/7:GREEN Twe5/0/8:GREEN Twe5/0/9:GREEN Twe5/0/10:GREEN Twe5/0/11:GREEN
 Twe5/0/12:GREEN Twe5/0/13:GREEN Twe5/0/14:GREEN Twe5/0/15:GREEN Twe5/0/16:GREEN
Twe5/0/17:GREEN Twe5/0/18:GREEN Twe5/0/19:GREEN Twe5/0/20:GREEN Twe5/0/21:GREEN
Twe5/0/22:GREEN Twe5/0/23:GREEN Twe5/0/24:GREEN Twe5/0/25:GREEN Twe5/0/26:GREEN
Twe5/0/27:GREEN Twe5/0/28:GREEN Twe5/0/29:GREEN Twe5/0/30:GREEN Twe5/0/31:GREEN
Twe5/0/32:GREEN Twe5/0/33:GREEN Twe5/0/34:GREEN Twe5/0/35:GREEN Twe5/0/36:GREEN
Twe5/0/37:GREEN Twe5/0/38:GREEN Twe5/0/39:GREEN Twe5/0/40:GREEN Twe5/0/41:GREEN
Twe5/0/42:GREEN Twe5/0/43:GREEN Twe5/0/44:GREEN Twe5/0/45:GREEN Twe5/0/46:GREEN
Twe5/0/47:BLACK Twe5/0/48:BLACK
BEACON: BLACK
STATUS: GREEN

Line Card : 6
PORT STATUS: (48) Twe6/0/1:BLACK Twe6/0/2:GREEN Twe6/0/3:GREEN Twe6/0/4:GREEN Twe6/0/5:GREEN
 Twe6/0/6:GREEN Twe6/0/7:GREEN Twe6/0/8:GREEN Twe6/0/9:GREEN Twe6/0/10:GREEN Twe6/0/11:GREEN
 Twe6/0/12:GREEN Twe6/0/13:GREEN Twe6/0/14:GREEN Twe6/0/15:GREEN Twe6/0/16:GREEN
Twe6/0/17:GREEN Twe6/0/18:GREEN Twe6/0/19:GREEN Twe6/0/20:GREEN Twe6/0/21:GREEN
Twe6/0/22:GREEN Twe6/0/23:GREEN Twe6/0/24:GREEN Twe6/0/25:GREEN Twe6/0/26:GREEN
Twe6/0/27:GREEN Twe6/0/28:GREEN Twe6/0/29:GREEN Twe6/0/30:GREEN Twe6/0/31:GREEN
Twe6/0/32:GREEN Twe6/0/33:GREEN Twe6/0/34:GREEN Twe6/0/35:GREEN Twe6/0/36:BLACK
Twe6/0/37:BLACK Twe6/0/38:BLACK Twe6/0/39:BLACK Twe6/0/40:GREEN Twe6/0/41:GREEN
Twe6/0/42:GREEN Twe6/0/43:GREEN Twe6/0/44:GREEN Twe6/0/45:GREEN Twe6/0/46:BLACK
Twe6/0/47:BLACK Twe6/0/48:BLACK
BEACON: BLACK
STATUS: GREEN

RJ45 CONSOLE: GREEN
```

```
GigabitEthernet0/0 (MGMT): GREEN

TenGigabitEthernet0/1 (SFP MGMT): BLACK
FANTRAY STATUS: GREEN
FANTRAY BEACON: BLACK
```

Displaying On Board Failure Logging (OBFL) information

The OBFL feature records operating temperatures, hardware uptime, interrupts, and other important events and messages that can assist with diagnosing problems with line cards and supervisor modules installed in a switch. Data is logged to files stored in nonvolatile memory. When the onboard hardware is started up, a first record is made for each area monitored and becomes a base value for subsequent records. The OBFL feature provides a circular updating scheme for collecting continuous records and archiving older (historical) records, ensuring accurate data about the system. Data is recorded in one of two formats: continuous information that displays a snapshot of measurements and samples in a continuous file, and summary information that provides details about the data being collected. The data is displayed using the **show logging onboard** command. The message "No historical data to display" is seen when historical data is not available.

```
Device# show logging onboard RP active voltage detail

--------------------------------------------------------------------------------
VOLTAGE SUMMARY INFORMATION
--------------------------------------------------------------------------------
Number of sensors          : 33
--------------------------------------------------------------------------------
Sensor                     ID         Normal Range          Maximum Sensor Value
--------------------------------------------------------------------------------
CPU_P5V                    0          0 - 5                 5
CPU_P3V3                   1          0 - 5                 3
CPU_P2V5_VPP               2          0 - 5                 2
CPU_PVCCSCFUSESUS          3          0 - 5                 1
CPU_PVCCIN                 4          0 - 5                 1
CPU_P1V5_PCH               5          0 - 5                 1
CPU_PVCCKRHV               6          0 - 5                 1
CPU_P1V2_VDDQ              7          0 - 5                 1
CPU_P1V05_COMBINED         8          0 - 5                 1
CPU_P0V6_VTT               9          0 - 5                 1
BB_P1V0_BCM82752           10         0 - 5                 3
BB_P3V3_A                  11         0 - 5                 12
BB_P12V0                   12         0 - 12                12
BB_P7V0                    13         0 - 7                 7
BB_P5V0                    14         0 - 5                 5
BB_P1V5                    15         0 - 5                 3
BB_P3V3                    16         0 - 5                 3
BB_P2V5                    17         0 - 5                 2
BB_P1V8                    18         0 - 5                 1
BB_P0V9_DP0_PLL            19         0 - 5                 0
BB_P0V9_DP1_PLL            20         0 - 5                 0
BB_P0V9_DP2_PLL            21         0 - 5                 0
BB_P0V8_DP0_VDD            22         0 - 5                 0
BB_P0V8_DP1_VDD            23         0 - 5                 0
BB_P0V8_DP2_VDD            24         0 - 5                 0
BB_P0V9_DP0_AVDD           25         0 - 5                 0
BB_P0V9_DP1_AVDD           26         0 - 5                 0
BB_P0V9_DP2_AVDD           27         0 - 5                 1
BB_P1V1_HATH               28         0 - 5                 1
BB_P1V1_DP0_AVDDH          29         0 - 5                 1
BB_P1V2_HATH               30         0 - 5                 3
BB_3V3_IRC                 31         0 - 5                 3
```

```
BB_P3V3_EUSB              32          0 - 5                0

-------------------------------------------------------------------------------
Sensor Value
Total Time of each Sensor
-------------------------------------------------------------------------------

value: 0
0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 61d, 94d, 577h, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 61d, 112d, 112d,
 112d, 112d, 112d, 112d, 112d, 112d, 50d, 0s, 0s, 0s, 0s, 112d,
value: 1
0s, 0s, 0s, 112d, 112d, 112d, 112d, 112d, 50d, 426h, 645h, 0s, 0s, 0s, 61d, 50d, 0s, 61d,
50d, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 61d, 112d, 112d, 50d, 0s, 0s,
value: 2
0s, 0s, 112d, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 61d, 50d, 0s, 0s, 0s, 0s,
 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s,
value: 3
0s, 112d, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 61d, 50d, 0s, 0s, 0s, 61d, 50d, 0s, 0s, 0s, 0s,
0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 61d, 112d, 0s,
value: 4
900h, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 160d, 43d, 0s, 0s, 0s, 0s, 0s, 0s,
0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s, 0s,
value: 5
<output truncated>
```

Emergency Actions

The chassis can power down a single card, providing a detailed response to over-temperature conditions on line cards. However, the chassis cannot safely operate when the temperature of the supervisor module itself exceeds the critical threshold. The supervisor module turns off the chassis' power supplies to protect itself from overheating. When this happens, you can recover the switch only by cycling the power on and off switches on the power supplies or by cycling the AC or DC inputs to the power supplies.

Shutdown temperature emergencies on a supervisor will trigger chassis shutdown. Shutdown temperature emergencies on a linecard will shut down the linecard but not the chassis. Critical temperature emergencies will trigger a warning message and the fan will be at its highest speed, but the chassis will not shut down. This applies to all slots.

The following table lists temperature emergencies but does not distinguish between critical and shutdown emergencies.

Table 17: Emergency and Action

Case 1. Complete fan failure emergency.	SYSLOG message displays and the chassis shuts down.
Case 2. Temperature emergency on a line card.	Power down the line card.
Case 3. Temperature emergency on a power supply. When the shutdown alarm threshold is exceeded, all the power supplies will shut down.	Power cycle the device to recover from power supply shut down.
Case 4. Temperature emergency on the active supervisor module.	Power down the chassis.

System Alarms

Any system has two types of alarms: major and minor. A major alarm indicates a critical problem that could lead to system shutdown. A minor alarm is informational—it alerts you to a problem that could become critical if corrective action is not taken.

The following table lists the possible environment alarms.

Table 18: Possible Environmental Alarms

A temperature sensor over its warning threshold	minor
A temperature sensor over its critical threshold	major
A temperature sensor over its shutdown threshold	major
A partial fan failure	minor
A complete fan failure **Note** A complete fan failure alarm does not result in system shutdown.	major

Fan failure alarms are issued as soon as the fan failure condition is detected and are canceled when the fan failure condition clears. Temperature alarms are issued as soon as the temperature reaches the threshold temperature. An LED on the supervisor module indicates whether an alarm has been issued.

When the system issues a major alarm, it starts a timer whose duration depends on the alarm. If the alarm is not canceled before the timer expires, the system takes emergency action to protect itself from the effects of overheating. The timer values and the emergency actions depend on the type of supervisor module.

Note Refer to the *Hardware Installation Guide* for information on LEDs, including the startup behavior of the supervisor module system LED.

Table 19: Alarms on Supervisor Module

Event	Alarm Type	Supervisor LED Color	Description and Action
Card temperature exceeds the critical threshold.	Major	Red	Syslog message displays when the alarm is issued.
Card temperature exceeds the shutdown threshold.	Major	Red	Syslog message displays when the alarm is issued.
Chassis temperature exceeds the warning threshold.	Minor	Orange	Syslog message displays when the alarm is issued.
Chassis fan tray experiences partial failure.	Minor	Orange	Syslog message displays when the alarm is issued.

Event	Alarm Type	Supervisor LED Color	Description and Action
Chassis fan tray experiences complete failure.	Major	Red	Syslog message displays when the alarm is issued.

Power Management

This section describes the power management feature in the Cisco Catalyst 9600 Series Switchesand the aspects of power management that you can control and configure. For information about the hardware, including installation, removal and power supply specifications, see the *Cisco Catalyst 9600 Series Switches Hardware Installation Guide*.

Restrictions for Power Management

- When using an AC power source for the power supply modules, you cannot mix 110V and 220V inputs.
- When using a combination of AC and DC power sources for the power supply modules, the input voltage for all the power supply modules needs to be the same. The input voltage can either be 110V or 220V for all the power supply modules. This applies to both the combined mode and n+1 redundant power supply mode.

Power Supply Modes

Cisco Catalyst 9600 Series Switches offer combined and redundant configuration modes for power supplies.

Combined Mode

This is the default power supply mode.

The system operates on one to four power supplies. All available power supplies are active and sharing power and can operate at up to 100 percent capacity.

Available power in the combined mode is the sum of the individual power supplies.

Redundant Mode

In a redundant configuration, a given power supply module can be either active, or in standby mode, and switch to active when required.

You can configure an n+1 redundant mode.

- n+1 redundant Mode—n number of power supply modules are active (n can be one to seven power supply modules). +1 is the power supply module reserved for redundancy.

  The default power supply slot is PS4.

  Specify a standby slot, by entering the **power redundancy-mode redundant n+1** *standby-PSslot* command.

Enter the **show power detail** command in priviledged EXEC mode, to display detailed information about the currently configured power supply mode.

Operating States

The operating state refers to the system's capacity to respond to a situation where all active power supply modules fail. The system deems the chassis operating state as full protected, normal protected, or combined depending on these factors:

- Total active output power, which is the total output power that is available from all the active power supply modules in the chassis.
- Required budgeted power, which is the power the system requires only for the supervisor modules, switching modules (line cards), and fan tray to operate in the chassis.

  In the **show** command outputs (**show power**, **show power detail**), this is displayed as `System Power`.
- Total standby output power, which is the total output power that is available from all the power supply modules in the chassis that are configured as standby.

Whether in the n+1, the system considers the chassis in a full protected state, when ALL of these conditions are met:

- Total active output power is greater than the required budgeted power
- Total standby output power is greater than or equal to total active output power

Whether in the n+1, the system considers the chassis in a normal protected state, when ALL of these conditions are met:

- Total active output power is greater than the required budgeted power
- Total standby output power is lesser than the total active output power

The system operates in a combined state, when it encounters these conditions (any redundancy configuration is rejected):

- Total active output power is lesser than the required budgeted power
- A standby power supply module is not configured or installed.

Information about the operating state is also displayed in the **show power** and **show power detail** command output.

Power Management Considerations

It is possible to configure a switch that requires more power than the power supplies provide.

The following list the conditions where the power requirements for the installed modules exceed the power provided by the power supplies.

- If the switch has a single power supply module that is unable to meet power requirements, the following error message is displayed:

  ```
  Insufficient power supplies present for specified configuration
  ```

  The **show power** command output will also indicate this state of insufficient input power.

- If the switch has more than one power supply module, and requirements for the installed modules still exceed the power provided by the power supplies, the following error message is displayed:

  ```
  Insufficient number of power supplies (2) are installed for power redundancy mode
  ```

  The **show power** command output will also indicate this state of insufficient input power.

If you attempt to insert additional modules into your switch and exceed the power supply, the switch immediately places the newly inserted module into reset mode, and the following error message is displayed:

```
Power doesn't meet minimum system power requirement.
```

Additionally, if you power down a functioning chassis and insert an additional linecard or change the module configuration so that the power requirements exceed the available power, one or more linecards enter reset mode when you power on the switch again.

Selecting a Power Supply Mode

Your switch hardware configuration dictates which power supply or supplies you should use. For example, if your switch configuration requires more power than a single power supply provides, use the Cisco power calculator on cisco.com to help determine the number of power supplies that is required for either combined or redundant mode.

Configuring the Redundant Mode

By default, the power supplies in the switch are set to operate in combined mode. To effectively use redundant mode, note the following:

- If you have the power supply mode set to redundant mode and only one power supply installed, your switch accepts the configuration but operates without redundancy.
- Choose a power supply module that is powerful enough to support the switch configuration.
- Use the Cisco Power Calculator to help assess the number of power supplies required by the system. Ensure that you install a sufficient number of power supply modules, so that the chassis and PoE requirements are less than the maximum available power. Power supplies automatically adjust the power resources at startup to accommodate the chassis and PoE requirements. Modules are brought up first, followed by IP phones.
- For optimal use of system power, choose power supply modules of the same capacity when configuring a redundant mode on the switch.

To configure redundant mode, perform this task:

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters the global configuration mode.
Step 2	**power redundancy-mode redundant** [**n+1** *standby-PSslot* | **n+1** *standby-PSslot*]	**power redundancy-mode redundant n+1** *standby-PSslot*—Configures the n+1 redundant

	Command or Action	Purpose
	Example: `Device(config)# power redundancy-mode redundant n+1 4`	mode. Enter the standby power supply module slot number. In the n+1 example here, the power supply module in slot PS4 is the designated standby module and has been configured accordingly. Operational power supply modules installed in all other slots, are active. If you are using power supply modules of different capacities, you must configure the power supply module with the highest wattage or capacity as the standby for the n+1 redundant mode.
Step 3	**end** **Example:** `Device(config)# end`	Exits global configuration mode.
Step 4	**show power** **Example:** `Device# show power`	Displays the power redundancy mode information.

Configuring the Combined Mode

To use the combined mode effectively, follow these guidelines:

- If you have the power supply mode set to combined mode and only one power supply installed, your switch accepts the configuration, but power is available from only one power supply.
- When your switch is configured to combined mode, available power is the sum of the individual power supplies

To configure combined mode on your switch, perform this task:

Before you begin

Note that this mode utilizes the available power from all the power supplies; however, your switch has no power redundancy.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters the global configuration mode.
Step 2	**power redundancy-mode combined** **Example:**	Sets the power supply mode to combined mode.

	Command or Action	Purpose
	`Device(config)# power redundancy-mode combined`	
Step 3	**end** **Example:** `Device(config)# end`	Exits global configuration mode.
Step 4	**show power** **Example:** `Device# show power`	Displays the power redundancy mode information.

Power Budgeting for Supervisor Modules

The power budget, or required budgeted power, is the power the system ***requires*** and ***reserves*** for supervisor modules, switching modules (line cards), and the fan tray to operate in the chassis. In the **show power**, and **show power detail** command outputs, this is displayed as `System Power`. The system does not allow any part of this required budgeted power to be automatically redirected for use by other components in the system.

This section describes how power budgeting works with respect to supervisor modules and the configuration options that are available.

By default, the system reserves power for a redundant setup, to ensure high availability. This means that the system reserves the power required by both the supervisor modules in the chassis, as part of the required budgeted power (`System Power`).

You can also configure the system to reserve power for a single supervisor. This configuration option is suited to situations where a single supervisor is installed and the total available power is not sufficent to enable all line cards and PoE ports. In such a scenario, configuring the switch to reserve power for a single supervisor enables you to free-up power and use it for other components, such as PoE ports, or line cards instead.

Note the following restrictions and guidelines:

- If you have installed both supervisor modules, you cannot configure the power budget mode for a single supervisor. The system rejects the configuration and following message is displayed: `cannot enable single sup mode when remote supervisor is present.`
- If you have installed both supervisor modules and the default setting is effective, you must install the necessary number of power supply modules to meet overall system requirements (including line cards and fan tray). Do not remove the second supervisor to remedy a situation where there is an insufficient number of power supply modules.
- If you have installed a single supervisor module and configured the power budget mode for a single supervisor, and you install a second supervisor:
  - The system will reject the configuration, and allow the first supervisor to come up.
  - If this action is accompanied by a low power condition where the system does not have sufficient power, linecards maybe denied power.

  For information about how to safely move from a single to a dual supervisor setup, see task *Moving from a Single to a Dual Supervisor Setup* below.

The following tasks describe the available configuration options:

Configuring the Power Budget Mode for a Single Supervisor

Beginning in the privileged EXEC mode, perform these steps to configure the power budget mode for a single supervisor setup:

Before you begin

Ensure that these prerequisites are met:

- You have installed only one supervisor module in the chassis.
- You have installed a blank in the second supervisor slot.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters the global configuration mode.
Step 2	**power budget mode** {**single-sup**} **Example:** `Device(config)# power budget mode single-sup`	Reserves power for one supervisor module in the chassis.
Step 3	**end** **Example:** `Device(config)# end`	Exits the global configuration mode.

Moving from a Single to a Dual Supervisor Setup

Beginning in the privileged EXEC mode, perform these steps to move from single to a dual supervisor setup:

Before you begin

Calculate the required power for a dual supervisor setup. Cisco Power Calculator (CPC) enables you to calculate the power supply requirements for a specified configuration:

1. Go to https://cpc.cloudapps.cisco.com/cpc → **Launch Cisco Power Calculator**.
2. Select applicable values for the `Product Family`, `Chassis`, `Supervisor Engine` (both supervisor slots), `Input Voltage`, and `Line Card` fields. Click **Next** to display results.
3. In the results that are displayed, locate the `Configuration Details` section and note the `Output Power` for the supervisor module. This is the amount of spare power that must be available in the system to safely install the second supervisor.
4. Enter the **show power** command in privileged EXEC mode.

   This command displays power supply configuration information.

In the output, check the difference between the `Total Maximum Available` and `Total Used`, this must be greater that what the CPC says in the `Output Power` column for the supervisor module. If this is the case, proceed with the task, if not, first install the required number of additional power supply modules.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters the global configuration mode.
Step 2	**no power budget mode** {**single-sup**} **Example:** `Device(config)# no power budget mode single-sup`	Reverts to the default setting where the system reserves power for both the supervisor modules in the chassis.
Step 3	**end** **Example:** `Device(config)# end`	Exits configuration mode.
Step 4	Insert the second supervisor module in the supervisor slot.	For detailed steps, see the Supervisor Module Installation Note → Removal and Replacement Procedures, on cisco.com.

Powering Down a Line Card

If your system does not have enough power for all modules installed in the switch, you can power down one or more line cards and place them in power-off mode.

To power down a line card, perform this task:

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters the global configuration mode.
Step 2	**hw-module slot** *card slot/slot number* **shutdown unpowered** **Example:** `Device(config)# hw-module slot 1/0 shutdown unpowered`	Powers down the specified module by placing it in low power mode.
Step 3	**end** **Example:**	Exits the global configuration mode

	Command or Action	Purpose
	`Device(config)# end`	

Configuration Examples for Operating States

The examples in this section show how to view the operating states of the system.

show power

The following is sample output of the **show power** command.

```
Device# show power
Power                                                          Fan States
Supply       Model No              Type Capacity  Status         1     2
------       --------------------  ---- --------  ------------  -----------
PS1          C9600-PWR-2KWAC       ac   2000 W    active        good good
PS2          C9600-PWR-2KWAC       ac   2000 W    active        good good
PS3          C9600-PWR-2KWAC       ac   2000 W    active        good good
PS4          C9600-PWR-2KWAC       ac   2000 W    active        good good

PS Current Configuration Mode : Combined
PS Current Operating State : none

Power supplies currently active : 4
Power supplies currently available : 4

Power Summary Maximum
(in Watts) Used Available
------------ ------ ---------
System Power 2860 7820
------------ ------ ---------
Total 2860 7820
```

show power detail

The **show power detail** command inlcudes the output of **show power** and **show power module** commmand in privileged EXEC mode.

```
Device# show power detail
Power                                                    Fan States
Supply  Model No              Type  Capacity  Status     1     2     3     4
------  --------------------  ----  --------  ---------- ----------------------

PS1     C9600-PWR-2KWAC       AC    2000 W    active     good  good  good  good
PS2     C9600-PWR-2KWAC       AC    2000 W    active     good  good  good  good
PS3     C9600-PWR-2KWAC       AC    2000 W    active     good  good  good  good
PS4     C9600-PWR-2KWAC       AC    2100 W    active     good  good  good  good

PS Current Configuration Mode : Combined
PS Current Operating State    : none

Power supplies currently active    : 4
Power supplies currently available : 4

Power Summary          Maximum
(in Watts)    Used    Available
------------  ------  ---------
```

```
System Power   2860    7820
-------------  ------  ---------
Total          2860    7820


Power Budget Mode           : Dual Sup

                                        Power                                  Out of  In
Mod     Model No              Priority  State     Budget  Instantaneous  Peak  Reset   Reset
---     --------------------  --------  --------  ------  -------------  ----  ------  -----
1       C9600-LC-24C            0       accepted  200     0              0     200     10
2       C9600-LC-48YL           1       accepted  230     0              0     230     10
3       C9600-SUP-1             0       accepted  775     0              0     775     202
4       C9600-SUP-1             0       accepted  775     0              0     775     202
5       C9600-LC-48YL           2       accepted  230     0              0     230     10
6       C9600-LC-24C            3       accepted  200     0              0     200     10
FM1     C9606-FAN                       accepted  450     --             --    450     --
---  --------------------  --------  --------  ------  -------------  ----  ------  -----
Total allocated power: 2860
Total required power: 2860
```

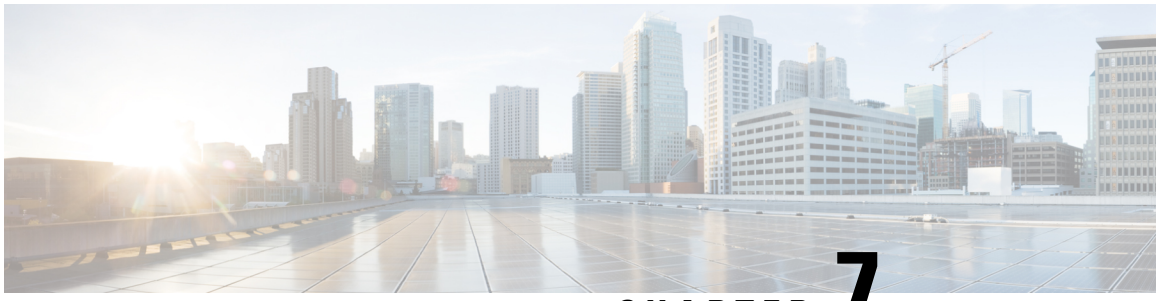
Feature History for Environmental Monitoring and Power Management

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.11.1	Environmental Monitoring and Power Management	Environmental monitoring of chassis components provides early warning indications of possible component failure. This warning helps you to ensure the safe and reliable operation of your system and avoid network interruptions.
Cisco IOS XE Cupertino 17.7.1	Environmental Monitoring and Power Management	Support for this feature was introduced on the Cisco Catalyst 9600 Series Supervisor 2 Module (C9600X-SUP-2.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.



CHAPTER 7

Configuring SDM Templates

Restrictions for Switch Device Manager Template

- Only the default core SDM template and some custom SDM templates are supported on the Cisco Catalyst 9600 Series Supervisor 2 Module. SDM templates like NAT, distribution, and customizable multicast are not supported.
- If the device is operating with NAT template, Switch Device Manager (SDM) templates cannot be customized.
- In a customizable SDM template the combined limit for multicast entries for Layer 2 and Layer 3 is 48K (K = 1024 entries).
- It is mandatory to assign a priority value to each of the features when customizing an SDM template. The priority value decides the resource allocation for the features, when the total number of all the resources specified in the customizable SDM template exceeds the total number of system resources assigned to a customizable SDM Template.
- The priority value of each feature should be unique. You cannot assign the same priority value to different features.
- In case of RMA or Supervisor replacement, restoring the backup configuration does not restore the customized template. You have to reconfigure the customized template.
- You can enable the 4K VLAN feature only through a customizable SDM template for 4K VLAN. You cannot customize any other FIB or ACL related features in the custom VLAN template.

  In a customizable SDM template for 4K VLAN, you can only increase the scale of VLAN from 1K to 4K. You cannot have custom VLAN values between 1K and 4K. Scales of other features that are limited due to limitations of the 1K VLAN table will remain the same.

Information About SDM Templates

You can use SDM templates to configure system resources to optimize support for specific features, depending on how your device is used in the network. You can select a standard template to provide maximum system usage for some functions.

The following table provides information about the SDM templates supported on the Cisco Catalyst 9600 Series Switches:

Table 20: Supported SDM Templates

Template Name	Cisco Catalyst 9600 Series Supervisor 1 Module	Cisco Catalyst 9600 Series Supervisor 2 Module
Core	Yes	Yes
NAT	Yes	NA (provided by custom)
Distribution	Yes	NA (provided by custom)
Custom	Yes	Yes

It is recommended that you reload the system as soon as you make a change to the SDM template. After you change the template and the system reboots, you can use the **show sdm prefer** privileged EXEC command to verify the new template configuration. If you enter the **show sdm prefer** command before you enter the **reload** privileged EXEC command, the **show sdm prefer** command shows the template currently in use and the template that will become active after a reload.

Note

- The default standard SDM template is the Core template.
- The NAT template cannot be used to create a customizable SDM template.

Customizable SDM Template

Overview of Customizable SDM Template

Switch Device Manager (SDM) templates can be used to configure system resources and optimize support for specific features. However standard SDM templates are defined based on how the device is deployed in the network.

A custom SDM template allows you to configure the features of the template based on your requirements and not the location of the device in the network.

- Starting with the Cisco IOS XE Amsterdam 17.3.1 release, you can configure a custom SDM template for Forwarding Information Base (FIB) features using the **sdm prefer custom fib** command. On the Cisco Catalyst 9600 Series Supervisor 2 Module, a custom SDM template for FIB features is configurable starting with the Cisco IOS XE Cupertino 17.7.1 release.

- Starting with the Cisco IOS XE Bengaluru 17.4.1 release, you can configure a custom SDM template for Access Control List (ACL) features using the **sdm prefer custom acl** command. Cisco Catalyst 9600 Series Supervisor 2 Module does not support SDM template for ACL.
- Starting with the Cisco IOS XE Bengaluru 17.5.1 release, you can configure a custom SDM template for 4K VLAN using the **sdm prefer custom vlan** command. Cisco Catalyst 9600 Series Supervisor 2 Module does not support SDM template for 4K VLAN, as these platforms natively support 4K VLANs.

A Customizable SDM template supports the following FIB features:

- Unicast MAC addresses
- Layer 3 Unicast forwarding
- Layer 2 Multicast forwarding
- Layer 3 Multicast forwarding
- Ingress Netflow
- Egress Netflow
- SGT/DGT Index/MPLS VPN Label

A Customizable SDM template on the Cisco Catalyst 9600 Series Supervisor 2 Module supports the following FIB features:

- Unicast MAC addresses
- FIB host routes for IPv4 and IPv6
- ACL compression for IPv4 and IPv6
- MPLS VPN Label

A Customizable SDM template supports the following ACL features:

- Ingress Access Control List (ACL)
- Egress ACL
- Ingress Quality of Service (QoS)
- Egress QoS
- Netflow ACL
- Policy Based Routing (PBR)/Network Address Translation (NAT)
- Locator/ID Separation Protocol (LISP)
- Tunnels

A Customizable SDM template for 4K VLAN supports only the 4K VLAN feature. You can increase the scale of VLAN from 1k to 4k. A Customizable SDM template for 4K VLAN increases the number of supported Switch Virtual Interfaces (SVI) to 4000.

The following table shows the minimum and maximum scale values that can be configured for each of the FIB features, the step units and the default values that will be applied when no custom values are chosen for a feature.

Table 21: Scale values and Default values for FIB features

Feature name	Scale Values (Min-Max)	Step Units	Default Values
MAC addresses	32768 - 131072	16384	32768
Unicast routes	65536 - 262144	16384	65536
Layer 2 Multicast	0, 16384 - 32768	16384	16384
Layer 3 Multicast	0, 16384 - 32768	16384	16384
SG Hash/MPLS	0, 32768 - 65536	32768	32768
Ingress Netflow	0, 32768 - 65536	32768	32768
Egress NetFlow	0, 32768 - 65536	32768	0

Table 22: Scale values and Default values for FIB features on the Cisco Catalyst 9600 Series Supervisor 2 Module

Feature name	Scale Values (Min-Max)	Step Units	Default Values
MAC addreses	32768 - 262144	1024	131072
FIB host routes (IPv4/IPv6)	32768 - 262144	1024	131072/65536
ACL (object-group and security group) compression (IPv4/IPv6)	0 - 262144	1024	32768/16384
MPLS label[9]	0 – 524288	1024	262144

[9] The MPLS label scale is limited to a maximum of 64000 entries by the **Prefix Object Gid** resource.

The following table shows the minimum and maximum scale values that can be configured for each of the ACL features, the step units and the default values that will be applied when no custom values are chosen for a feature.

Table 23: Scale values and Default values for ACL features

Feature name	Scale Values (Min-Max)	Step Units	Default Values
Ingress ACL	4096 - 26624, 27648	2048	4096
Egress ACL	4096 - 26624, 27648	2048	4096
Ingress QoS	1024, 2048 - 16384	2048	1024
Egress QoS	1024, 2048 - 16384	2048	1024
Netflow ACL	1024 - 2048	1024	1024
PBR/ NAT	1024, 2048 - 16384	2048	1024
LISP	1024 - 2048	1024	1024

Feature name	Scale Values (Min-Max)	Step Units	Default Values
Tunnels	1024 - 3072	1024	1024

You can determine which features are allocated the resources first by assigning them a priority using the **priority** keyword. The lower the priority-value assigned to a feature the higher its priority in resource allocation. The resource allocation algorithm uses the priority-values to determine the number of resources assigned to each feature.

Once you have configured a customized template the device will have to be reloaded for the template to take effect.

Note

- NetFlow FIB entries consume twice as many hardware entries as configured, and SG Hash FIB entries consume half as many hardware entries as configured when NetFlow allocation is less than the allowed maximum value of 128K.

  Note This is not applicable to the Cisco Catalyst 9600 Series Supervisor 2 Module.

- For features where the scale value can be set to zero, you need to specify the scale value as zero. If not, the default value will be assigned as the scale value.

System resource allocation for Customizable SDM Template

The total number of system resources assigned to a Customizable SDM Template is 416K for FIB features and 52K for ACL features. If the total number of all the resources specified exceeds 416K for FIB features or 52K for ACL features, the system starts to lower the number of allotted resources starting with the feature assigned the highest number. A higher priority value or number assigned to a feature indicates a lower priority.

When the total number of resources assigned in the Customizable SDM Template is less than 416K for FIB features or less than 52K for ACL features:

- All the features specified in the template are allotted resources as customized in the template. Any features not specified in the template are allotted the default number of resources.
- If the total number of resources assigned to the FIB features multicast layer 2 and layer 3 exceeds 48K, then the scale of the multicast feature assigned the lower priority is reduced until the total number of resources assigned is equal to 48K.
- Resources that are not allotted will not be distributed.

When the total number of resources assigned in the Customizable SDM Template is more than 416K for FIB features and more than 52K for ACL features:

- All the features for which a custom scale is not specified are allotted the default values.
- If the total number of resources assigned to FIB features multicast layer 2 and layer 3 exceeds 48K, then the scale of the multicast feature that is assigned the lower priority is reduced until the total number of resources assigned is less than or equal to 48K.

- The number of resources allotted to the feature with the highest priority value are decreased by the step value.
- If the total number of resources still exceeds 416K for FIB features or 52K for ACL features, the resources allotted to the next feature with the highest priority value are decreased by the step value.
- While lowering the resources allotted to a feature, the scale is lowered only until the default value for that feature. If further adjustment is required, the resources allotted to the next feature on the priority list are reduced.

Note The custom value entered by you for any feature is rounded up to the next step value. For example, if you enter a value of 40K for SGT it's rounded up to 64K.

System resource allocation for Customizable SDM Template on the Cisco Catalyst 9600 Series Supervisor 2 Module

The total number of system resources assigned to a Customizable SDM Template is 608K for FIB features. If the total number of resources specified exceeds 608K for FIB features, the system starts to lower the number of allotted resources starting with the feature assigned the highest number. A higher priority value or number assigned to a feature indicates a lower priority.

When the total number of resources assigned in the Customizable SDM Template is less than 608K for FIB features:

- All the features specified in the template are allotted resources as customized in the template. Any features not specified in the template are allotted the default number of resources.
- Resources that are not allotted will not be distributed.

When the total number of resources assigned in the Customizable SDM Template is more than 608K for FIB features:

- All the features for which a custom scale is not specified are allotted the default values.
- The number of resources allotted to the feature with the highest priority value are decreased by the step value.
- If the total number of resources still exceeds 608K for FIB features, the resources allotted to the next feature with the highest priority value are decreased by the step value.
- While lowering the resources allotted to a feature, the scale is lowered only until the default value for that feature. If further adjustment is required, the resources allotted to the next feature on the priority list are reduced.

Customizable SDM Template and High Availability

On a device which supports High Availability, when a Customizable SDM Template is configured on the active Supervisor it also takes effect on the standby Supervisor.

If the standby Supervisor is configured with a different custom template than the active Supervisor, the Customizable SDM Template of the active Supervisor is configured on the standby Supervisor during initialization.

Customizable SDM Template and StackWise Virtual

On a device which supports StackWise Virtual, when an SDM Template is configured on the active Supervisor it also takes effect on the standby chassis.

If the standby chassis is configured with a different custom template than the active Supervisor, the SDM Template of the active Supervisor is configured on the standby chassis during initialization. The standby chassis undergoes an extra reload for the template to take effect.

Customizable SDM Template and ISSU

When a device undergoes an In-Service Software Upgrade (ISSU) to a higher release and there's a change in the resource allocation algorithm, this upgrade can result in a different scale for the same user input. The change in scale is detected and notified via a syslog message. The system continues to operate with the earlier scale.

You can view the change in scale by using the **show sdm prefer custom scale-change** command. You can apply this change in scale by using the **sdm prefer custom commit** command. The device has to be reloaded for the change to take effect.

When a device with a customizable SDM template for FIB features undergoes a downgrade to a release earlier than the Cisco IOS XE Amsterdam 17.3.1 release, you need to change the SDM template to a static SDM template before the downgrade. You can change the template using the **sdm prefer template name** command. Reload the system for the change to take effect before proceeding with the downgrade.

When a device with a customizable SDM template for ACL features undergoes a downgrade to a release earlier than the Cisco IOS XE Bengaluru 17.4.1 release, you need to change the SDM template to a static SDM template before the downgrade.

When a device has customizable SDM templates for both FIB and ACL features customized in the Cisco IOS XE Bengaluru 17.4.1 release and it downgrades to the Cisco IOS XE Amsterdam 17.3.1 release, the device will be restored with the customizations for the FIB features. The scale numbers for the ACL features will be alloted based on the scale values of the standard SDM template. The information about the customization of the ACL features will be preserved. The device will be restored with the customizations for the ACL features when it upgrades to the Cisco IOS XE Bengaluru 17.4.1 release.

How to Configure SDM Templates

Setting the SDM Template

Follow these steps to use the SDM template to maximize feature usage:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**sdm prefer** { **core** | **nat** | **distribution** | **custom** } **Example:** `Device(config)# sdm prefer distribution`	Specifies the SDM template to be used on the switch. The keywords have these meanings: • **core** —Sets the Core template. • **nat** —Maximizes the NAT configuration on the switch. • **distribution** —Sets the Distribution template. • **custom** —Sets the Custom template for FIB, ACL features or for VLAN.The custom templates allow you to configure the values of certain FIB features, ACL features or the VLAN feature. **Note** The **no sdm prefer** command and a default template is not supported.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 5	**reload** **Example:** `Device# reload`	Reloads the operating system. After the system reboots, you can use the **show sdm prefer** privileged EXEC command to verify the new template configuration. If you enter the **show sdm prefer** command before you enter the reload privileged EXEC command, the **show sdm prefer** command shows the template currently in use and the template that will become active after a reload.

Configuring a Customizable SDM Template for FIB Features

To create a customizable SDM Template for FIB features, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**sdm prefer custom fib** **Example:** `Device(config)#sdm prefer custom fib`	Creates a customizable SDM template for FIB features. Enters a sub-mode for customizing features.
Step 4	**mac-address** *number-of-entries* **priority** *priority-value* **Example:** `Device(config-sdm-fib)#mac-address 128 priority 1`	Specifies the number of entries allotted for MAC addresses. The value ranges from 32K to 128K. The value is rounded up to the next 16K unit. The priority values range 1–7.
Step 5	**ipv4_and_ipv6 unicast** *number-of-entries* **priority** *priority-value* **Example:** `Device(config-sdm-fib)#ipv4_and_ipv6 unicast 256 priority 2`	Specifies the number of entries allotted for IPv4 and IPv6 Unicast. The value ranges from 64K to 256K.The priority values range 1–7.
Step 6	**ipv4_and_ipv6 multicast l3** *number-of-entries* **priority** *priority-value* **Example:** `Device(config-sdm-fib)#ipv4_and_ipv6 multicast l3 32 priority 3`	Specifies the number of entries allotted for layer 3 IPv4 and IPv6 Multicast. The value ranges from 16 to 32, 0 (zero) can also be entered as the value. The priority values range 1–7.
Step 7	**ipv4_and_ipv6 multicast l2** *number-of-entries* **priority** *priority-value* **Example:** `Device(config-sdm-fib)#ipv4_and_ipv6 multicast l2 32 priority 4`	Specifies the number of entries allotted for layer 2 IPv4 and IPv6 Multicast. The value ranges from 16 to 32, 0 (zero) can also be entered as the value.The priority values range 1–7.
Step 8	**netflow_out** *number-of-entries* **priority** *priority-value* **Example:** `Device(config-sdm-fib)#netflow_out 64 priority 5`	Specifies the number of entries allotted for Netflow egress. The value ranges from 32K to 64K, 0 (zero) can also be entered as the value.The priority values range 1–7.

	Command or Action	Purpose
Step 9	**netflow-in** *number-of-entries* **priority** *priority-value* **Example:** `Device(config-sdm-fib)# netflow_in 64 priority 6`	Specifies the number of entries allotted for Netflow ingress. The value ranges from 32K to 64K, 0 (zero) can also be entered as the value.The priority values range 1–7.
Step 10	**sgt_or_mpls_vpn** *number-of-entries* **priority** *priority-value* **Example:** `Device(config-sdm-fib)# sgt_or_mpls_vpn 64 priority 7`	Specifies the number of entries allotted for SGT or MPLS VPN. The value ranges from 32K to 64K, 0 (zero) can also be entered as the value.The priority values range 1–7.
Step 11	**end** **Example:** `Device(config-sdm-fib)# end`	Returns to privileged EXEC mode.
Step 12	**show sdm prefer custom** **Example:** `Device# show sdm prefer custom`	Displays the custom values that will be applied to the features in the customizable SDM template.
Step 13	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 14	**sdm prefer custom commit** **Example:** `Device(config)# sdm prefer custom commit`	Changes the running SDM preferences to the values in the customized template. The new template takes effect on the next reload.
Step 15	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 16	**reload** **Example:** `Device# reload`	Reloads the device and applies the customized SDM template.

What to do next

Once you view the custom values that will be applied to the features in the customizable SDM template using the **show sdm prefer custom** command, if required you can make changes to the values. To clear all the custom values that you have assigned to the features in the customized SDM template use the **sdm prefer custom fib clear** command.

If you want to change the custom value assigned to a feature without changing its priority value, you can simply overwrite the custom value assigned to the feature. For example, if you have assigned **mac-address 128 priority 1** you can overwrite this to **mac-address 32 priority 1**. If you want to change the priority value assigned to a feature, and if that priority value is already assigned to another feature you'll have to clear the custom value assigned to the other feature by using the **no** form of the command for that feature. You can then assign the priority value to the first feature. You'll have to reconfigure the other feature for it to have a non-default value.

The current customization context is valid only until **sdm prefer custom commit** command is issued. If you want to change any value after the commit CLI is issued, it will be considered as a new customization context. You will need to re-enter all the required feature values.

Configuring a Customizable SDM Template for FIB Features on the C9600X-SUP-2 Module

To create a customizable SDM Template for FIB features on the C9600X-SUP-2 module, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**sdm prefer custom fib** **Example:** `Device(config)#sdm prefer custom fib`	Creates a customizable SDM template for FIB features. Enters a sub-mode for customizing features.
Step 4	**mac-address** *number-of-entries* **priority** *priority-value* **Example:** `Device(config-sdm-fib)#mac-address 128 priority 1`	Specifies the number of entries allotted for MAC addresses. The maximum supported entries is 256K. The priority values range from 1 to 4.
Step 5	**ipv4-and-ipv6 host-route** *number-of-entries* **priority** *priority-value*	Specifies the number of entries allotted for IPv4 and IPv6 host route. The value ranges

	Command or Action	Purpose
	Example: `Device(config-sdm-fib)#ipv4-and-ipv6 host-route 256 priority 2`	from 1K to 256K. The priority values range from 1 to 4.
Step 6	**mpls-labels** *number-of-entries* **priority** *priority-value* **Example:** `Device(config-sdm-fib)#mpls-labels 256 priority 3`	Specifies the number of entries allotted for MPLS labels. The value ranges from 0 to 512K, 0 (zero) can also be entered as the value. The priority values range from 1 to 4.
Step 7	**og-sgacl** *number-of-entries* **priority** *priority-value* **Example:** `Device(config-sdm-fib)# og-sgacl 256 priority 4`	Specifies the number of entries allotted for object-group access control list (OGACL) or security group access control list (SGACL) hosts. The value ranges from 0 to 256K, 0 (zero) can also be entered as the value. The priority values range from 1 to 4.
Step 8	**end** **Example:** `Device(config-sdm-fib)# end`	Returns to privileged EXEC mode.
Step 9	**show sdm prefer custom** **Example:** `Device# show sdm prefer custom`	Displays the custom values that will be applied to the features in the customizable SDM template.
Step 10	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 11	**sdm prefer custom commit** **Example:** `Device(config)# sdm prefer custom commit`	Changes the running SDM preferences to the values in the customized template. The new template takes effect on the next reload.
Step 12	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 13	**reload** **Example:** `Device# reload`	Reloads the device and applies the customized SDM template.

Configuring a Customizable SDM Template for ACL Features

To create a customizable SDM Template for ACL features, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**sdm prefer custom acl** **Example:** `Device(config)#sdm prefer custom acl`	Creates a customizable SDM template for ACL features. Enters a sub-mode for customizing features.
Step 4	**acl-ingress** *number-of-entries* **priority** *priority-value* **Example:** `Device(config-sdm-acl)#acl-ingress 26 priority 1`	Specifies the number of entries allotted for Ingress ACL. The value ranges from 4K to 27K. The value is rounded up to the next 2K unit. The priority values range 1–8.
Step 5	**acl-egress** *number-of-entries* **priority** *priority-value* **Example:** `Device(config-sdm-acl)#acl-engress 20 priority 2`	Specifies the number of entries allotted for Engress ACL. The value ranges from 4K to 27K. The value is rounded up to the next 2K unit. The priority values range 1–8.
Step 6	**qos-ingress** *number-of-entries* **priority** *priority-value* **Example:** `Device(config-sdm-acl)#qos-ingress 2 priority 3`	Specifies the number of entries allotted for Ingress QoS. The value ranges from 2K to 16K. The value is rounded up to the next 2K unit. The priority values range 1–8.
Step 7	**qos-egress** *number-of-entries* **priority** *priority-value* **Example:** `Device(config-sdm-acl)#qos-egress 2 priority 4`	Specifies the number of entries allotted for Egress QoS. The value ranges from 2K to 16K. The value is rounded up to the next 2K unit. The priority values range 1–8.
Step 8	**nfl** *number-of-entries* **priority** *priority-value* **Example:** `Device(config-sdm-acl)#nfl 2 priority 5`	Specifies the number of entries allotted for Netflow ACL. The value ranges from 1K to 2K. The priority values range 1–8. The entries alloted for Netflow ACL are divided equally between ingress and egress entries.

	Command or Action	Purpose
Step 9	**pbr** *number-of-entries* **priority** *priority-value* **Example:** `Device(config-sdm-acl)#pbr 2 priority 6`	Specifies the number of entries allotted for PBR/NAT. The value ranges from 2K to 16K. The value is rounded up to the next 2K unit. The priority values range 1–8.
Step 10	**lisp** *number-of-entries* **priority** *priority-value* **Example:** `Device(config-sdm-acl)#lisp 2 priority 7`	Specifies the number of entries allotted for LISP. The value ranges from 1K to 2K. The priority values range 1–8.
Step 11	**tunnels** *number-of-entries* **priority** *priority-value* **Example:** `Device(config-sdm-acl)#tunnels 1 priority 8`	Specifies the number of entries allotted for Tunnel Termination Entries. The value ranges from 1K to 3K. The specified value will be lowered by 256 entries. 1K, 2K, 3K tunnel scale will be mapped to 0.75K, 1.75K, 2.75K respectively.The priority values range 1–8.
Step 12	**end** **Example:** `Device(config-sdm-acl)# end`	Returns to privileged EXEC mode.
Step 13	**show sdm prefer custom** **Example:** `Device# show sdm prefer custom`	Displays the custom values that will be applied to the features in the customizable SDM template.
Step 14	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 15	**sdm prefer custom commit** **Example:** `Device(config)# sdm prefer custom commit`	Changes the running SDM preferences to the values in the customized template. The new template takes effect on the next reload.
Step 16	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 17	**reload** **Example:** `Device# reload`	Reloads the device and applies the customized SDM template.

What to do next

Once you view the custom values that will be applied to the features in the customizable SDM template using the **show sdm prefer custom** command, if required you can make changes to the values. To clear all the custom values that you have assigned to the features in the customized SDM template use the **sdm prefer custom acl clear** command.

If you want to change the custom value assigned to a feature without changing its priority value, you can simply overwrite the custom value assigned to the feature. For example, if you have assigned **acl-ingress 26 priority 1** you can overwrite this to **acl-ingress 24 priority 1**. If you want to change the priority value assigned to a feature, and if that priority value is already assigned to another feature you'll have to clear the custom value assigned to the other feature by using the **no** form of the command for that feature. You can then assign the priority value to the first feature. You'll have to reconfigure the other feature for it to have a non-default value.

The current customization context is valid only until **sdm prefer custom commit** command is issued. If you want to change any value after the commit CLI is issued, it will be considered as a new customization context. You will need to re-enter all the required feature values.

Configuring a Customizable SDM Template for 4k VLAN

To create a customizable SDM Template for 4k VLAN, perform this procedure:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**sdm prefer custom vlan** **Example:** `Device(config)#sdm prefer custom vlan`	Creates a customizable SDM template for 4k VLAN.
Step 4	**end** **Example:** `Device(config-sdm-vlan)# end`	Returns to privileged EXEC mode.
Step 5	**show sdm prefer custom** **Example:** `Device# show sdm prefer custom`	Displays the custom values that will be applied to the features in the customizable SDM template.
Step 6	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 7	**sdm prefer custom commit** **Example:** `Device(config)# sdm prefer custom commit`	Changes the running SDM preferences to the values in the customized template. The new template takes effect on the next reload.
Step 8	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 9	**reload** **Example:** `Device# reload`	Reloads the device and applies the customized SDM template.

Clearing the customized values of the SDM Template

To clear the custom values that have been assigned to the features in the customized SDM template use the **sdm prefer custom fib clear** command or the **sdm prefer custom acl clear** command.

This command will clear the customization configuration that is not committed yet.

Once you issue this command, all the custom values for the features have to be reconfigured.

Monitoring and Maintaining SDM Templates

Verifying SDM Templates

Use the following commands to monitor and maintain SDM templates.

Command	Purpose
show sdm prefer	Displays the SDM template in use.

Note The SDM templates contain only those commands that are defined as part of the templates. If a template enables another related command that is not defined in the template, then this other command will be visible when the **show running config** command is entered. For example, if the SDM template enables the **switchport voice vlan** command, then the **spanning-tree portfast edge** command may also be enabled (although it is not defined on the SDM template).

If the SDM template is removed, then other such related commands are also removed and have to be reconfigured explicitly.

Verifying Customizable SDM Templates

Use the following commands to verify the customizable SDM Template that will be applied.

Table 24: Commands to verify the customizable SDM template

Command	Description
show sdm prefer custom	Displays the custom values that will be applied to the features in the customizable SDM template.
show sdm prefer custom user-input	Displays the values that were entered by the user in the customizable SDM template.
show sdm prefer	Displays the customized SDM template that is currently active.

If any feature in the Customizable SDM template has been assigned a scale value of zero, the feature will not be listed in the output of the **show sdm prefer custom** command after the device is reloaded.

Configuration Examples for SDM Templates

Examples: Displaying SDM Templates

The following example output shows the core template information:

```
Device# show sdm prefer core

This is the Core template.
  Security Ingress IPv4 Access Control Entries*:       7168  (current) - 7168  (proposed)
  Security Ingress Non-IPv4 Access Control Entries*:   5120  (current) - 5120  (proposed)
  Security Egress IPv4 Access Control Entries*:        7168  (current) - 7168  (proposed)
  Security Egress Non-IPv4 Access Control Entries*:    8192  (current) - 8192  (proposed)
  QoS Ingress IPv4 Access Control Entries*:            4096  (current) - 4096  (proposed)
  QoS Ingress Non-IPv4 Access Control Entries*:        4096  (current) - 4096  (proposed)
  QoS Egress IPv4 Access Control Entries*:             4096  (current) - 4096  (proposed)
  QoS Egress Non-IPv4 Access Control Entries*:         4096  (current) - 4096  (proposed)
  Netflow Input Access Control Entries*:                512  (current) -  512  (proposed)
  Netflow Output Access Control Entries*:               512  (current) -  512  (proposed)
  Flow SPAN Input Access Control Entries*:              512  (current) -  512  (proposed)
  Flow SPAN Output Access Control Entries*:             512  (current) -  512  (proposed)
  Number of VLANs:                                    4094
  Unicast MAC addresses:                             32768
  Overflow Unicast MAC addresses:                      768
  Overflow L2 Multicast entries:                      2304
  L3 Multicast entries:                              32768
  Overflow L3 Multicast entries:                       768
  Ipv4/Ipv6 shared unicast routes:                  212992
  Overflow shared unicast routes:                     1536
  Policy Based Routing ACEs / NAT ACEs:               3072
  Tunnels:                                            2816
  LISP Instance Mapping Entries:                      2048
  Control Plane Entries:                               512
  Input Netflow flows:                               32768
  Output Netflow flows:                              32768
  SGT/DGT (or) MPLS VPN entries:                     32768
  SGT/DGT (or) MPLS VPN Overflow entries:              768
  Wired clients:                                      2048
  MACSec SPD Entries:                                  256
  MPLS L3 VPN VRF:                                    1024
  MPLS Labels:                                       45056
```

```
  MPLS L3 VPN Routes VRF Mode:                          209920
  MPLS L3 VPN Routes Prefix Mode:                       32768
  MVPN MDT Tunnels:                                     1024
  L2 VPN EOMPLS Attachment Circuit:                     1024
  MAX VPLS Bridge Domains :                             1000
  MAX VPLS Peers Per Bridge Domain:                     128
  MAX VPLS/VPWS Pseudowires :                           16384
Ipv4/Ipv6 Direct and Indirect unicast routes share same space
* values can be modified by sdm cl
```

The following example output shows the NAT template information:

```
Device# show sdm prefer nat

This is the NAT template.
  Security Ingress IPv4 Access Control Entries*:       7168  (current) - 7168  (proposed)
  Security Ingress Non-IPv4 Access Control Entries*:   5120  (current) - 5120  (proposed)
  Security Egress IPv4 Access Control Entries*:        3072  (current) - 3072  (proposed)
  Security Egress Non-IPv4 Access Control Entries*:    5120  (current) - 5120  (proposed)
  QoS Ingress IPv4 Access Control Entries*:            2560  (current) - 2560  (proposed)
  QoS Ingress Non-IPv4 Access Control Entries*:        1536  (current) - 1536  (proposed)
  QoS Egress IPv4 Access Control Entries*:             3072  (current) - 3072  (proposed)
  QoS Egress Non-IPv4 Access Control Entries*:         1024  (current) - 1024  (proposed)
  Netflow Input Access Control Entries*:               1024  (current) - 1024  (proposed)
  Netflow Output Access Control Entries*:              1024  (current) - 1024  (proposed)
  Flow SPAN Input Access Control Entries*:             512   (current) - 512   (proposed)
  Flow SPAN Output Access Control Entries*:            512   (current) - 512   (proposed)
  Number of VLANs:                                     4094
  Unicast MAC addresses:                               32768
  Overflow Unicast MAC addresses:                      768
  Overflow L2 Multicast entries:                       2304
  L3 Multicast entries:                                32768
  Overflow L3 Multicast entries:                       768
  Ipv4/Ipv6 shared unicast routes:                     212992
  Overflow shared unicast routes:                      1536
  Policy Based Routing ACEs / NAT ACEs:                15872
  Tunnels:                                             1792
  LISP Instance Mapping Entries:                       1024
  Control Plane Entries:                               1024
  Input Netflow flows:                                 32768
  Output Netflow flows:                                32768
  SGT/DGT (or) MPLS VPN entries:                       32768
  SGT/DGT (or) MPLS VPN Overflow entries:              768
  Wired clients:                                       2048
  MACSec SPD Entries:                                  256
  MPLS L3 VPN VRF:                                     1024
  MPLS Labels:                                         45056
  MPLS L3 VPN Routes VRF Mode:                         209920
  MPLS L3 VPN Routes Prefix Mode:                      32768
  MVPN MDT Tunnels:                                    1024
  L2 VPN EOMPLS Attachment Circuit:                    1024
  MAX VPLS Bridge Domains :                            1000
  MAX VPLS Peers Per Bridge Domain:                    128
  MAX VPLS/VPWS Pseudowires :                          16384
Ipv4/Ipv6 Direct and Indirect unicast routes share same space
* values can be modified by sdm cli
```

The following example output shows the distribution template information:

```
Device# show sdm prefer distribution

This is the Distribution template.
  Security Ingress IPv4 Access Control Entries*:       7168  (current) - 7168  (proposed)
  Security Ingress Non-IPv4 Access Control Entries*:   5120  (current) - 5120  (proposed)
  Security Egress IPv4 Access Control Entries*:        7168  (current) - 7168  (proposed)
```

```
  Security Egress Non-IPv4 Access Control Entries*:    8192  (current) - 8192  (proposed)
  QoS Ingress IPv4 Access Control Entries*:            5632  (current) - 5632  (proposed)
  QoS Ingress Non-IPv4 Access Control Entries*:        2560  (current) - 2560  (proposed)
  QoS Egress IPv4 Access Control Entries*:             6144  (current) - 6144  (proposed)
  QoS Egress Non-IPv4 Access Control Entries*:         2048  (current) - 2048  (proposed)
  Netflow Input Access Control Entries*:               1024  (current) - 1024  (proposed)
  Netflow Output Access Control Entries*:              1024  (current) - 1024  (proposed)
  Flow SPAN Input Access Control Entries*:             512   (current) - 512   (proposed)
  Flow SPAN Output Access Control Entries*:            512   (current) - 512   (proposed)
  Number of VLANs:                                     4094
  Unicast MAC addresses:                               81920
  Overflow Unicast MAC addresses:                      768
  Overflow L2 Multicast entries:                       2304
  L3 Multicast entries:                                16384
  Overflow L3 Multicast entries:                       768
  Ipv4/Ipv6 shared unicast routes:                     114688
  Overflow shared unicast routes:                      1536
  Policy Based Routing ACEs / NAT ACEs:                3072
  Tunnels:                                             2816
  LISP Instance Mapping Entries:                       1024
  Control Plane Entries:                               1024
  Input Netflow flows:                                 49152
  Output Netflow flows:                                49152
  SGT/DGT (or) MPLS VPN entries:                       32768
  SGT/DGT (or) MPLS VPN Overflow entries:              768
  Wired clients:                                       2048
  MACSec SPD Entries:                                  256
  MPLS L3 VPN VRF:                                     1024
  MPLS Labels:                                         45056
  MPLS L3 VPN Routes VRF Mode:                         112640
  MPLS L3 VPN Routes Prefix Mode:                      32768
  MVPN MDT Tunnels:                                    1024
  L2 VPN EOMPLS Attachment Circuit:                    1024
  MAX VPLS Bridge Domains :                            1000
  MAX VPLS Peers Per Bridge Domain:                    128
  MAX VPLS/VPWS Pseudowires :                          16384
Ipv4/Ipv6 Direct and Indirect unicast routes share same space
* values can be modified by sdm cli
```

The following example output shows the SDM template information on Cisco Catalyst 9600 Series Supervisor 2 Module:

```
Device# show sdm prefer

Showing SDM Template Info

This is the Core template.
Feature-Name                                    Reserved-Scale
Unicast MAC addresses                           131072
  Resource-Programmed: EM
FIB Host Route                                  131072
  Resource-Programmed: EM
OG/SGACL Hosts/Cells                            32768
  Resource-Programmed: EM
Max MPLS Label                                  262144
  Resource-Programmed: EM
L3 Multicast entries                            32768 (**)
L2 Multicast entries                            16384 (**)
Number of VLANs                                 4094 (**)
Overflow Unicast MAC addresses                  512 (**)
Overflow L2 Multicast entries                   512 (**)
Overflow L3 Multicast entries                   512 (**)
Ipv4/Ipv6 shared unicast routes                 262144 (**)
Overflow shared unicast routes                  2000000 (**)
STP Instances                                   4094 (**)
```

```
Tunnels                                                 1024 (**)
VRF                                                     3840 (**)
Max MPLS VPN Routes Per-Vrf label mode                  2000000 (**)
Max MPLS VPN Routes Per-Prefix label mode               65536 (**)
Max L3 adjacency                                        131072 (**)
Max L3 Interface                                        8192 (**)
Max MPLS TE TUNNEL                                      4096 (**)

(**) - SDM library is referred to only obtain scale

Resource scale information
EM                                                      622592
```

Example: Configuring a customized SDM template

The following example output shows how to configure a customized SDM template for FIB features. In this example, as the SG Hash/MPLS and Ingress Netflow features haven't been assigned any resources in the customized template they are allotted resources according to their default values.

```
Device(config)# sdm prefer custom fib
Device(config-sdm-fib)# mac-address 128 priority 1
Device(config-sdm-fib)# ipv4_and_ipv6 unicast 256 priority 2
Device(config-sdm-fib)# ipv4_and_ipv6 multicast l3 32 priority 3
Device(config-sdm-fib)# ipv4_and_ipv6 multicast l2 32 priority 4
Device(config-sdm-fib)# netflow_out 64 priority 5
Device(config-sdm-fib)# end
```

The following example output shows how to configure a customized SDM template for FIB features on the Cisco Catalyst 9600 Series Supervisor 2 Module.

```
Device(config)# sdm prefer custom fib
Device(config-sdm-fib)# mac-address 128 priority 1
Device(config-sdm-fib)# ipv4-and-ipv6 host-route 256 priority 2
Device(config-sdm-fib)# mpls-labels 256 priority 3
Device(config-sdm-fib)# og-sgacl 256 priority 4
Device(config-sdm-fib)# end
```

In the following examples as the SGT/ MPLS VPN features are assigned zero resources, no resources will be allotted to these features.

```
Device(config)# sdm prefer custom fib
Device(config-sdm-fib)# ipv4_and_ipv6 unicast 164 priority 1
Device(config-sdm-fib)# mac-address 80 priority 2
Device(config-sdm-fib)# ipv4_and_ipv6 multicast l2 16 priority 4
Device(config-sdm-fib)# ipv4_and_ipv6 multicast l3 16 priority 3
Device(config-sdm-fib)# sgt_or_mpls_vpn 0
Device(config-sdm-fib)# netflow_in 32 priority 5
Device(config-sdm-fib)# netflow_out 32 priority 6
Device(config-sdm-fib)# end
```

The following example output shows how to configure a customized SDM template for ACL features. In this example, as the Tunnels feature hasn't been assigned any resources in the customized template it is allotted resources according to the default values.

```
Device(config)# sdm prefer custom acl
Device(config-sdm-acl)# acl-ingress 26 priority 1
Device(config-sdm-acl)# acl-engress 20 priority 2
Device(config-sdm-acl)# lisp 2 priority 3
Device(config-sdm-acl)# nfl 2 priority 4
Device(config-sdm-acl)# pbr 2 priority 5
Device(config-sdm-acl)# qos-ingress 2 priority 6
```

```
Device(config-sdm-acl)# qos-egress 2 priority 7
Device(config-sdm-acl)# end
```

The following example output shows how to configure a customized SDM template for 4K VLAN.

```
Device(config)# sdm prefer custom VLAN
Device(config-sdm-vlan)# end
```

Example: Displaying the customized SDM template

The following example output shows the proposed values in the customized SDM template for FIB and ACL features.

```
Device# show sdm prefer custom

Showing SDM Template Info

This is the Custom template
<SNIP>
  Number of VLANs:                                 4094
  Unicast MAC addresses*:                         32768  (current) - 131072 (proposed)

  Overflow Unicast MAC addresses*:                768    (current) - 1536   (proposed)

  L2 Multicast entries*:                          0      (current) - 16384  (proposed)

  Overflow L2 Multicast entries*:                 2304   (current) - 768    (proposed)

  L3 Multicast entries*:                          32768  (current) - 16384  (proposed)

  Overflow L3 Multicast entries*:                 768    (current) - 768    (proposed)

  Ipv4/Ipv6 shared unicast routes*:               212992 (current) - 180224 (proposed)

  Overflow shared unicast routes*:                1536   (current) - 2304   (proposed)

  Ingress Security Access Control Entries*:       24576  (current) - 26624  (proposed)

  Egress Security Access Control Entries*:        3072   (current) - 20480  (proposed)

  Ingress QoS Access Control Entries*:            8192   (current) - 1024   (proposed)

  Egress QoS Access Control Entries*:             8192   (current) - 1024   (proposed)

  Policy Based Routing ACEs / NAT ACEs*:          3072   (current) - 1024   (proposed)

  Netflow Input ACEs*:                            256    (current) - 512    (proposed)

  Netflow Output ACEs*:                           768    (current) - 512    (proposed)

  Flow SPAN ACEs*:                                256    (current) - 512    (proposed)

  Output Flow SPAN ACEs*:                         256    (current) - 512    (proposed)

  Tunnels*:                                       2816   (current) - 768    (proposed)

  LISP Instance Mapping Entries*:                 2048   (current) - 1024   (proposed)

  Control Plane Entries*:                         512    (current) - 512    (proposed)

  Input Netflow flows*:                           32768  (current) - 32768  (proposed)

  Output Netflow flows*:                          32768  (current) - 0      (proposed)
```

```
  SGT/DGT (or) MPLS VPN entries*:                     32768  (current) - 32768  (proposed)

  SGT/DGT (or) MPLS VPN Overflow entries*:            768    (current) - 768    (proposed)

  Wired clients:                                      2048
  MACSec SPD Entries*:                                256    (current) - 256    (proposed)

  VRF:                                                1024
  MPLS Labels:                                        45056
  MPLS L3 VPN Routes VRF Mode*:                       209920 (current) - 180224 (proposed)

  MPLS L3 VPN Routes Prefix Mode*:                    32768  (current) - 32768  (proposed)

  MVPN MDT Tunnels:                                   1024
  L2 VPN EOMPLS Attachment Circuit:                   1024
  MAX VPLS Bridge Domains :                           1000
  MAX VPLS Peers Per Bridge Domain:                   128
  MAX VPLS/VPWS Pseudowires :                         16384

Ipv4/Ipv6 Direct and Indirect unicast routes share same space
(*) values can be modified by sdm cli
The proposed values will take effect post reload.
```

The following example output shows the proposed values in the customized SDM template for FIB features on the Cisco Catalyst 9600 Series Supervisor 2 Module.

```
Device# show sdm  prefer custom

Showing SDM Template Info

This is the Core template.
Feature-Name                                        Reserved-Scale
Unicast MAC addresses*                              131072 (current) - 131072 (proposed)
  Resource-Programmed: EM
FIB Host Route*                                     131072 (current) - 147456 (proposed)
  Resource-Programmed: EM
OG/SGACL Hosts/Cells*                               32768  (current) - 32768  (proposed)
  Resource-Programmed: EM
Max MPLS Label*                                     262144 (current) - 262144 (proposed)
  Resource-Programmed: EM
L3 Multicast entries*                               32768  (current) - 32768  (proposed)
(**)
L2 Multicast entries*                               16384  (current) - 16384  (proposed)
(**)
Number of VLANs                                     4094 (**)
Overflow Unicast MAC addresses                      512 (**)
Overflow L2 Multicast entries                       512 (**)
Overflow L3 Multicast entries                       512 (**)
Ipv4/Ipv6 shared unicast routes                     262144 (**)
Overflow shared unicast routes                      2000000 (**)
STP Instances                                       4094 (**)
Tunnels                                             1024 (**)
VRF                                                 3840 (**)
Max MPLS VPN Routes Per-Vrf label mode              2000000 (**)
Max MPLS VPN Routes Per-Prefix label mode           65536 (**)
Max L3 adjacency                                    131072 (**)
Max L3 Interface                                    8192 (**)
Max MPLS TE TUNNEL                                  4096 (**)

(**) - SDM library is referred to only obtain scale

Resource scale information
EM                                                  622592
```

The following example output shows the values and priorities specified by the user in the custom template. As the SG Hash/MPLS, Ingress Netflow and Tunnels features haven't been assigned any resources in the customized template, they will be allotted resources according to their default values.

```
Device# show sdm prefer custom user-input

FIB FEATURE USER INPUT
User Input values
==========================

  FEATURE NAME                                                PRIORITY   SCALE
-------------------------------------------------------------------------------
  Unicast MAC addresses:                                       1          128*1024
  L2 Multicast entries:                                        4          32*1024
  L3 Multicast entries:                                        3          32*1024
  Ipv4/Ipv6 shared unicast routes:                             2          256*1024
  Output Netflow flows:                                        5          64*1024


System Default values
==========================

  FEATURE NAME                                                PRIORITY   SCALE
-------------------------------------------------------------------------------
  Input Netflow flows:                                        NA         32768
  SGT/DGT (or) MPLS VPN entries:                              NA         32768

ACL FEATURE USER INPUT
User Input values
==========================

  FEATURE NAME                                                PRIORITY   SCALE
-------------------------------------------------------------------------------
  Security Access Control Entries:                            1          26*1024
  Egress Security Access Control Entries:                     2          20*1024
  QoS Access Control Entries:                                 3          2*1024
  Egress QoS Access Control Entries:                          4          2*1024
  Policy Based Routing ACEs / NAT ACEs:                       5          2*1024
  Netflow ACEs:                                               6          2*1024
  LISP Instance Mapping Entries:                              7          2*1024


System Default values
==========================

  FEATURE NAME                                                PRIORITY   SCALE
-------------------------------------------------------------------------------
  Tunnels:                                                     NA         1024
```

The following example output shows the values and priorities specified by the user in the custom template on the Cisco Catalyst 9600 Series Supervisor 2 Module.

```
Device# show sdm  prefer custom user-input

FIB FEATURE USER INPUT
User Input values
==========================

  FEATURE NAME                                                PRIORITY   SCALE
-------------------------------------------------------------------------------
  Unicast MAC addresses:                                      1          128*1024
  FIB Host Route:                                             2          144*1024
  OG/SGACL Hosts/Cells:                                       4          32*1024
  Max MPLS Label:                                             3          256*1024
```

```
System Default values
=========================

  FEATURE NAME                                                PRIORITY   SCALE
-------------------------------------------------------------------------------
  L3 Multicast entries:                                       NA         32768
  L2 Multicast entries:                                       NA         16384
```

The following example output shows the proposed values in the customized SDM template. As the SGT/ MPLS VPN features are assigned zero resources, no resources will be allotted to these features.

```
Device# show sdm prefer custom

Showing SDM Template Info

This is the Custom template
<SNIP>
  Unicast MAC addresses*:                          32768  (current) - 81920  (proposed)
  Overflow Unicast MAC addresses*:                 768    (current) - 1536   (proposed)
  L2 Multicast entries*:                           0      (current) - 16384  (proposed)
  Overflow L2 Multicast entries*:                  2304   (current) - 768    (proposed)
  L3 Multicast entries*:                           32768  (current) - 16384  (proposed)
  Overflow L3 Multicast entries*:                  768    (current) - 768    (proposed)
  Ipv4/Ipv6 shared unicast routes*:                212992 (current) - 180224 (proposed)
  Overflow shared unicast routes*:                 1536   (current) - 2304   (proposed)
  Ingress Security Access Control Entries*:        24576  (current) - 26624  (proposed)
  Egress Security Access Control Entries*:         3072   (current) - 20480  (proposed)
  Ingress QoS Access Control Entries*:             8192   (current) - 1024   (proposed)
  Egress QoS Access Control Entries*:              8192   (current) - 1024   (proposed)
  Policy Based Routing ACEs / NAT ACEs*:           3072   (current) - 1024   (proposed)
  Netflow Input ACEs*:                             256    (current) - 512    (proposed)
  Netflow Output ACEs*:                            768    (current) - 512    (proposed)
  Flow SPAN ACEs*:                                 256    (current) - 512    (proposed)
  Output Flow SPAN ACEs*:                          256    (current) - 512    (proposed)
  Tunnels*:                                        2816   (current) - 768    (proposed)
  LISP Instance Mapping Entries*:                  2048   (current) - 1024
  Input Netflow flows*:                            32768  (current) - 32768  (proposed)
  Output Netflow flows*:                           32768  (current) - 32768  (proposed)
  SGT/DGT (or) MPLS VPN entries*:                  32768  (current) - 0      (proposed)
  SGT/DGT (or) MPLS VPN Overflow entries*:         768    (current) - 768    (proposed)
  Wired clients:                                   2048
  MACSec SPD Entries*:                             256    (current) - 256    (proposed)
  VRF:                                             1024
  MPLS Labels:                                     45056
  MPLS L3 VPN Routes VRF Mode*:                    209920 (current) - 180224 (proposed)
  MPLS L3 VPN Routes Prefix Mode*:                 32768  (current) - 32768  (proposed)
  MVPN MDT Tunnels:                                1024
  L2 VPN EOMPLS Attachment Circuit:                1024
  MAX VPLS Bridge Domains :                        1000
  MAX VPLS Peers Per Bridge Domain:                128
  MAX VPLS/VPWS Pseudowires :                      16384
```

The following example output shows the values and priorities specified by the user in the custom template. No resources have been alloted to SGT/MPLS VPN features.

```
Device# show sdm prefer custom user-input

FIB FEATURE USER INPUT
User Input values
=========================
  FEATURE NAME                                                PRIORITY       SCALE
-------------------------------------------------------------------------------
Unicast MAC addresses:                                           2          80*1024
```

```
L2 Multicast entries:                                4          16*1024
L3 Multicast entries:                                3          16*1024
Ipv4/Ipv6 shared unicast routes:                     1          164*1024
Input Netflow flows:                                 5          32*1024
Output Netflow flows:                                6          32*1024
SGT/DGT (or) MPLS VPN entries:                       NA         0

ACL FEATURE USER INPUT
User Input values
==========================
  FEATURE NAME                                       PRIORITY   SCALE
-------------------------------------------------------------------------
  Security Access Control Entries:                   1          26*1024
  Egress Security Access Control Entries:            2          20*1024
  QoS Access Control Entries:                        3          2*1024
  Egress QoS Access Control Entries:                 4          2*1024
  Policy Based Routing ACEs / NAT ACEs:              5          2*1024
  Netflow ACEs:                                      6          2*1024
  LISP Instance Mapping Entries:                     7          2*1024


System Default values
==========================

  FEATURE NAME                                       PRIORITY   SCALE
-------------------------------------------------------------------------
  Tunnels:                                            NA         1024
```

The following example output shows the proposed values in the customized SDM template for 4k VLAN.

```
Device# show sdm prefer custom

Showing SDM Template Info

This is the Custom template.
Security Ingress IPv4 Access Control Entries*:          7168 (current) - 7168 (proposed)
Security Ingress Non-IPv4 Access Control Entries*:      5120 (current) - 5120 (proposed)
Security Egress IPv4 Access Control Entries*:           7168 (current) - 7168 (proposed)
Security Egress Non-IPv4 Access Control Entries*:       8192 (current) - 8192 (proposed)
QoS Ingress IPv4 Access Control Entries*:               5632 (current) - 5632 (proposed)
QoS Ingress Non-IPv4 Access Control Entries*:           2560 (current) - 2560 (proposed)
QoS Egress IPv4 Access Control Entries*:                6144 (current) - 6144 (proposed)
QoS Egress Non-IPv4 Access Control Entries*:            2048 (current) - 2048 (proposed)
Netflow Input Access Control Entries*:                  512  (current) - 512  (proposed)
Netflow Output Access Control Entries*:                 512  (current) - 512  (proposed)
Flow SPAN Input Access Control Entries*:                512  (current) - 512  (proposed)
Flow SPAN Output Access Control Entries*:               512  (current) - 512  (proposed)
Number of VLANs:                                        4094
Unicast MAC addresses*:                                 98304
Overflow Unicast MAC addresses*:                        768
Overflow L2 Multicast entries*:                         2048
L3 Multicast entries*:                                  16384
Overflow L3 Multicast entries*:                         768
Ipv4/Ipv6 shared unicast routes*:                       81920
Overflow shared unicast routes*:                        1536
Policy Based Routing ACEs / NAT ACEs*:                  3072
Tunnels*:                                               2816
LISP Instance Mapping Entries*:                         2048
Control Plane Entries*:                                 512
Input Netflow flows*:                                   49152
Output Netflow flows*:                                  49152
SGT/DGT (or) MPLS VPN entries*:                         32768
SGT/DGT (or) MPLS VPN Overflow entries*:                768
Wired clients:                                          2048
MACSec SPD Entries*:                                    256
```

```
VRF:                                                    1024
MPLS Labels:                                            45056
MPLS L3 VPN Routes VRF Mode*:                           81920
MPLS L3 VPN Routes Prefix Mode*:                        32768
MVPN MDT Tunnels:                                       1024
L2 VPN EOMPLS Attachment Circuit:                       1024
MAX VPLS Bridge Domains :                               1000
MAX VPLS Peers Per Bridge Domain:                       128
MAX VPLS/VPWS Pseudowires :                             16384
VLAN Filter Entries:                                    16384
```

Example: Applying the customized SDM template

The following example output shows how to apply a customized SDM template:

```
Device(config)# sdm prefer custom commit
Changes to the running SDM preferences have been stored and will take effect on the next
reload.
Device(config)# exit
Device# reload
```

Example: Clearing the customized values of the SDM template

The following example output shows how to clear a customized SDM template for FIB features after which the template can be recustomized:

```
Device(config)# sdm prefer custom fib clear
FIB customization changes, not yet committed will be cleared
Device(config-sdm-fib)# end
```

The following example output shows how to clear a customized SDM template for ACL features after which the template can be recustomized:

```
Device(config)# sdm prefer custom acl clear
ACL customization changes, not yet committed will be cleared
Device(config-sdm-fib)# end
```

Additional References for SDM Templates

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	*Command Reference (Catalyst 9600 Series Switches)*

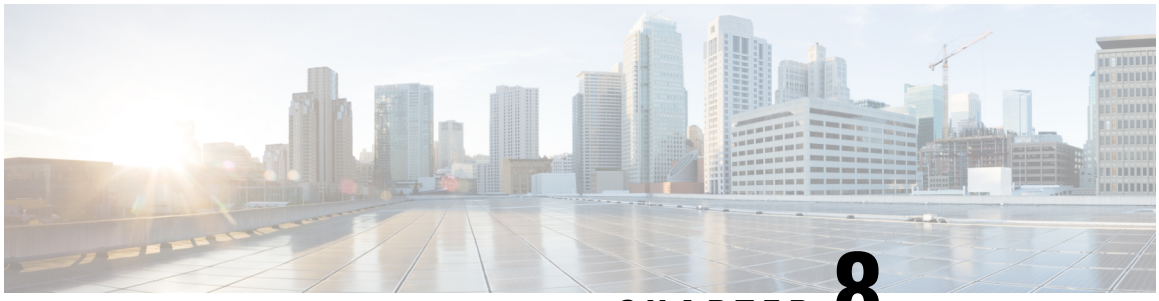
Feature History for SDM Templates

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.11.1	SDM Template	Standard SDM templates can be used to configure system resources to optimize support for specific features.
Cisco IOS XE Amsterdam 17.3.1	Customizable SDM Template for FIB Features	Support for customizable SDM templates for FIB features was introduced. Customizable SDM templates can be used to configure the features of the template as per the user's requirements.
Cisco IOS XE Bengaluru 17.4.1	Customizable SDM Template for ACL Features	Support for customizable SDM templates for ACL features was introduced. Customizable SDM templates can be used to configure the features of the template as per the user's requirements.
Cisco IOS XE Bengaluru 17.5.1	Customizable SDM template for 4k VLAN	Support for customizable SDM templates for 4k VLAN was introduced.
Cisco IOS XE Cupertino 17.7.1	SDM Template	Support for this feature was introduced on the Cisco Catalyst 9600 Series Supervisor 2 Module (C9600X-SUP-2).

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to Cisco Feature Navigator.



CHAPTER 8

Configuring System Message Logs

Information About Configuring System Message Logs

System Messsage Logging

By default, a switch sends the output from system messages and **debug** privileged EXEC commands to a logging process. . The logging process controls the distribution of logging messages to various destinations, such as the logging buffer, terminal lines, or a UNIX syslog server, depending on your configuration. The process also sends messages to the console.

When the logging process is disabled, messages are sent only to the console. The messages are sent as they are generated, so message and debug output are interspersed with prompts or output from other commands. Messages appear on the active consoles after the process that generated them has finished.

You can set the severity level of the messages to control the type of messages displayed on the consoles and each of the destinations. You can time-stamp log messages or set the syslog source address to enhance real-time debugging and management. For information on possible messages, see the system message guide for this release.

You can access logged system messages by using the switch command-line interface (CLI) or by saving them to a properly configured syslog server. The switch software saves syslog messages in an internal buffer on a standalone switch. If a standalone switch , the log is lost unless you had saved it to flash memory.

You can remotely monitor system messages by viewing the logs on a syslog server or by accessing the switch through Telnet, through the console port, or through the Ethernet management port.

Note The syslog format is compatible with 4.3 BSD UNIX.

System Log Message Format

System log messages can contain up to 80 characters and a percent sign (%), which follows the optional sequence number or time-stamp information, if configured. Depending on the switch, messages appear in one of these formats:

- *seq no:timestamp: %facility-severity-MNEMONIC:description (hostname-n)*
- *seq no:timestamp: %facility-severity-MNEMONIC:description*

The part of the message preceding the percent sign depends on the setting of these global configuration commands:

- **service sequence-numbers**
- **service timestamps log datetime**
- **service timestamps log datetime** [**localtime**] [**msec**] [**show-timezone**]
- **service timestamps log uptime**

Table 25: System Log Message Elements

Element	Description
seq no:	Stamps log messages with a sequence number only if the **service sequence-numbers** global configuration command is configured.
timestamp formats: *mm/dd h h:mm:ss* or *hh:mm:ss* (short uptime) or *d h* (long uptime)	Date and time of the message or event. This information appears only if the **service timestamps log** [**datetime** | **log**] global configuration command is configured.
facility	The facility to which the message refers (for example, SNMP, SYS, and so forth).
severity	Single-digit code from 0 to 7 that is the severity of the message.
MNEMONIC	Text string that uniquely describes the message.
description	Text string containing detailed information about the event being reported.

Default System Message Logging Settings

Table 26: Default System Message Logging Settings

Feature	Default Setting
System message logging to the console	Enabled.

Feature	Default Setting
Console severity	Debugging.
Logging file configuration	No filename specified.
Logging buffer size	4096 bytes.
Logging history size	1 message.
Time stamps	Disabled.
Synchronous logging	Disabled.
Logging server	Disabled.
Syslog server IP address	None configured.
Server facility	Local7
Server severity	Informational.

Syslog Message Limits

If you enabled syslog message traps to be sent to an SNMP network management station by using the **snmp-server enable trap** global configuration command, you can change the level of messages sent and stored in the switch history table. You also can change the number of messages that are stored in the history table.

Messages are stored in the history table because SNMP traps are not guaranteed to reach their destination. By default, one message of the level **warning** and numerically lower levels are stored in the history table even if syslog traps are not enabled.

When the history table is full (it contains the maximum number of message entries specified with the **logging history size** global configuration command), the oldest message entry is deleted from the table to allow the new message entry to be stored.

The history table lists the level keywords and severity level. For SNMP usage, the severity level values increase by 1. For example, *emergencies* equal 1, not 0, and *critical* equals 3, not 2.

How to Configure System Message Logs

Setting the Message Display Destination Device

If message logging is enabled, you can send messages to specific locations in addition to the console.

This task is optional.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**logging buffered** [*size*] **Example:** `Device(config)# logging buffered 8192`	Logs messages to an internal buffer on the switch. The range is 4096 to 2147483647 bytes. The default buffer size is 4096 bytes. If a standalone switchfails, the log file is lost unless you previously saved it to flash memory. See Step 4. **Note** Do not make the buffer size too large because the switch could run out of memory for other tasks. Use the **show memory** privileged EXEC command to view the free processor memory on the switch. However, this value is the maximum available, and the buffer size should *not* be set to this amount.
Step 3	**logging** *host* **Example:** `Device(config)# logging 125.1.1.100`	Logs messages to a UNIX syslog server host. *host* specifies the name or IP address of the host to be used as the syslog server. To build a list of syslog servers that receive logging messages, enter this command more than once.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 5	**terminal monitor** **Example:** `Device# terminal monitor`	Logs messages to a nonconsole terminal during the current session. Terminal parameter-setting commands are set locally and do not remain in effect after the session has ended. You must perform this step for each session to see the debugging messages.

Synchronizing Log Messages

You can synchronize unsolicited messages and **debug** privileged EXEC command output with solicited device output and prompts for a specific console port line or virtual terminal line. You can identify the types of messages to be output asynchronously based on the level of severity. You can also configure the maximum number of buffers for storing asynchronous messages for the terminal after which messages are dropped.

When synchronous logging of unsolicited messages and **debug** command output is enabled, unsolicited device output appears on the console or printed after solicited device output appears or is printed. Unsolicited messages and **debug** command output appears on the console after the prompt for user input is returned. Therefore, unsolicited messages and **debug** command output are not interspersed with solicited device output and prompts. After the unsolicited messages appear, the console again displays the user prompt.

This task is optional.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**line** [**console** | **vty**] *line-number* [*ending-line-number*] **Example:** `Device(config)# line console`	Specifies the line to be configured for synchronous logging of messages. • **console** —Specifies configurations that occur through the switch console port or the Ethernet management port. • **line vty** *line-number*—Specifies which vty lines are to have synchronous logging enabled. You use a vty connection for configurations that occur through a Telnet session. The range of line numbers is from 0 to 15. You can change the setting of all 16 vty lines at once by entering: line vty 0 15 You can also change the setting of the single vty line being used for your current connection. For example, to change the setting for vty line 2, enter: line vty 2 When you enter this command, the mode changes to line configuration.

	Command or Action	Purpose
Step 3	**logging synchronous** [**level** [*severity-level* | **all**] | **limit** *number-of-buffers*] **Example:** `Device(config)# logging synchronous level 3 limit 1000`	Enables synchronous logging of messages. • (Optional) **level** *severity-level*—Specifies the message severity level. Messages with a severity level equal to or higher than this value are printed asynchronously. Low numbers mean greater severity and high numbers mean lesser severity. The default is 2. • (Optional) **level all**—Specifies that all messages are printed asynchronously regardless of the severity level. • (Optional) **limit** *number-of-buffers*—Specifies the number of buffers to be queued for the terminal after which new messages are dropped. The range is 0 to 2147483647. The default is 20.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Disabling Message Logging

Message logging is enabled by default. It must be enabled to send messages to any destination other than the console. When enabled, log messages are sent to a logging process, which logs messages to designated locations asynchronously to the processes that generated the messages.

Disabling the logging process can slow down the switch because a process must wait until the messages are written to the console before continuing. When the logging process is disabled, messages appear on the console as soon as they are produced, often appearing in the middle of command output.

The **logging synchronous** global configuration command also affects the display of messages to the console. When this command is enabled, messages appear only after you press **Return**.

To reenable message logging after it has been disabled, use the **logging on** global configuration command.

This task is optional.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 2	**no logging console** **Example:** `Device(config)# no logging console`	Disables message logging.
Step 3	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Enabling and Disabling Time Stamps on Log Messages

By default, log messages are not time-stamped.

This task is optional.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 2	Use one of these commands:
	• **service timestamps log uptime**
	• **service timestamps log datetime**[**msec** | **localtime** | **show-timezone**]
	Example:
	`Device(config)# service timestamps log uptime`
	or
	`Device(config)# service timestamps log datetime`
		Enables log time stamps.
		• **log uptime**—Enables time stamps on log messages, showing the time since the system was rebooted.
		• **log datetime**—Enables time stamps on log messages. Depending on the options selected, the time stamp can include the date, time in milliseconds relative to the local time zone, and the time zone name.

Step 3	**end** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config)# end`	

Enabling and Disabling Sequence Numbers in Log Messages

If there is more than one log message with the same time stamp, you can display messages with sequence numbers to view these messages. By default, sequence numbers in log messages are not displayed.

This task is optional.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**service sequence-numbers** **Example:** `Device(config)# service sequence-numbers`	Enables sequence numbers.
Step 3	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Defining the Message Severity Level

Limit messages displayed to the selected device by specifying the severity level of the message.

This task is optional.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 2	**logging console** *level* **Example:** `Device(config)# logging console 3`	Limits messages logged to the console. By default, the console receives debugging messages and numerically lower levels.
Step 3	**logging monitor** *level* **Example:** `Device(config)# logging monitor 3`	Limits messages logged to the terminal lines. By default, the terminal receives debugging messages and numerically lower levels.
Step 4	**logging trap** *level* **Example:** `Device(config)# logging trap 3`	Limits messages logged to the syslog servers. By default, syslog servers receive informational messages and numerically lower levels.
Step 5	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Limiting Syslog Messages Sent to the History Table and to SNMP

This task explains how to limit syslog messages that are sent to the history table and to SNMP.

This task is optional.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 2	**logging history** *level* **Example:** `Device(config)# logging history 3`	Changes the default level of syslog messages stored in the history file and sent to the SNMP server. By default, **warnings**, **errors**, **critical**, **alerts**, and **emergencies** messages are sent.
Step 3	**logging history size** *number* **Example:**	Specifies the number of syslog messages that can be stored in the history table.

	Command or Action	Purpose
	`Device(config)# logging history size 200`	The default is to store one message. The range is 0 to 500 messages.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Logging Messages to a UNIX Syslog Daemon

This task is optional.

Note Some recent versions of UNIX syslog daemons no longer accept by default syslog packets from the network. If this is the case with your system, use the UNIX **man syslogd** command to decide what options must be added to or removed from the syslog command line to enable logging of remote syslog messages.

Before you begin

- Log in as root.
- Before you can send system log messages to a UNIX syslog server, you must configure the syslog daemon on a UNIX server.

Procedure

	Command or Action	Purpose
Step 1	Add a line to the file /etc/syslog.conf. **Example:** **local7.debug /usr/adm/logs/***cisco.log*	• **local7**—Specifies the logging facility. • **debug**—Specifies the syslog level. The file must already exist, and the syslog daemon must have permission to write to it.

Step 2	Enter these commands at the UNIX shell prompt.
	Example:
	`$ touch /var/log/cisco.log`
	`$ chmod 666 /var/log/cisco.log`
		Creates the log file. The syslog daemon sends messages at this level or at a more severe level to this file.

	Command or Action	Purpose
Step 3	Make sure the syslog daemon reads the new changes. **Example:** `$ kill -HUP `cat /etc/syslog.pid``	For more information, see the **man syslog.conf** and **man syslogd** commands on your UNIX system.

Monitoring and Maintaining System Message Logs

Monitoring Configuration Archive Logs

Command	Purpose
show archive log config {**all** | **number** [*end-number*] | **user** *username* [**session** *number*] *number* [*end-number*] | **statistics**} [**provisioning**]	Displays the entire configuration log or the log for specified parameters.

Configuration Examples for System Message Logs

Example: Switch System Message

This example shows a partial switch system message on a switch:

```
00:00:46: %LINK-3-UPDOWN: Interface Port-channel1, changed state to up
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state to up
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet0/2, changed state to up
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to down
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state
 to down 2
*Mar  1 18:46:11: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
18:47:02: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
*Mar  1 18:48:50.483 UTC: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
```

Additional References for System Message Logs

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	*Command Reference (Catalyst 9600 Series Switches)*

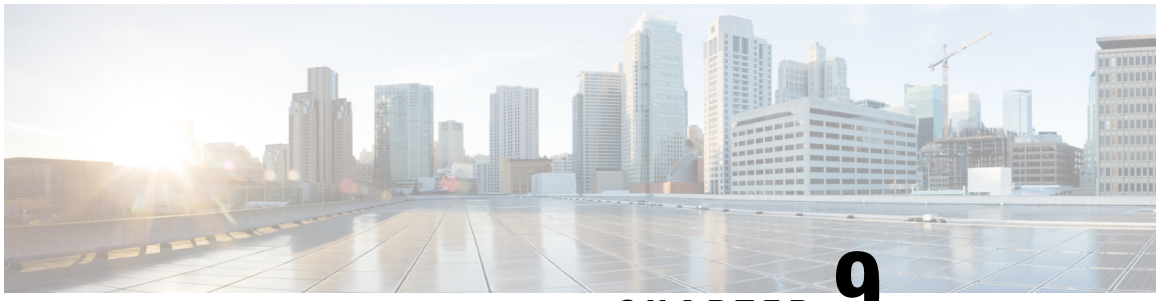
Feature History for System Message Logs

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.11.1	System Message Logs	A switch sends the output from system messages to a logging process. The logging process controls the distribution of logging messages to various destinations, such as the logging buffer, terminal lines, or a UNIX syslog server, depending on your configuration
Cisco IOS XE Cupertino 17.7.1	System Message Logs	Support for this feature was introduced on the Cisco Catalyst 9600 Series Supervisor 2 Module (C9600X-SUP-2.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.



CHAPTER 9

Configuring Online Diagnostics

Information About Configuring Online Diagnostics

With online diagnostics, you can test and verify the hardware functionality of a device while the device is connected to a live network. Online diagnostics contains packet-switching tests that check different hardware components and verify the data path and control signals.

Online diagnostics detects problems in these areas:

- Hardware components
- Interfaces (Ethernet ports and so forth)
- Solder joints

Online diagnostics are categorized as on-demand, scheduled, or health-monitoring diagnostics. On-demand diagnostics run from the CLI; scheduled diagnostics run at user-designated intervals or at specified times when the device is connected to a live network; and health-monitoring runs in the background with user-defined intervals. The health-monitoring test runs every 90, 100, or 150 seconds based on the test.

After you configure online diagnostics, you can manually start diagnostic tests or display the test results. You can also see which tests are configured for the device and the diagnostic tests that have already run.

Generic Online Diagnostics (GOLD) Tests

Note

- Before you enable online diagnostics tests, enable console logging to see all the warning messages.
- While tests are running, all the ports are shut down because a stress test is being performed with looping ports internally, and external traffic might affect the test results. Reboot the switch to bring it to normal operation. When you run the command to reload a switch, the system will ask you if the configuration should be saved. Do not save the configuration.
- If you are running tests on other modules, after a test is initiated and complete, you must reset the module.

The following sections provide information about GOLD tests.

TestGoldPktLoopback

This GOLD packet loopback test verifies the MAC-level loopback functionality. In this test, a GOLD packet, for which Unified Access Data Plane (UADP) ASIC provides support in hardware, is sent. The packet loops back at the MAC-level and is matched against the stored packet.

Attribute	Description
Disruptive or Nondisruptive	Nondisruptive.
Recommendation	Run this on-demand test as per requirement.
Default	Off.
Intitial release	Cisco IOS XE Gibraltar 16.11.1.
Corrective action	Displays a syslog message if the test fails for a port.
Hardware support	All line cards. Not supported on supervisor engines.

TestOBFL

This test verifies the on-board failure logging capabilities. During this test, a diagnostic message is logged to the Onboard Failure Logging (OBFL).

Attribute	Description
Disruptive or Nondisruptive	Nondisruptive.
Recommendation	Run this on-demand test as per requirement.
Default	Off.
Intitial release	Cisco IOS XE Gibraltar 16.11.1.
Corrective action	Displays a syslog message if the test fails for a port.
Hardware support	All line cards and supervisor engines.

TestFantray

This test verifies if a fan tray has been inserted and is working properly on the board. This test runs every 100 seconds.

Attribute	Description
Disruptive or Nondisruptive	Nondisruptive
Recommendation	Do not disable. This can be run as a health-monitoring test and as an on-demand test.
Default	On.
Intitial release	Cisco IOS XE Gibraltar 16.11.1.
Corrective action	Displays a syslog message if the fan tray is not present, or if any of the fans fail.
Hardware support	Only supervisor engines.

TestPhyLoopback

This PHY loopback test verifies the PHY-level loopback functionality. In this test, a packet, which loops back at the PHY level and is matched against the stored packet, is sent. It cannot be run as a health-monitoring test.

Attribute	Description
Disruptive or Nondisruptive	Disruptive.
Recommendation	Run this as an on-demand test as per requirement.
Default	Off.
Intitial release	Cisco IOS XE Gibraltar 17.1.1.
Corrective action	Displays a syslog message if the test fails for any port.
Hardware support	Only on the C9600-LC-48TX line card.

TestThermal

This test verifies the temperature reading from a device sensor if it is below the yellow temperature threshold. This test runs every 90 seconds.

Attribute	Description
Disruptive or Nondisruptive	Nondisruptive
Recommendation	Do not disable. Run this as an on-demand test and a health-monitoring test.
Default	On.
Intitial release	Cisco IOS XE Gibraltar 16.11.1.
Corrective action	Displays a syslog message if the test fails.

Attribute	Description
Hardware support	All line cards and supervisor engines.

TestScratchRegister

This Scratch Register test monitors the health of ASICs by writing values into registers and reading back the values from these registers. This test runs every 90 seconds.

Attribute	Description
Disruptive or Nondisruptive	Nondisruptive.
Recommendation	Do not disable. This can be run as a health-monitoring test and also as an on-demand test.
Default	On.
Intitial release	Cisco IOS XE Gibraltar 16.11.1.
Corrective action	Displays a syslog message if the test fails.
Hardware support	Only supervisor engines.

TestConsistencyCheck

This test checks if the hardware programming is correct. It checks with the forwarding object manager to identify incomplete entries or long-pending configurations to hardware. This test runs every 90 seconds.

Attribute	Description
Disruptive or Nondisruptive	Nondisruptive.
Recommendation	Do not disable. This can be run as a health-monitoring test and also as an on-demand test.
Default	On.
Intitial release	Cisco IOS XE Gibraltar 17.2.1.
Corrective action	Displays a syslog message if the test fails.
Hardware support	Only supervisor engines.

TestPortTxMonitoring

This test monitors the transmit counters of a connected interface. It verifies if a connected port is able to send packets or not. This test runs every 150 seconds.

Attribute	Description
Disruptive or Nondisruptive	Nondisruptive.
Recommendation	Do not disable. This can be run as a health-monitoring test and also as an on-demand test.

Attribute	Description
Default	On.
Intitial release	Cisco IOS XE Gibraltar 16.11.1.
Corrective action	Displays a syslog message if the test fails for a port.
Hardware support	All line cards. Not supported on supervisor engines.

How to Configure Online Diagnostics

The following sections provide information about the various procedures that comprise the online diagnostics configuration.

Starting Online Diagnostic Tests

After you configure diagnostic tests to run on a device, use the **diagnostic start** privileged EXEC command to begin diagnostic testing.

After starting the tests, you cannot stop the testing process midway.

Use the **diagnostic start switch** privileged EXEC command to manually start online diagnostic testing:

Procedure

	Command or Action	Purpose
Step 1	**diagnostic start module** *number* **test** {*name* | *test-id* | *test-id-range* | **all** | **basic** | **complete** | **minimal** | **non-disruptive** | **per-port**} **Example:** `Device# diagnostic start module 2 test basic`	Starts the diagnostic tests. You can specify the tests by using one of these options: • *name*: Enters the name of the test. • *test-id*: Enters the ID number of the test. • *test-id-range*: Enters the range of test IDs by using integers separated by a comma and a hyphen. • **all**: Starts all of the tests. • **basic**: Starts the basic test suite. • **complete**: Starts the complete test suite. • **minimal**: Starts the minimal bootup test suite. • **non-disruptive**: Starts the nondisruptive test suite. • **per-port**: Starts the per-port test suite.

Configuring Online Diagnostics

You must configure the failure threshold and the interval between tests before enabling diagnostic monitoring.

Monitoring and Maintaining Online Diagnostics

You can display the online diagnostic tests that are configured for a device or a device stack and check the test results by using the privileged EXEC **show** commands in this table:

Table 27: Commands for Diagnostic Test Configuration and Results

Command	Purpose
show diagnostic content module [*number* | **all**]	Displays the online diagnostics configured for a switch.
show diagnostic status	Displays the diagnostic tests that are running currently. .
show diagnostic result module [*number* | **all**] [**detail** | **test** {*name* | *test-id* | *test-id-range* | **all**} [**detail**]]	Displays the online diagnostics test results.
show diagnostic post	Displays the POST results. (The output is the same as the **show post** command output.)
show diagnostic events {**event-type** | **module**}	Displays diagnostic events such as error, information, or warning based on the test result.
show diagnostic description module [**number**] **test** { **name** | **test-id** | **all** }	Displays the short description of the results from an individual test or all the tests.

Configuration Examples for Online Diagnostics

The following sections provide examples of online diagnostics configurations.

Examples: Start Diagnostic Tests

This example shows how to start a diagnostic test by using the test name:

```
Device#

diagnostic start module 3 test DiagFanTest
```

This example shows how to start all of the basic diagnostic tests:

```
Device# diagnostic start module 3 test all
```

Example: Displaying Online Diagnostics

This example shows how to display on-demand diagnostic settings:

```
Device# show diagnostic ondemand settings

Test iterations = 1
Action on test failure = continue
```

This example shows how to display diagnostic events for errors:

```
Device# show diagnostic events event-type error

Diagnostic events (storage for 500 events, 0 events recorded)
Number of events matching above criteria = 0

No diagnostic log entry exists.
```

This example shows how to display the description for a diagnostic test:

```
   Device# show diagnostic description module 3 test all
TestGoldPktLoopback :
 The GOLD packet Loopback test verifies the MAC level loopback
 functionality. In this test, a GOLD packet, for which doppler
 provides the support in hardware, is sent. The packet loops back
 at MAC level and is matched against the stored packet. It is a
 non-disruptive test.

TestFantray :
 This test verifies all fan modules have been inserted and working
 properly on the board. It is a non-disruptive test and can be
 run as a health monitoring test.

TestPhyLoopback :
 The PHY Loopback test verifies the PHY level loopback
 functionality. In this test, a packet is sent which loops back
 at PHY level and is matched against the stored packet. It is a
 disruptive test and cannot be run as a health monitoring test.

TestThermal :
 This test verifies the temperature reading from the sensor is
 below the yellow temperature threshold. It is a non-disruptive
 test and can be run as a health monitoring test.

TestScratchRegister :
 The Scratch Register test monitors the health of
 application-specific integrated circuits (ASICs) by writing values
 into registers and reading back the values from these registers.
 It is a non-disruptive test and can be run as a health monitoring
 test.

TestMemory :
 This test runs the exhaustive ASIC memory test during normal
 switch operation. Switch utilizes mbist for this test. Memory test
 is very disruptive in nature and requires switch reboot after
 the test.
```

Additional References for Online Diagnostics

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	*Command Reference (Catalyst 9600 Series Switches)*

Feature History for Configuring Online Diagnostics

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.11.1	Online Diagnostics	With online diagnostics, you can test and verify the hardware functionality of the device while the device is connected to a live network.
Cisco IOS XE Cupertino 17.7.1	Online Diagnostics	Support for this feature was introduced on the Cisco Catalyst 9600 Series Supervisor 2 Module.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.

CHAPTER 10

Consistency Checker

Limitations for Consistency Checker

The Consistency Checker has the following limitations:

- Consistency Checkers are CPU intensive. It is not recommended to run the checkers at very short intervals.
- Legacy Consistency Checkers do not have support for snapshot. So, the previous runs cannot be displayed.
- There is no command to stop/abort the already running Consistency Checkers.
- Forwarding Engine hardware entry validations are partially implemented. Only programming failures can be detected and reported.
- Layer2 MAC Consistency Checker can validate the MAC address in hardware with software copy.
- Consistency checker is designed to reduce false positives in all cases. However, there could be rare cases of reporting a false positive in the following scenarios:
  - Large table state changes (i.e clear, relearn etc).
  - Under very high CPU usage due to any other feature while a consistency checker running. The consistency checker may report inconsistency in processes where CPU usage is high.
- Forwarding engine hardware (FED) check is not entirely supported in Layer3 Multicast Consistency Checker. You can only detect and report on programming failures.
- Forwarding Manager-RP software entry is not supported in Layer3 Multicast Consistency Checker.

Information about Consistency Checker

Overview of Consistency Checker

The Consistency Checker collects information on various table states within the software and the hardware. It compares the software state with the hardware state. If there is any inconsistency, it flags the issue immediately. This helps to reduce increased troubleshooting time at a later period. The consistency checker supplements basic troubleshooting and helps to identify scenarios where inconsistent states between software and hardware tables are causing issues in the network, thereby reducing the mean time to resolve the issue.

There are two types of consistency checker implementation available:

- Legacy Consistency Checker - supports validating the entry from control plane to the forwarding engine (or hardware copy).
- End-to-End Consistency Checker - supports validating the software entry from control plane to all processes involved in distributing and handling the entry, as well as the forwarding engine's hardware copy.

End-to-End Consistency Checker

End-to-End (E2E) Consistency Checker supports full scan and single entry and should be started manually or run via gold diagnostic. The consistency checker can be started for a single entry using the command which helps to isolate the issue at which forwarding process entry is not consistent and helps speed up the debugging.

Every time the consistency checker is started, a runID is provided. Using the runID, its status, summary, details can be viewed. The last 5 snapshots are available any time for you to check the previous run's result.

E2E consistency checker performs the following functions:

- Validates the IOS entry to software tables/processes (Forwarding manger-RP, Forwarding manager-FP and FED) for all modules.
- Reports various inconsistencies (entry inconsistent, entry missing, stale entry) and sends a syslog to alert the administrator.
- Helps to speed up the fault isolation.
- Records any inconsistent entry with relevant data.
- Consistency checker supports the recursive single entry check which can validate the dependent objects along with the actual entry. (i.e, A Layer 3 Multicast with N outgoing interfaces can be validated for multicast entries along with OIFs programming, OIF's Adjacency validation, etc)
- Constant memory usages irrespective of total entries in a table.

Note The consistency checker is bound to CPU utilization and can not exceed the configured value while validating the tables across processes.

Features Supported in Consistency Checker

The following features are supported in consistency checker:

- Legacy Consistency Checker
  - **Layer2 MAC Consistency Checker**: This consistency checker validates the IOS entry to FED software entry. It also validates the MAC address into hardware tables.
  - **Layer3 FMANFP Entry Consistency Checker**: This consistency checker validates the Layer 2, Layer 3, and multicast objects status in the Forwarding Manager-FP process. This includes stale objects and long pending objects.
- E2E Consistency Checker
  - **Layer2 Multicast Consistency Checker**: This consistency checker validates the IOS Layer 2 multicast IGMP/MLD VLAN, the group entry to Forwarding Manager-FP software entry, FED software entry, and FED hardware programming errors.
  - **Layer3 Multicast Consistency Checker**: This consistency checker validates the IOS Layer 3 multicast IGMP/MLD VLAN, the group entry to Forwarding Manager-FP software entry and FED software entry.

Running the Consistency Checker

The table shown below lists the commands to run the various consistency checkers:

Command	Purpose
show consistency-checker l2	Runs the consistency-checker on the Layer 2 forwarding tables.
show consistency-checker l3	Runs the consistency-checker on the Layer 3 forwarding tables.
show consistency-checker mcast l2m	Runs the consistency-checker on the Layer 2 multicast forwarding tables.
show consistency-checker mcast l3m	Runs the consistency-checker on the Layer 3 multicast forwarding tables.
show consistency-checker objects	Runs the End-to-End consistency-checker on objects.
show consistency-checker run-id *run-id*	Runs the End-to-End consistency-checker by run ID.
show consistency-checker switch	Runs the consistency-checker on the specified switch.

Output Examples for Consistency Checker

The following is a sample output for the **show consistency-checker mcast l2m** command where the consistency checker runs a full scan:

```
Device# show consistency-checker mcast l2m start all
L2 multicast Full scan started. Run_id: 2
Use 'show consistency-checker run-id 2 status' for completion status.

Device#
*Feb 17 06:19:14.889: %FED_CCK_ERRMSG-4-INCONSISTENCY_FOUND: F0/0: fed: Consistency
Checker(CCK) detected inconsistency for l2m_vlan. Check 'show consistency run-id 2 detail'.
*Feb 17 06:19:14.890: %FED_CCK_ERRMSG-4-INCONSISTENCY_FOUND: F0/0: fed: Consistency
Checker(CCK) detected inconsistency for l2m_group. Check 'show consistency run-id 2 detail'.
Device#
*Feb 17 06:19:19.432: %IOSXE_FMANRP_CCK-6-FMANRP_COMPLETED: Consistency Check for Run-Id 2
 is completed. Check 'show consistency-checker run-id 2'.
Device#
Device# show consistency-checker run-id 2 status
Process: IOSD
  Object-Type      Status           Time(sec)     Exceptions
  l2m_vlan         Completed        13            No
  l2m_group        Completed        13            No

Process: FMAN-FP
  Object-Type      Status           Time(sec)     State
  l2m_vlan         Completed        9             Consistent
  l2m_group        Completed        9             Consistent

Process: FED
  Object-Type      Status           Time(sec)     State
  l2m_vlan         Completed        9             Inconsistent
  l2m_group        Completed        9             Inconsistent

Device#
Device# show consistency-checker run-id 2
Process: IOSD
  Object-Type    Start-time              Entries       Exceptions
  l2m_vlan       2021/02/17 06:19:05          22                0
  l2m_group      2021/02/17 06:19:05          24                0

Process: FMAN-FP
  *Statistics(A/I/M/S/Oth): Actual/Inherited/Missing/Stale/Others

  Object-Type    Start-time              State             A/  I/  M/  S/Oth
  l2m_vlan       2021/02/17 06:19:05     Consistent        0/  0/  0/  0/  0
  l2m_group      2021/02/17 06:19:05     Consistent        0/  0/  0/  0/  0

Process: FED
  *Statistics(A/I/M/S/HW/Oth): Actual/Inherited/Missing/Stale/Hardware/Others

  Object-Type    Start-time              State              A/  I/  M/  S/ HW/Oth
  l2m_vlan       2021/02/17 06:19:05     Inconsistent       1/  0/  0/168/  0/  0
  l2m_group      2021/02/17 06:19:05     Inconsistent       4/  0/  2/  0/  0/  0


Device#
Device# show consistency-checker run-id 2 detail
Process: IOSD

Process: FMAN-FP

Process: FED
  Object-Type:l2m_vlan   Start-time:2021/02/17 06:19:05
    Status:Completed   State:Inconsistent
    Key/data                                Reason
    (Ipv4, vlan: 768)                       Stale
      snoop:off stp_tcn:off flood:off pimsn:off
    (Ipv4, vlan: 769)                       Stale
```

```
      snoop:off stp_tcn:off flood:off pimsn:off
    (Ipv6, vlan: 900)                       Inconsistent
      snoop:on stp_tcn:on flood:on pimsn:off
    (Ipv6, vlan: 767)                       Stale
      snoop:off stp_tcn:off flood:off pimsn:off

  Object-Type:l2m_group   Start-time:2021/02/17 06:19:05
    Status:Completed   State:Inconsistent
    Key/data                                Reason
    (Ipv4, vlan:100 (*,227.0.0.0))          Inconsistent
      Group ports: total entries: 0
    (Ipv4, vlan:100 (*,227.1.0.0))          Missing

Device#
```

The following is a sample output for the **show consistency-checker mcast l2m** command where the consistency checker runs a recursive single-entry scan:

```
Device# show consistency-checker mcast l2m start vlan 900 229.1.1.1 recursive
Single entry scan started with Run_id: 2

*Feb 17 06:54:09.880: %IOSXE_FMANRP_CCK-6-FMANRP_COMPLETED: Consistency Check for Run-Id 2
 is completed.
Check 'show consistency-checker run-id 2'.
Device#
Device# show consistency-checker run-id 2
Process: IOSD
  Object-Type    Start-time              Entries       Exceptions
  l2m_vlan       2021/02/17 06:54:01           1                0
  l2m_group      2021/02/17 06:54:01           1                0

Process: FMAN-FP
  *Statistics(A/I/M/S/O): Actual/Inherited/Missing/Stale/Others

  Object-Type    Start-time              State            A / I / M / S / O
  l2m_vlan       1970/01/01 00:10:03     Consistent        0/  0/  0/  0/  0
  l2m_group      1970/01/01 00:10:03     Consistent        0/  0/  0/  0/  0


Process: FED
  *Statistics(A/I/M/S/HW/O): Actual/Inherited/Missing/Stale/Hardware/Others

  Object-Type    Start-time              State             A / I / M / S / HW/ O
  l2m_vlan       2021/02/17 06:54:01     Inconsistent       1/  0/  0/  0/  0/  0
  l2m_group      2021/02/17 06:54:01     Inconsistent       0/  1/  0/  0/  0/  0

Device#
Device# show consistency-checker run-id 2 detail
Process: IOSD
  Object-Type:l2m_vlan   Start-time:2021/02/17 06:54:01
    Key/data                                Reason
    (Ipv4, vlan:900)                        Success
      snoop:on stp_tcn:off flood:off pimsn:off

  Object-Type:l2m_group   Start-time:2021/02/17 06:54:01
    Key/data                                Reason
    (Ipv4, vlan:900, (*,229.1.1.1))         Success
      Twe1/0/5


Process: FMAN-FP

Process: FED
  Object-Type:l2m_group   Start-time:2021/02/17 06:54:01
```

```
    Status:Completed   State:Inconsistent
    Key/data                                Reason
    (Ipv4, vlan:900 (*,229.1.1.1))          Inherited
      Group ports: total entries: 1
        TwentyFiveGigE1/0/5

      ---------------Recursion-level-1-----------------
      Object-Type:l2m_vlan   Start-time:2021/02/17 06:54:01
        Status:Completed   State:Inconsistent
        Key/data                                Reason
        (Ipv4, vlan: 900)                       Inconsistent
          snoop:on stp_tcn:off flood:on pimsn:off

Device#
```

The following is a sample output for the **show consistency-checker objects** command where the consistency checker runs a scan on objects:

```
Device# show consistency-checker objects l2m_group
Process: IOSD
  Run-id    Start-time              Exception
  1         2021/02/17 05:20:42     0
  2         2021/02/17 06:19:05     0

Process: FMAN-FP
  *Statistics(A/I/M/S/Oth): Actual/Inherited/Missing/Stale/Others

  Run-id    Start-time              State           A/  I/  M/  S/Oth
  1         2021/02/17 05:20:42     Consistent      0/  0/  0/  0/  0
  2         2021/02/17 06:19:05     Consistent      0/  0/  0/  0/  0

Process: FED
  *Statistics(A/I/M/S/HW/Oth): Actual/Inherited/Missing/Stale/Hardware/Others

  Run-id    Start-time              State           A/  I/  M/  S/ HW/Oth
  1         2021/02/17 05:20:42     Consistent      0/  0/  0/  0/  0/  0
  2         2021/02/17 06:19:05     Inconsistent    4/  0/  2/  0/  0/  0

Device#
Stark#sh consistency-checker run 2 detail
Process: IOSD
  Object-Type:l2m_vlan   Start-time:2021/02/17 06:54:01
    Key/data                                Reason
    (Ipv4, vlan:900)                        Success
      snoop:on stp_tcn:off flood:off pimsn:off

  Object-Type:l2m_group   Start-time:2021/02/17 06:54:01
    Key/data                                Reason
    (Ipv4, vlan:900, (*,229.1.1.1))         Success
      Twe1/0/5


Process: FMAN-FP

Process: FED
  Object-Type:l2m_group   Start-time:2021/02/17 06:54:01
    Status:Completed   State:Inconsistent
    Key/data                                Reason
    (Ipv4, vlan:900 (*,229.1.1.1))          Inherited
      Group ports: total entries: 1
        TwentyFiveGigE1/0/5

      ---------------Recursion-level-1-----------------
      Object-Type:l2m_vlan   Start-time:2021/02/17 06:54:01
```

```
        Status:Completed   State:Inconsistent
        Key/data                                Reason
        (Ipv4, vlan: 900)                       Inconsistent
          snoop:on stp_tcn:off flood:on pimsn:off

Device# show consistency-checker objects l2m_group 2 detail
Process: IOSD

Process: FMAN-FP

Process: FED
  Object-Type:l2m_group   Start-time:2021/02/17 06:19:05
    Status:Completed   State:Inconsistent
    Key/data                                Reason
    (Ipv4, vlan:100 (*,227.0.0.0))          Inconsistent
      Group ports: total entries: 0
    (Ipv4, vlan:100 (*,227.1.0.0))          Missing
    (Ipv4, vlan:100 (*,227.0.0.1))          Inconsistent
      Group ports: total entries: 0
    (Ipv4, vlan:100 (*,227.1.0.1))          Missing
    (Ipv4, vlan:100 (*,227.0.0.2))          Inconsistent
      Group ports: total entries: 0
    (Ipv4, vlan:100 (*,227.0.0.3))          Inconsistent
      Group ports: total entries: 0


Device#
```

The following is a sample output for the **show consistency-checker mcast l3m** command where the consistency checker runs a full scan:

```
Device#sh consistency-checker mcast l3m start all
L3 multicast Full scan started. Run_id: 1
Use 'show consistency-checker run-id 1 status' for completion status.

Device#
*Apr  2 17:30:01.831: %IOSXE_FMANRP_CCK-6-FMANRP_COMPLETED: Consistency Check for Run-Id 1
 is completed. Check 'show consistency-checker run-id 1'.
Device#sh consistency-checker run-id 1
Process: IOSD
Flags:    F - Full Table Scan, S - Single Entry Run
          RE -  Recursive Check, GD -  Garbage Detector
          Hw -  Hardware Check, HS - Hardware Shadow Copy
  Object-Type    Start-time              Entries  Exceptions  Flags
  l3m_entry      2021/04/02 17:29:35           8         0    F GD Hw HS

Process: FMAN-FP
  *Statistics(A/I/M/S/Oth): Actual/Inherited/Missing/Stale/Others

  Object-Type    Start-time              State             A/  I/  M/  S/Oth
  l3m_entry      2021/04/02 17:29:35     Consistent        0/  0/  0/  0/  0

Process: FED
  *Statistics(A/I/M/S/HW/Oth): Actual/Inherited/Missing/Stale/Hardware/Others

  Object-Type    Start-time              State              A/  I/  M/  S/ HW/Oth
  l3m_entry      2021/04/02 17:29:35     Consistent         0/  0/  0/  0/  0/  0

Device#sh consistency-checker mcast l3m start 225.1.1.1 recursive
Single entry scan started with Run_id: 2
Use 'show consistency-checker run-id 2 status' for completion status.

Device#sh consistency-checker run-id 2 status
Process: IOSD
```

```
  Object-Type      Status           Time(sec)     Exceptions
  l2m_vlan         Completed        11            No
  l2m_group        Completed        11            No
  l3m_entry        Completed        11            No

Process: FMAN-FP
  Object-Type      Status           Time(sec)     State
  l2m_vlan         Completed        12            Consistent
  l2m_group        Completed        12            Consistent
  l3m_entry        Completed        12            Consistent

Process: FED
  Object-Type      Status           Time(sec)     State
  l2m_vlan         Completed        12            Consistent
  l2m_group        Completed        12            Consistent
  l3m_entry        Completed        12            Consistent

Device#sh consistency-checker run-id 2 detail
Process: IOSD
  Object-Type:l2m_vlan   Start-time:2021/04/02 17:34:12
    Key/data                                 Reason
    (Ipv4, vlan:100)                         Success
      snoop:on stp_tcn:off flood:off pimsn:off

  Object-Type:l2m_group   Start-time:2021/04/02 17:34:12
    Key/data                                 Reason
    (Ipv4, vlan:100, (*,225.1.1.1))          Success
      Fo1/0/3

  Object-Type:l3m_entry   Start-time:2021/04/02 17:34:12
    Key/data                                 Reason
    (Ipv4, (*,225.1.1.1))                    Success
      Entry flags: C
      Total entries: 1
      Obj_id: F80004A1 Obj_flags: F

Process: FMAN-FP
Process: FED
```

The following is a sample output for the **show consistency-checker mcast l3m** command where the consistency checker runs a recursive single-entry scan:

```
Device#sh consistency-checker mcast l3m start 225.1.1.1 15.1.1.1 recursive
Single entry scan started with Run_id: 4
Use 'show consistency-checker run-id 4 status' for completion status.
Device#sh consistency-checker run-id 4 status
Process: IOSD
  Object-Type      Status           Time(sec)     Exceptions
  l2m_vlan         Completed        10            No
  l2m_group        Completed        10            No
  l3m_entry        Completed        10            No

Process: FMAN-FP
  Object-Type      Status           Time(sec)     State
  l2m_vlan         Completed        11            Consistent
  l2m_group        Completed        11            Consistent
  l3m_entry        Completed        11            Consistent

Process: FED
  Object-Type      Status           Time(sec)     State
  l2m_vlan         Completed        11            Consistent
  l2m_group        Completed        11            Consistent
  l3m_entry        Completed        11            Consistent
Device#sh consistency-checker run-id 4 detail
Process: IOSD
```

```
  Object-Type:l2m_vlan   Start-time:2021/04/02 17:37:36
    Key/data                                Reason
    (Ipv4, vlan:100)                        Success
      snoop:on stp_tcn:off flood:off pimsn:off

  Object-Type:l2m_group   Start-time:2021/04/02 17:37:36
    Key/data                                Reason
    (Ipv4, vlan:100, (*,225.1.1.1))         Success
      Fo1/0/3

  Object-Type:l3m_entry   Start-time:2021/04/02 17:37:36
    Key/data                                Reason
    (Ipv4, vrf:, (15.1.1.1,225.1.1.1))      Success
      Entry flags:
      Total entries: 2
      Obj_id: F80004A1 Obj_flags: F
      Obj_id: F80003C1 Obj_flags: A

Process: FMAN-FP
Process: FED
```

The following is a sample output for the **show diagnostic content** command where end to end consistency is checked through gold diagnostics:

```
Device#show diagnostic content switch all

switch 2  module 1:

  Diagnostics test suite attributes:
    M/C/* - Minimal bootup level test / Complete bootup level test / NA
      B/* - Basic ondemand test / NA
    P/V/* - Per port test / Per device test / NA
    D/N/* - Disruptive test / Non-disruptive test / NA
      S/* - Only applicable to standby unit / NA
      X/* - Not a health monitoring test / NA
      F/* - Fixed monitoring interval test / NA
      E/* - Always enabled monitoring test / NA
      A/I - Monitoring is active / Monitoring is inactive

                                                          Test Interval   Thre-
  ID   Test Name                           Attributes     day hh:mm:ss.ms shold
  ==== ================================== ============   =============== =====
    1) TestGoldPktLoopback -------------> *BPN*X**I      not configured  n/a
    2) TestOBFL ------------------------> *B*N*X**I      not configured  n/a
    3) TestFantray ---------------------> *B*N****A      000 00:01:40.00 1
    4) TestPhyLoopback -----------------> *BPD*X**I      not configured  n/a
    5) TestThermal ---------------------> *B*N****A      000 00:01:30.00 1
    6) TestScratchRegister -------------> *B*N****A      000 00:01:30.00 5
    7) TestPortTxMonitoring ------------> *BPN****A      000 00:02:30.00 1
    8) TestConsistencyCheckL2 ----------> *B*N****A      000 00:01:30.00 1
    9) TestConsistencyCheckL3 ----------> *B*N****A      000 00:01:30.00 1
   10) TestConsistencyCheckMcast -------> *B*N****A      000 00:01:30.00 1
   11) TestConsistencyCheckL2m ---------> *B*N****A      000 00:01:30.00 1
   12) TestConsistencyCheckL3m ---------> *B*N****A      000 00:01:30.00 1
This gives the status of consistency check for multicast
```

Feature History for Consistency Checker

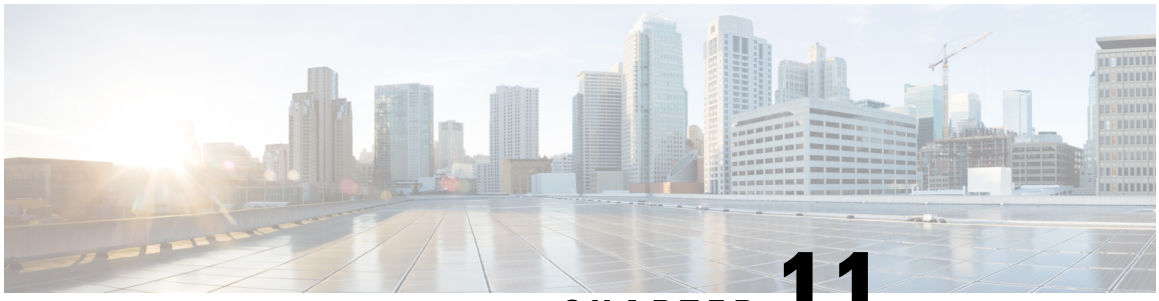
This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Amsterdam 17.3.1	Consistency Checker	The Consistency Checker collects information on various table states within the software and the hardware and flags any inconsistency it finds immediately. It supplements basic troubleshooting and helps to identify scenarios where inconsistent states between software and hardware tables are causing issues in the network, thereby reducing the mean time to resolve the issue.
Cisco IOS XE Bengaluru 17.6.1	Consistency Checker	This feature was enhanced and the multicast consistency checkers were introduced. The following keywords were added to the **show consistency-checker** command: **mcast**, **objects**, and **run-id**.
Cisco IOS XE Cupertino 17.7.1	Consistency Checker	Support for this feature was introduced on the Cisco Catalyst 9600 Series Supervisor 2 Module.

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/

http://www.cisco.com/go/cfn.



CHAPTER 11

Managing Configuration Files

Prerequisites for Managing Configuration Files

- You should have at least a basic familiarity with the Cisco IOS environment and the command-line interface.
- You should have at least a minimal configuration running on your system. You can create a basic configuration file using the **setup** command.

Restrictions for Managing Configuration Files

- Many of the Cisco IOS commands described in this document are available and function only in certain configuration modes on the device.
- Some of the Cisco IOS configuration commands are only available on certain device platforms, and the command syntax may vary on different platforms.

Information About Managing Configuration Files

Types of Configuration Files

Configuration files contain the Cisco IOS software commands used to customize the functionality of your Cisco device. Commands are parsed (translated and executed) by the Cisco IOS software when the system is booted (from the startup-config file) or when you enter commands at the CLI in a configuration mode.

Startup configuration files (startup-config) are used during system startup to configure the software. Running configuration files (running-config) contain the current configuration of the software. The two configuration

files can be different. For example, you may want to change the configuration for a short time period rather than permanently. In this case, you would change the running configuration using the **configure terminal** EXEC command but not save the configuration using the **copy running-config startup-config** EXEC command.

To change the running configuration, use the **configure terminal** command, as described in the Modifying the Configuration File, on page 325 section. As you use the Cisco IOS configuration modes, commands generally are executed immediately and are saved to the running configuration file either immediately after you enter them or when you exit a configuration mode.

To change the startup configuration file, you can either save the running configuration file to the startup configuration using the **copy running-config startup-config** EXEC command or copy a configuration file from a file server to the startup configuration (see the Copying a Configuration File from a TFTP Server to the Device section for more information).

Configuration Mode and Selecting a Configuration Source

To enter configuration mode on the device, enter the **configure** command at the privileged EXEC prompt. The Cisco IOS software responds with the following prompt asking you to specify the terminal, memory, or a file stored on a network server (network) as the source of configuration commands:

```
Configuring from terminal, memory, or network [terminal]?
```

Configuring from the terminal allows you to enter configuration commands at the command line, as described in the following section. See the Re-executing the Configuration Commands in the Startup Configuration File section for more information.

Configuring from the network allows you to load and execute configuration commands over the network. See the Copying a Configuration File from a TFTP Server to the Device section for more information.

Configuration File Changes Using the CLI

The Cisco IOS software accepts one configuration command per line. You can enter as many configuration commands as you want. You can add comments to a configuration file describing the commands you have entered. Precede a comment with an exclamation point (!). Because comments are *not* stored in NVRAM or in the active copy of the configuration file, comments do not appear when you list the active configuration with the **show running-config** or **more system:running-config** EXEC command. Comments are not displayed when you list the startup configuration with the **show startup-config** or **more nvram:startup-config** EXEC mode command. Comments are stripped out of the configuration file when it is loaded onto the device. However, you can list the comments in configuration files stored on a File Transfer Protocol (FTP), Remote Copy Protocol (RCP), or Trivial File Transfer Protocol (TFTP) server. When you configure the software using the CLI, the software executes the commands as you enter them.

Location of Configuration Files

Configuration files are stored in the following locations:

- The running configuration is stored in RAM.
- On all platforms except the Class A Flash file system platforms, the startup configuration is stored in nonvolatile random-access memory (NVRAM).

- On Class A Flash file system platforms, the startup configuration is stored in the location specified by the CONFIG_FILE environment variable (see the Specifying the CONFIG_FILE Environment Variable on Class A Flash File Systems , on page 346 section). The CONFIG_FILE variable defaults to NVRAM and can be a file in the following file systems:
  - **nvram:** (NVRAM)
  - **flash:** (internal flash memory)
  - **usbflash0:** (external usbflash file system)
  - **usbflash1:** (external usbflash file system)

Copy Configuration Files from a Network Server to the Device

You can copy configuration files from a TFTP, rcp, or FTP server to the running configuration or startup configuration of the device. You may want to perform this function for one of the following reasons:

- To restore a backed-up configuration file.
- To use the configuration file for another device. For example, you may add another device to your network and want it to have a similar configuration to the original device. By copying the file to the new device, you can change the relevant parts rather than recreating the whole file.
- To load the same configuration commands on to all of the devices in your network so that all of the devices have similar configurations.

The **copy**{**ftp:** | **rcp:** | **tftp:system:running-config**} EXEC command loads the configuration files into the device as if you were typing the commands on the command line. The device does not erase the existing running configuration before adding the commands. If a command in the copied configuration file replaces a command in the existing configuration file, the existing command is erased. For example, if the copied configuration file contains a different IP address in a particular command than the existing configuration, the IP address in the copied configuration is used. However, some commands in the existing configuration may not be replaced or negated. In this case, the resulting configuration file is a mixture of the existing configuration file and the copied configuration file, with the copied configuration file having precedence.

To restore a configuration file to an exact copy of a file stored on a server, you need to copy the configuration file directly to the startup configuration (using the **copy ftp:**| **rcp:**| **tftp:**} **nvram:startup-config** command) and reload the device.

To copy configuration files from a server to a device, perform the tasks described in the following sections.

The protocol that you use depends on which type of server you are using. The FTP and rcp transport mechanisms provide faster performance and more reliable delivery of data than TFTP. These improvements are possible because the FTP and rcp transport mechanisms are built on and use the TCP/IP stack, which is connection-oriented.

Copying a Configuration File from the Device to a TFTP Server

In some implementations of TFTP, you must create a dummy file on the TFTP server and give it read, write, and execute permissions before copying a file over it. Refer to your TFTP documentation for more information.

Copying a Configuration File from the Device to an RCP Server

You can copy a configuration file from the device to an RCP server.

One of the first attempts to use the network as a resource in the UNIX community resulted in the design and implementation of the remote shell protocol, which included the remote shell (rsh) and remote copy (rcp) functions. Rsh and rcp give users the ability to execute commands remotely and copy files to and from a file system residing on a remote host or server on the network. The Cisco implementation of rsh and rcp interoperates with standard implementations.

The rcp **copy** commands rely on the rsh server (or daemon) on the remote system. To copy files using rcp, you need not create a server for file distribution, as you do with TFTP. You need only to have access to a server that supports the remote shell (rsh). (Most UNIX systems support rsh.) Because you are copying a file from one place to another, you must have read permission on the source file and write permission on the destination file. If the destination file does not exist, rcp creates it for you.

Although the Cisco rcp implementation emulates the functions of the UNIX rcp implementation—copying files among systems on the network—the Cisco command syntax differs from the UNIX rcp command syntax. The Cisco rcp support offers a set of **copy** commands that use rcp as the transport mechanism. These rcp **copy** commands are similar in style to the Cisco TFTP **copy** commands, but they offer an alternative that provides faster performance and reliable delivery of data. These improvements are possible because the rcp transport mechanism is built on and uses the TCP/IP stack, which is connection-oriented. You can use rcp commands to copy system images and configuration files from the device to a network server and vice versa.

You also can enable rcp support to allow users on remote systems to copy files to and from the device.

To configure the Cisco IOS software to allow remote users to copy files to and from the device, use the **ip rcmd rcp-enable** global configuration command.

Restrictions

The RCP protocol requires a client to send a remote username on each RCP request to a server. When you copy a configuration file from the device to a server using RCP, the Cisco IOS software sends the first valid username it encounters in the following sequence:

1. The username specified in the **copy** EXEC command, if a username is specified.
2. The username set by the **ip rcmd remote-username** global configuration command, if the command is configured.
3. The remote username associated with the current tty (terminal) process. For example, if the user is connected to the device through Telnet and was authenticated through the **username** command, the device software sends the Telnet username as the remote username.
4. The device host name.

For the RCP copy request to execute successfully, an account must be defined on the network server for the remote username. If the server has a directory structure, the configuration file or image is written to or copied from the directory associated with the remote username on the server. For example, if the system image resides in the home directory of a user on the server, you can specify that user name as the remote username.

Use the **ip rcmd remote-username** command to specify a username for all copies. (Rcmd is a UNIX routine used at the super-user level to execute commands on a remote machine using an authentication scheme based on reserved port numbers. Rcmd stands for “remote command”). Include the username in the **copy** command if you want to specify a username for that copy operation only.

If you are writing to the server, the RCP server must be properly configured to accept the RCP write request from the user on the device. For UNIX systems, you must add an entry to the .rhosts file for the remote user on the RCP server. For example, suppose the device contains the following configuration lines:

```
hostname Device1
ip rcmd remote-username User0
```

If the device IP address translates to device1.example.com, then the .rhosts file for User0 on the RCP server should contain the following line:

```
Device1.example.com Device1
```

Requirements for the RCP Username

The RCP protocol requires a client to send a remote username on each RCP request to a server. When you copy a configuration file from the device to a server using RCP, the Cisco IOS software sends the first valid username it encounters in the following sequence:

1. The username specified in the **copy** EXEC command, if a username is specified.
2. The username set by the **ip rcmd remote-username** global configuration command, if the command is configured.
3. The remote username associated with the current tty (terminal) process. For example, if the user is connected to the device through Telnet and is authenticated through the **username** command, the device software sends the Telnet username as the remote username.
4. The device host name.

For the RCP copy request to execute, an account must be defined on the network server for the remote username. If the server has a directory structure, the configuration file or image is written to or copied from the directory associated with the remote username on the server. For example, if the system image resides in the home directory of a user on the server, specify that user name as the remote username.

Refer to the documentation for your RCP server for more information.

Copying a Configuration File from the Device to an FTP Server

You can copy a configuration file from the device to an FTP server.

Understanding the FTP Username and Password

Note The password must not contain the special character '@'. If the character '@' is used, the copy fails to parse the IP address of the server.

The FTP protocol requires a client to send a remote username and password on each FTP request to a server. When you copy a configuration file from the device to a server using FTP, the Cisco IOS software sends the first valid username it encounters in the following sequence:

1. The username specified in the **copy** EXEC command, if a username is specified.
2. The username set by the **ip ftp username** global configuration command, if the command is configured.

3. Anonymous.

The device sends the first valid password it encounters in the following sequence:

1. The password specified in the **copy** command, if a password is specified.
2. The password set by the **ip ftp password** command, if the command is configured.
3. The device forms a password *username @devicename.domain* . The variable *username* is the username associated with the current session, *devicename* is the configured host name, and *domain* is the domain of the device.

The username and password must be associated with an account on the FTP server. If you are writing to the server, the FTP server must be properly configured to accept the FTP write request from the user on the device.

If the server has a directory structure, the configuration file or image is written to or copied from the directory associated with the username on the server. For example, if the system image resides in the home directory of a user on the server, specify that user name as the remote username.

Refer to the documentation for your FTP server for more information.

Use the **ip ftp username** and **ip ftp password** global configuration commands to specify a username and password for all copies. Include the username in the **copy** EXEC command if you want to specify a username for that copy operation only.

Copying files through a VRF

You can copy files through a VRF interface specified in the **copy** command. Specifying the VRF in the **copy** command is easier and more efficient as you can directly change the source interface without using a change request for the configuration.

Example

The following example shows how to copy files through a VRF, using the **copy** command:

```
Device#
Address or name of remote host [10.1.2.3]?
Source username [ScpUser]?
Source filename [/auto/tftp-server/ScpUser/vrf_test.txt]?
Destination filename [vrf_test.txt]?
Getting the vrf name as test-vrf
Password:
Sending file modes: C0644 10 vrf_test.txt
!
223 bytes copied in 22.740 secs (10 bytes/sec)
```

Copy Configuration Files from a Switch to Another Switch

You can copy the configurations from one switch to another. This is a 2-step process - Copy the configurations from the switch to the TFTP server, and then from TFTP to another switch.

To copy your current configurations from the switch, run the command **copy startup-config tftp:** and follow the instructions. The configurations are copied onto the TFTP server.

Then, login to another switch and run the command **copy tftp: startup-config** and follow the instructions. The configurations are now copied onto the other switch.

After the configurations are copied, to save your configurations, use **write memory** command and then either reload the switch or run the **copy startup-config running-config** command

Configuration Files Larger than NVRAM

To maintain a configuration file that exceeds the size of NVRAM, you should be aware of the information in the following sections.

Compressing the Configuration File

The **service compress-config** global configuration command specifies that the configuration file be stored compressed in NVRAM. Once the configuration file has been compressed, the device functions normally. When the system is booted, it recognizes that the configuration file is compressed, expands it, and proceeds normally. The **more nvram:startup-config** EXEC command expands the configuration before displaying it.

Before you compress configuration files, refer to the appropriate hardware installation and maintenance publication. Verify that your system's ROMs support file compression. If not, you can install new ROMs that support file compression.

The size of the configuration must not exceed three times the NVRAM size. For a 128-KB size NVRAM, the largest expanded configuration file size is 384 KB.

The **service compress-config** global configuration command works only if you have Cisco IOS software Release 10.0 or later release boot ROMs. Installing new ROMs is a one-time operation and is necessary only if you do not already have Cisco IOS Release 10.0 in ROM. If the boot ROMs do not recognize a compressed configuration, the following message is displayed:

```
Boot ROMs do not support NVRAM compression Config NOT written to NVRAM
```

Storing the Configuration in Flash Memory on Class A Flash File Systems

On class A Flash file system devices, you can store the startup configuration in flash memory by setting the **CONFIG_FILE** environment variable to a file in internal flash memory or flash memory in a PCMCIA slot.

See the Specifying the CONFIG_FILE Environment Variable on Class A Flash File Systems , on page 346 section for more information.

Care must be taken when editing or changing a large configuration. Flash memory space is used every time a **copy system:running-config nvram:startup-config** EXEC command is issued. Because file management for flash memory (such as optimizing free space) is not done automatically, you must pay close attention to available flash memory. Use the **squeeze** command to reclaim used space. We recommend that you use a large-capacity Flash card of at least 20 MB.

Loading the Configuration Commands from the Network

You can also store large configurations on FTP, RCP, or TFTP servers and download them at system startup. To use a network server to store large configurations, see the Copying a Configuration File from the Device to a TFTP Server , on page 326 and Configuring the Device to Download Configuration Files, on page 323 sections for more information on these commands.

Configuring the Device to Download Configuration Files

You can configure the device to load one or two configuration files at system startup. The configuration files are loaded into memory and read in as if you were typing the commands at the command line. Thus, the

configuration for the device is a mixture of the original startup configuration and the one or two downloaded configuration files.

Network Versus Host Configuration Files

For historical reasons, the first file the device downloads is called the network configuration file. The second file the device downloads is called the host configuration file. Two configuration files can be used when all of the devices on a network use many of the same commands. The network configuration file contains the standard commands used to configure all of the devices. The host configuration files contain the commands specific to one particular host. If you are loading two configuration files, the host configuration file should be the configuration file you want to have precedence over the other file. Both the network and host configuration files must reside on a network server reachable via TFTP, RCP, or FTP, and must be readable.

How to Manage Configuration File Information

Displaying Configuration File Information

To display information about configuration files, complete the tasks in this section:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show boot** **Example:** `Device# show boot`	Lists the contents of the BOOT environment variable (if set), the name of the configuration file pointed to by the CONFIG_FILE environment variable, and the contents of the BOOTLDR environment variable.
Step 3	**more** *file-url* **Example:** `Device# more 10.1.1.1`	Displays the contents of a specified file.
Step 4	**show running-config** **Example:** `Device# show running-config`	Displays the contents of the running configuration file. (Command alias for the **more system:running-config** command.)
Step 5	**show startup-config** **Example:** `Device# show startup-config`	Displays the contents of the startup configuration file. (Command alias for the **more nvram:startup-config** command.)

	Command or Action	Purpose
		On all platforms except the Class A Flash file system platforms, the default startup-config file usually is stored in NVRAM. On the Class A Flash file system platforms, the CONFIG_FILE environment variable points to the default startup-config file. The CONFIG_FILE variable defaults to NVRAM.

Modifying the Configuration File

The Cisco IOS software accepts one configuration command per line. You can enter as many configuration commands as you want. You can add comments to a configuration file describing the commands you have entered. Precede a comment with an exclamation point (!). Because comments are *not* stored in NVRAM or in the active copy of the configuration file, comments do not appear when you list the active configuration with the **show running-config** or **more system:running-config** EXEC commands. Comments do not display when you list the startup configuration with the **show startup-config** or **more nvram:startup-config** EXEC mode commands. Comments are stripped out of the configuration file when it is loaded onto the device. However, you can list the comments in configuration files stored on a File Transfer Protocol (FTP), Remote Copy Protocol (RCP), or Trivial File Transfer Protocol (TFTP) server. When you configure the software using the CLI, the software executes the commands as you enter them. To configure the software using the CLI, use the following commands in privileged EXEC mode:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**configuration command** **Example:** `Device(config)# configuration command`	Enter the necessary configuration commands. The Cisco IOS documentation set describes configuration commands organized by technology.
Step 4	Do one of the following: • **end** • **^Z** **Example:**	Ends the configuration session and exits to EXEC mode. **Note** When you press the Ctrl and Z keys simultaneously, ^Z is displayed to the screen.

	Command or Action	Purpose
	`Device(config)# end`	
Step 5	**copy system:running-config nvram:startup-config** **Example:** `Device# copy system:running-config nvram:startup-config`	Saves the running configuration file as the startup configuration file. You may also use the **copy running-config startup-config** command alias, but you should be aware that this command is less precise. On most platforms, this command saves the configuration to NVRAM. On the Class A Flash file system platforms, this step saves the configuration to the location specified by the CONFIG_FILE environment variable (the default CONFIG_FILE variable specifies that the file should be saved to NVRAM).

Examples

In the following example, the device prompt name of the device is configured. The comment line, indicated by the exclamation mark (!), does not execute any command. The **hostname** command is used to change the device name from device to new_name. By pressing Ctrl-Z (^Z) or entering the **end** command, the user quits configuration mode. The **copy system:running-config nvram:startup-config** command saves the current configuration to the startup configuration.

```
Device# configure terminal
Device(config)# !The following command provides the switch host name.
Device(config)# hostname new_name
new_name(config)# end
new_name# copy system:running-config nvram:startup-config
```

When the startup configuration is NVRAM, it stores the current configuration information in text format as configuration commands, recording only non-default settings. The memory is checksummed to guard against corrupted data.

Note Some specific commands might not get saved to NVRAM. You need to enter these commands again if you reboot the machine. These commands are noted in the documentation. We recommend that you keep a list of these settings so that you can quickly reconfigure your device after rebooting.

Copying a Configuration File from the Device to a TFTP Server

To copy configuration information on a TFTP network server, complete the tasks in this section:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**copy system:running-config tftp:** [[[*//location*]*/directory*]*/filename*] **Example:** `Device# copy system:running-config tftp: //server1/topdir/file10`	Copies the running configuration file to a TFTP server.
Step 3	**copy nvram:startup-config tftp:** [[[*//location*]*/directory*]*/filename*] **Example:** `Device# copy nvram:startup-config tftp: //server1/1stdir/file10`	Copies the startup configuration file to a TFTP server.

Examples

The following example copies a configuration file from a device to a TFTP server:

```
Device# copy system:running-config tftp://172.16.2.155/tokyo-confg
Write file tokyo-confg on host 172.16.2.155? [confirm] Y
Writing tokyo-confg!!! [OK]
```

What to Do Next

After you have issued the **copy** command, you may be prompted for additional information or for confirmation of the action. The prompt displayed depends on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

Copying a Configuration File from the Device to an RCP Server

To copy a startup configuration file or a running configuration file from the device to an RCP server, use the following commands beginning in privileged EXEC mode:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:**	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
	`Device> enable`	
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip rcmd remote-username** *username* **Example:** `Device(config)# ip rcmd remote-username NetAdmin1`	(Optional) Changes the default remote username.
Step 4	**end** **Example:** `Device(config)# end`	(Optional) Exits global configuration mode.
Step 5	Do one of the following: • **copy system:running-config rcp:** [[[//[*username@*]*location*]/*directory*]/*filename*] • **copy nvram:startup-config rcp:** [[[//[*username@*]*location*]/*directory*]/*filename*] **Example:** `Device# copy system:running-config rcp: //NetAdmin1@example.com/dir-files/file1`	• Specifies that the device running configuration file is to be stored on an RCP server or • Specifies that the device startup configuration file is to be stored on an RCP server

Examples

Storing a Running Configuration File on an RCP Server

The following example copies the running configuration file named runfile2-confg to the netadmin1 directory on the remote host with an IP address of 172.16.101.101:

```
Device# copy system:running-config rcp://netadmin1@172.16.101.101/runfile2-confg
Write file runfile2-confg on host 172.16.101.101?[confirm]
Building configuration...[OK]
Connected to 172.16.101.101
Device#
```

Storing a Startup Configuration File on an RCP Server

The following example shows how to store a startup configuration file on a server by using RCP to copy the file:

```
Device# configure terminal

Device(config)# ip rcmd remote-username netadmin2

Device(config)# end

Device# copy nvram:startup-config rcp:

Remote host[]? 172.16.101.101

Name of configuration file to write [start-confg]?
Write file start-confg on host 172.16.101.101?[confirm]
![OK]
```

What to Do Next

After you have issued the **copy** EXEC command, you may be prompted for additional information or for confirmation of the action. The prompt displayed depends on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

Copying a Configuration File from the Device to the FTP Server

To copy a startup configuration file or a running configuration file from the device to an FTP server, complete the following tasks:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode on the device.
Step 3	**ip ftp username** *username* **Example:** `Device(config)# ip ftp username NetAdmin1`	(Optional) Specifies the default remote username.
Step 4	**ip ftp password** *password* **Example:** `Device(config)# ip ftp password adminpassword`	(Optional) Specifies the default password.
Step 5	**end** **Example:**	(Optional) Exits global configuration mode. This step is required only if you override the

	Command or Action	Purpose
	`Device(config)# end`	default remote username or password (see Steps 2 and 3).
Step 6	Do one of the following: • **copy system:running-config ftp:** [[[*//*[*username* [**:***password*]**@**]*location*]*/directory*]*/filename*] or • **copy nvram:startup-config ftp:** [[[*//*[*username* [**:***password*]**@**]*location*]*/directory*]*/filename*] **Example:** `Device# copy system:running-config ftp:`	Copies the running configuration or startup configuration file to the specified location on the FTP server.

Examples

Storing a Running Configuration File on an FTP Server

The following example copies the running configuration file named runfile-confg to the netadmin1 directory on the remote host with an IP address of 172.16.101.101:

```
Device# copy system:running-config ftp://netadmin1:mypass@172.16.101.101/runfile-confg
Write file runfile-confg on host 172.16.101.101?[confirm]
Building configuration...[OK]
Connected to 172.16.101.101
Device#
```

Storing a Startup Configuration File on an FTP Server

The following example shows how to store a startup configuration file on a server by using FTP to copy the file:

```
Device# configure terminal

Device(config)# ip ftp username netadmin2

Device(config)# ip ftp password mypass

Device(config)# end

Device# copy nvram:startup-config ftp:

Remote host[]? 172.16.101.101

Name of configuration file to write [start-confg]?
Write file start-confg on host 172.16.101.101?[confirm]
![OK]
```

What to Do Next

After you have issued the **copy** EXEC command, you may be prompted for additional information or for confirmation of the action. The prompt displayed depends on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

Copying a Configuration File from a TFTP Server to the Device

To copy a configuration file from a TFTP server to the device, complete the tasks in this section:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**copy tftp:** [[[*//location*]*/directory*]*/filename*] **system:running-config** **Example:** `Device# copy tftp://server1/dir10/datasource system:running-config`	Copies a configuration file from a TFTP server to the running configuration.
Step 3	**copy tftp:** [[[*//location*]*/directory*]*/filename*] **nvram:startup-config** **Example:** `Device# copy tftp://server1/dir10/datasource nvram:startup-config`	Copies a configuration file from a TFTP server to the startup configuration.
Step 4	**copy tftp:** [[[*//location*]*/directory*]*/filename*] **flash-[n]:/directory/startup-config** **Example:** `Device# copy tftp://server1/dir10/datasource flash:startup-config`	Copies a configuration file from a TFTP server to the startup configuration.

Examples

In the following example, the software is configured from the file named **tokyo-confg** at IP address 172.16.2.155:

```
Device# copy tftp://172.16.2.155/tokyo-confg system:running-config
```

```
Configure using tokyo-confg from 172.16.2.155? [confirm] Y

Booting tokyo-confg from 172.16.2.155:!!! [OK - 874/16000 bytes]
```

What to Do Next

After you have issued the **copy** EXEC command, you may be prompted for additional information or for confirmation of the action. The prompt displayed depends on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

Copying a Configuration File from the rcp Server to the Device

To copy a configuration file from an rcp server to the running configuration or startup configuration, complete the following tasks:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	(Optional) Enters configuration mode from the terminal. This step is required only if you override the default remote username (see Step 3).
Step 3	**ip rcmd remote-username** *username* **Example:** `Device(config)# ip rcmd remote-username NetAdmin1`	(Optional) Specifies the remote username.
Step 4	**end** **Example:** `Device(config)# end`	(Optional) Exits global configuration mode. This step is required only if you override the default remote username (see Step 2).
Step 5	Do one of the following: • **copy rcp://***[username@]location*`/`*[directory/]filename* **system:running-config** • **copy rcp://***[username@]location*`/`*[directory/]filename* **nvram:startup-config** **Example:** `Device# copy`	Copies the configuration file from an rcp server to the running configuration or startup configuration.

	Command or Action	Purpose
	`rcp://[user1@example.com/dir10/fileone] nvram:startup-config`	

Examples

Copy RCP Running-Config

The following example copies a configuration file named host1-confg from the netadmin1 directory on the remote server with an IP address of 172.16.101.101, and loads and runs the commands on the device:

```
device# copy rcp://netadmin1@172.16.101.101/host1-confg system:running-config
Configure using host1-confg from 172.16.101.101? [confirm]
Connected to 172.16.101.101
Loading 1112 byte file host1-confg:![OK]
device#
%SYS-5-CONFIG: Configured from host1-config by rcp from 172.16.101.101
```

Copy RCP Startup-Config

The following example specifies a remote username of netadmin1. Then it copies the configuration file named host2-confg from the netadmin1 directory on the remote server with an IP address of 172.16.101.101 to the startup configuration.

```
device# configure terminal
device(config)# ip rcmd remote-username netadmin1
device(config)# end
device# copy rcp: nvram:startup-config
Address of remote host [255.255.255.255]? 172.16.101.101
Name of configuration file[rtr2-confg]? host2-confg
Configure using host2-confg from 172.16.101.101?[confirm]
Connected to 172.16.101.101
Loading 1112 byte file host2-confg:![OK]
[OK]
device#
%SYS-5-CONFIG_NV:Non-volatile store configured from host2-config by rcp from 172.16.101.101
```

What to Do Next

After you have issued the **copy** EXEC command, you may be prompted for additional information or for confirmation of the action. The prompt displayed depends on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

Copying a Configuration File from an FTP Server to the Device

To copy a configuration file from an FTP server to the running configuration or startup configuration, complete the tasks in this section:

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	(Optional) Allows you to enter global configuration mode. This step is required only if you want to override the default remote username or password (see Steps 3 and 4).
Step 3	**ip ftp username** *username* **Example:** `Device(config)# ip ftp username NetAdmin1`	(Optional) Specifies the default remote username.
Step 4	**ip ftp password** *password* **Example:** `Device(config)# ip ftp password adminpassword`	(Optional) Specifies the default password.
Step 5	**end** **Example:** `Device(config)# end`	(Optional) Exits global configuration mode. This step is required only if you override the default remote username or password (see Steps 3 and 4).
Step 6	Do one of the following: • **copy ftp:** [[[//[*username*[**:***password*]**@**]*location*] **/***directory*]/*filename*]**system:running-config** • **copy ftp:** [[[*/[username[:password]@]location]/directory/]filename]nvram:startup-config** **Example:** `Device# copy ftp:nvram:startup-config`	Using FTP copies the configuration file from a network server to running memory or the startup configuration.

Examples

Copy FTP Running-Config

The following example copies a host configuration file named host1-confg from the netadmin1 directory on the remote server with an IP address of 172.16.101.101, and loads and runs the commands on the device:

```
device# copy ftp://netadmin1:mypass@172.16.101.101/host1-confg system:running-config
Configure using host1-confg from 172.16.101.101? [confirm]
Connected to 172.16.101.101
Loading 1112 byte file host1-confg:![OK]
```

```
device#
%SYS-5-CONFIG: Configured from host1-config by ftp from 172.16.101.101
```

Copy FTP Startup-Config

The following example specifies a remote username of netadmin1. Then it copies the configuration file named host2-confg from the netadmin1 directory on the remote server with an IP address of 172.16.101.101 to the startup configuration:

```
device# configure terminal
device(config)# ip ftp username netadmin1
device(config)# ip ftp password mypass
device(config)# end
device# copy ftp: nvram:startup-config
Address of remote host [255.255.255.255]? 172.16.101.101
Name of configuration file[host1-confg]? host2-confg
Configure using host2-confg from 172.16.101.101?[confirm]
Connected to 172.16.101.101
Loading 1112 byte file host2-confg:![OK]
[OK]
device#
%SYS-5-CONFIG_NV:Non-volatile store configured from host2-config by ftp from 172.16.101.101
```

What to Do Next

After you have issued the **copy** EXEC command, you may be prompted for additional information or for confirmation of the action. The prompt displayed depends on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

Maintaining Configuration Files Larger than NVRAM

To maintain a configuration file that exceeds the size of NVRAM, perform the tasks described in the following sections:

Compressing the Configuration File

To compress configuration files, complete the tasks in this section:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**service compress-config** **Example:** `Device(config)# service compress-config`	Specifies that the configuration file be compressed.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode.
Step 5	Do one of the following: • Use FTP, RCP, or TFTP to copy the new configuration. • **configure terminal** **Example:** `Device# configure terminal`	Enters the new configuration: • If you try to load a configuration that is more than three times larger than the NVRAM size, the following error message is displayed: "[buffer overflow - *file-size* /*buffer-size* bytes]."
Step 6	**copy system:running-config nvram:startup-config** **Example:** `Device(config)# copy system:running-config nvram:startup-config`	When you have finished changing the running-configuration, save the new configuration.

Examples

The following example compresses a 129-KB configuration file to 11 KB:

```
Device# configure terminal

Device(config)# service compress-config

Device(config)# end

Device# copy tftp://172.16.2.15/tokyo-confg system:running-config

Configure using tokyo-confg from 172.16.2.155? [confirm] y

Booting tokyo-confg from 172.16.2.155:!!! [OK - 874/16000 bytes]
Device# copy system:running-config nvram:startup-config

Building configuration...
Compressing configuration from 129648 bytes to 11077 bytes
[OK]
```

Storing the Configuration in Flash Memory on Class A Flash File Systems

To store the startup configuration in flash memory, complete the tasks in this section:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**copy nvram:startup-config** *flash-filesystem:filename* **Example:** `Device# copy nvram:startup-config usbflash0:switch-config`	Copies the current startup configuration to the new location to create the configuration file.
Step 3	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 4	**boot config flash-filesystem:** *filename* **Example:** `Device(config)# boot config usbflash0:switch-config`	Specifies that the startup configuration file be stored in flash memory by setting the CONFIG_FILE variable.
Step 5	**end** **Example:** `Device(config)# end`	Exits global configuration mode.
Step 6	Do one of the following: • Use FTP, RCP, or TFTP to copy the new configuration. If you try to load a configuration that is more than three times larger than the NVRAM size, the following error message is displayed: "[buffer overflow - *file-size* /*buffer-size* bytes]. " • **configure terminal** **Example:** `Device# configure terminal`	Enters the new configuration.

	Command or Action	Purpose
Step 7	**copy system:running-config nvram:startup-config** **Example:** `Device(config)# copy system:running-config nvram:startup-config`	When you have finished changing the running-configuration, save the new configuration.

Examples

The following example stores the configuration file in usbflash0:

```
Device# copy nvram:startup-config usbflash0:switch-config
Device# configure terminal
Device(config)# boot config usbflash0:switch-config
Device(config)# end
Device# copy system:running-config nvram:startup-config
```

Loading the Configuration Commands from the Network

To use a network server to store large configurations, complete the tasks in this section:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**copy system:running-config** {**ftp:** | **rcp:** | **tftp:**} **Example:** `Device# copy system:running-config ftp:`	Saves the running configuration to an FTP, RCP, or TFTP server.
Step 3	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 4	**boot network** {**ftp:**[[[//[*username* [**:***password*]**@**]*location*]*/directory*]*/filename*] | **rcp:**[[[//[*username***@**]*location*]*/directory*	Specifies that the startup configuration file be loaded from the network server at startup.

	Command or Action	Purpose
	]/*filename*] | **tftp:**[[[//*location*]/*directory*]/*filename*]} **Example:** `Device(config)# boot network ftp://user1:guessme@example.com/dir10/file1`	
Step 5	**service config** **Example:** `Device(config)# service config`	Enables the switch to download configuration files at system startup.
Step 6	**end** **Example:** `Device(config)# end`	Exits global configuration mode.
Step 7	**copy system:running-config nvram:startup-config** **Example:** `Device# copy system:running-config nvram:startup-config`	Saves the configuration.

Copying Configuration Files from Flash Memory to the Startup or Running Configuration

To copy a configuration file from flash memory directly to your startup configuration in NVRAM or your running configuration, enter one of the commands in Step 2:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	Do one of the following: • **copy** *filesystem*: [*partition-number***:**][*filename*] **nvram:startup-config** • **copy** *filesystem*: [*partition-number***:**][*filename*] **system:running-config**	• Loads a configuration file directly into NVRAM or • Copies a configuration file to your running configuration

	Command or Action	Purpose
	Example: `Device# copy usbflash0:4:ios-upgrade-1 nvram:startup-config`	

Examples

The following example copies the file named ios-upgrade-1 from partition 4 of the flash memory PC Card in usbflash0 to the device startup configurations:

```
Device# copy usbflash0:4:ios-upgrade-1 nvram:startup-config

Copy 'ios-upgrade-1' from flash device as 'startup-config' ? [yes/no] yes

[OK]
```

Copying Configuration Files Between Flash Memory File Systems

On platforms with multiple flash memory file systems, you can copy files from one flash memory file system, such as internal flash memory to another flash memory file system. Copying files to different flash memory file systems lets you create backup copies of working configurations and duplicate configurations for other devices. To copy a configuration file between flash memory file systems, use the following commands in EXEC mode:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show** *source-filesystem*: **Example:** `Device# show flash:`	Displays the layout and contents of flash memory to verify the filename.
Step 3	**copy** *source-filesystem*:[*partition-number***:**][*filename*] *dest-filesystem***:**[*partition-number***:**][*filename*] **Example:** `Device# copy flash: usbflash0:`	Copies a configuration file between flash memory devices. • The source device and the destination device cannot be the same. For example, the **copy usbflash0: usbflash0:** command is invalid.

Example

The following example copies the file named running-config from partition 1 on internal flash memory to partition 1 of usbflash0 on a device. In this example, the source partition is not specified, so the device prompts for the partition number:

```
Device# copy flash: usbflash0:

System flash
Partition   Size    Used      Free      Bank-Size  State          Copy Mode
  1         4096K   3070K     1025K     4096K      Read/Write     Direct
  2        16384K   1671K    14712K     8192K      Read/Write     Direct
[Type ?<no> for partition directory; ? for full directory; q to abort]
Which partition? [default = 1]
System flash directory, partition 1:
File  Length   Name/status
  1   3142748  dirt/network/mars-test/c3600-j-mz.latest
  2   850      running-config
[3143728 bytes used, 1050576 available, 4194304 total]
usbflash0 flash directory:
File  Length   Name/status
  1   1711088  dirt/gate/c3600-i-mz
  2   850      running-config
[1712068 bytes used, 2482236 available, 4194304 total]
Source file name? running-config

Destination file name [running-config]?
Verifying checksum for 'running-config' (file # 2)...  OK
Erase flash device before writing? [confirm]
Flash contains files. Are you sure you want to erase? [confirm]
Copy 'running-config' from flash: device
  as 'running-config' into usbflash0: device WITH erase? [yes/no] yes

Erasing device... eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee ...erased!

 [OK - 850/4194304 bytes]
Flash device copy took 00:00:30 [hh:mm:ss]
Verifying checksum...  OK (0x16)
```

Copying a Configuration File from an FTP Server to Flash Memory Devices

To copy a configuration file from an FTP server to a flash memory device, complete the task in this section:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	(Optional) Enters global configuration mode. This step is required only if you override the default remote username or password (see Steps 3 and 4).

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**ip ftp username** *username* **Example:** `Device(config)# ip ftp username Admin01`	(Optional) Specifies the remote username.
Step 4	**ip ftp password** *password* **Example:** `Device(config)# ip ftp password adminpassword`	(Optional) Specifies the remote password.
Step 5	**end** **Example:** `Device(config)# end`	(Optional) Exits configuration mode. This step is required only if you override the default remote username (see Steps 3 and 4).
Step 6	**copy ftp:** [[*//location*]*/directory*]*/bundle_name* **flash:** **Example:** `Device>copy ftp:/cat9k_iosxe.16.11.01.SPA.bin flash:`	Copies the configuration file from a network server to the flash memory device using FTP.

What to Do Next

After you have issued the **copy** EXEC command, you may be prompted for additional information or for confirmation of the action. The prompt displayed depends on how much information you provide in the **copy** command and the current setting of the **file prompt** global configuration command.

Copying a Configuration File from an RCP Server to Flash Memory Devices

To copy a configuration file from an RCP server to a flash memory device, complete the tasks in this section:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	(Optional) Enters global configuration mode. This step is required only if you override the

	Command or Action	Purpose
	`Device# configure terminal`	default remote username or password (see Step 3).
Step 3	**ip rcmd remote-username** *username* **Example:** `Device(config)# ip rcmd remote-username Admin01`	(Optional) Specifies the remote username.
Step 4	**end** **Example:** `Device(config)# end`	(Optional) Exits configuration mode. This step is required only if you override the default remote username or password (see Step 3).
Step 5	**copy rcp:** [[[//[*username@*]*location*]/*directory*] /*bundle_name*] **flash:** **Example:** `Device# copy rcp://netadmin@172.16.101.101/bundle1 flash:`	Copies the configuration file from a network server to the flash memory device using RCP. Respond to any device prompts for additional information or confirmation. Prompting depends on how much information you provide in the **copy** command and the current setting of the **file prompt** command.

Copying a Configuration File from a TFTP Server to Flash Memory Devices

To copy a configuration file from a TFTP server to a flash memory device, complete the tasks in this section:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**copy tftp:** [[[//*location*]/*directory*]/bundle_name **flash:** **Example:** `Device# copy tftp://cat3k_caa-universalk9.SSA.03.12.02.EZP.150-12.02.EZP.150-12.02.EZP.bin flash:`	Copies the file from a TFTP server to the flash memory device. Reply to any device prompts for additional information or confirmation. Prompting depends on how much information you provide in the **copy** command and the current setting of the **file prompt** command.

Examples

The following example shows the copying of the configuration file named switch-config from a TFTP server to the flash memory card inserted in usbflash0. The copied file is renamed new-config.

```
Device#
copy tftp:switch-config usbflash0:new-config
```

Re-executing the Configuration Commands in the Startup Configuration File

To re-execute the commands located in the startup configuration file, complete the task in this section:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure memory** **Example:** `Device# configure memory`	Re-executes the configuration commands located in the startup configuration file.

Clearing the Startup Configuration

You can clear the configuration information from the startup configuration. If you reboot the device with no startup configuration, the device enters the Setup command facility so that you can configure the device from scratch. To clear the contents of your startup configuration, complete the task in this section:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**erase nvram** **Example:**	Clears the contents of your startup configuration.

	Command or Action	Purpose
	`Device# erase nvram`	**Note** For all platforms except the Class A Flash file system platforms, this command erases NVRAM. The startup configuration file cannot be restored once it has been deleted. On Class A Flash file system platforms, when you use the **erase startup-config** EXEC command, the device erases or deletes the configuration pointed to by the CONFIG_FILE environment variable. If this variable points to NVRAM, the device erases NVRAM. If the CONFIG_FILE environment variable specifies a flash memory device and configuration filename, the device deletes the configuration file. That is, the device marks the file as "deleted," rather than erasing it. This feature allows you to recover a deleted file.

Deleting a Specified Configuration File

To delete a specified configuration on a specific flash device, complete the task in this section:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**delete** *flash-filesystem*:*filename* **Example:**	Deletes the specified configuration file on the specified flash device.

	Command or Action	Purpose
	`Device# delete usbflash0:myconfig`	**Note** On Class A and B Flash file systems, when you delete a specific file in flash memory, the system marks the file as deleted, allowing you to later recover a deleted file using the **undelete** EXEC command. Erased files cannot be recovered. To permanently erase the configuration file, use the **squeeze** EXEC command. On Class C Flash file systems, you cannot recover a file that has been deleted. If you attempt to erase or delete the configuration file specified by the CONFIG_FILE environment variable, the system prompts you to confirm the deletion.

Specifying the CONFIG_FILE Environment Variable on Class A Flash File Systems

On Class A flash file systems, you can configure the Cisco IOS software to load the startup configuration file specified by the CONFIG_FILE environment variable. The CONFIG_FILE variable defaults to NVRAM. To change the CONFIG_FILE environment variable, complete the tasks in this section:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**copy** [*flash-url* | *ftp-url* | *rcp-url* | *tftp-url* | **system:running-config** | **nvram:startup-config**] *dest-flash-url* **Example:** `Device# copy system:running-config nvram:startup-config`	Copies the configuration file to the flash file system from which the device loads the file on restart.
Step 3	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 4	**boot config** *dest-flash-url* **Example:** `Device(config)# boot config 172.16.1.1`	Sets the CONFIG_FILE environment variable. This step modifies the runtime CONFIG_FILE environment variable.
Step 5	**end** **Example:** `Device(config)# end`	Exits global configuration mode.
Step 6	**copy system:running-config nvram:startup-config** **Example:** `Device# copy system:running-config nvram:startup-config`	Saves the configuration performed in Step 3 to the startup configuration.
Step 7	**show boot** **Example:** `Device# show boot`	(Optional) Allows you to verify the contents of the CONFIG_FILE environment variable.

Examples

The following example copies the running configuration file to the device. This configuration is then used as the startup configuration when the system is restarted:

```
Device# copy system:running-config usbflash0:config2
Device# configure terminal
Device(config)# boot config usbflash0:config2
Device(config)# end
Device# copy system:running-config nvram:startup-config
[ok]
Device# show boot
BOOT variable = usbflash0:rsp-boot-m
CONFIG_FILE variable = nvram:
Current CONFIG_FILE variable = usbflash0:config2
Configuration register is 0x010F
```

What to Do Next

After you specify a location for the startup configuration file, the **nvram:startup-config** command is aliased to the new location of the startup configuration file. The **more nvram:startup-config** EXEC command displays the startup configuration, regardless of its location. The **erase nvram:startup-config** EXEC command erases the contents of NVRAM and deletes the file pointed to by the CONFIG_FILE environment variable.

When you save the configuration using the **copy system:running-config nvram:startup-config** command, the device saves a complete version of the configuration file to the location specified by the CONFIG_FILE environment variable and a distilled version to NVRAM. A distilled version is one that does not contain access list information. If NVRAM contains a complete configuration file, the device prompts you to confirm your overwrite of the complete version with the distilled version. If NVRAM contains a distilled configuration, the device does not prompt you for confirmation and proceeds with overwriting the existing distilled configuration file in NVRAM.

Note If you specify a file in a flash device as the CONFIG_FILE environment variable, every time you save your configuration file with the **copy system:running-config nvram:startup-config** command, the old configuration file is marked as "deleted," and the new configuration file is saved to that device. Eventually, Flash memory fills up as the old configuration files still take up memory. Use the **squeeze** EXEC command to permanently delete the old configuration files and reclaim the space.

Configuring the Device to Download Configuration Files

You can specify an ordered list of network configuration and host configuration filenames. The Cisco IOS XE software scans this list until it loads the appropriate network or host configuration file.

To configure the device to download configuration files at system startup, perform at least one of the tasks described in the following sections:

- Configuring the Device to Download the Network Configuration File
- Configuring the Device to Download the Host Configuration File

If the device fails to load a configuration file during startup, it tries again every 10 minutes (the default setting) until a host provides the requested files. With each failed attempt, the device displays the following message on the console terminal:

```
Booting host-confg... [timed out]
```

If there are any problems with the startup configuration file, or if the configuration register is set to ignore NVRAM, the device enters the Setup command facility.

Configuring the Device to Download the Network Configuration File

To configure the Cisco IOS software to download a network configuration file from a server at startup, complete the tasks in this section:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**boot network** {**ftp:**[[[//[*username* [**:***password*]**@**]*location*]/*directory*]/*filename*] | **rcp:**[[[//[*username***@**]*location*]/*directory*]/*filename*] | **tftp:**[[[//*location*]/*directory*]/*filename*]} **Example:** `Device(config)# boot network tftp:hostfile1`	Specifies the network configuration file to download at startup, and the protocol to be used (TFTP, RCP, or FTP). • If you do not specify a network configuration filename, the Cisco IOS software uses the default filename network-confg. If you omit the address, the device uses the broadcast address. • You can specify more than one network configuration file. The software tries them in order entered until it loads one. This procedure can be useful for keeping files with different configuration information loaded on a network server.
Step 4	**service config** **Example:** `Device(config)# service config`	Enables the system to automatically load the network file on restart.
Step 5	**end** **Example:** `Device(config)# end`	Exits global configuration mode.
Step 6	**copy system:running-config nvram:startup-config** **Example:** `Device# copy system:running-config nvram:startup-config`	Saves the running configuration to the startup configuration file.

Configuring the Device to Download the Host Configuration File

To configure the Cisco IOS software to download a host configuration file from a server at startup, complete the tasks in this section:

Procedure

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**boot host** {**ftp:**[[[//[*username* [**:***password*]**@**]*location*]/*directory*]/*filename*] | **rcp:**[[[//[*username***@**]*location*]/*directory*]/*filename*] | **tftp:**[[[//*location*]/*directory*]/*filename*] } **Example:** `Device(config)# boot host tftp:hostfile1`	Specifies the host configuration file to download at startup, and the protocol to be used (FTP, RCP, or TFTP): • If you do not specify a host configuration filename, the device uses its own name to form a host configuration filename by converting the name to all lowercase letters, removing all domain information, and appending "-confg." If no host name information is available, the software uses the default host configuration filename device-confg. If you omit the address, the device uses the broadcast address. • You can specify more than one host configuration file. The Cisco IOS software tries them in order entered until it loads one. This procedure can be useful for keeping files with different configuration information loaded on a network server.
Step 4	**service config** **Example:** `Device(config)# service config`	Enables the system to automatically load the host file upon restart.
Step 5	**end** **Example:** `Device(config)# end`	Exits global configuration mode.
Step 6	**copy system:running-config nvram:startup-config** **Example:** `Device# copy system:running-config nvram:startup-config`	Saves the running configuration to the startup configuration file.

Example

In the following example, a device is configured to download the host configuration file named hostfile1 and the network configuration file named networkfile1. The device uses TFTP and the broadcast address to obtain the file:

```
Device# configure terminal
Device(config)# boot host tftp:hostfile1
Device(config)# boot network tftp:networkfile1
Device(config)# service config
Device(config)# end
Device# copy system:running-config nvram:startup-config
```

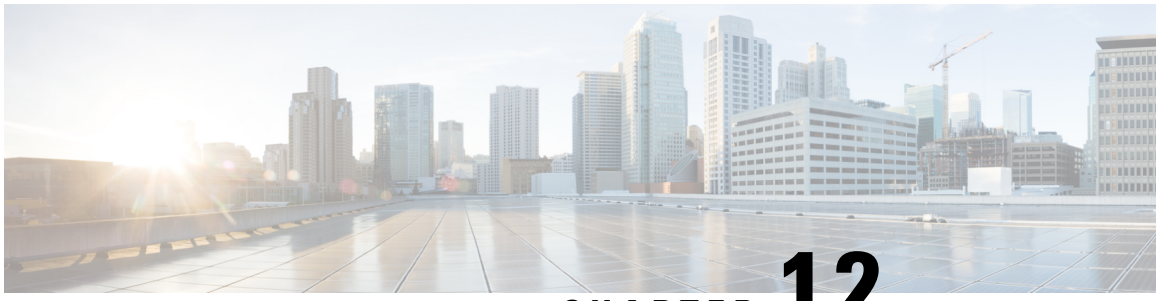
Feature History for Managing Configuration Files

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.11.1	Managing Configuration Files	Configuration files contain the Cisco IOS software commands used to customize the functionality of your Cisco device. Commands are parsed (translated and executed) by the Cisco IOS software when the system is booted (from the startup-config file) or when you enter commands at the CLI in a configuration mode.
Cisco IOS XE Cupertino 17.7.1	Managing Configuration Files	Support for this feature was introduced on the Cisco Catalyst 9600 Series Supervisor 2 Module.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.



CHAPTER 12

Secure Copy

This document provides the procedure to configure a Cisco device for Secure Copy (SCP) server-side functionality.

Prerequisites for Secure Copy

- Configure Secure Shell (SSH), authentication, and authorization on the device.
- Because the Secure Copy Protocol (SCP) relies on SSH for its secure transport, the device must have a Rivest, Shamir, and Adelman (RSA) key pair.

Information About Secure Copy

The Secure Copy feature provides a secure and authenticated method for copying switch configurations or switch image files. The Secure Copy Protocol (SCP) relies on Secure Shell (SSH), an application and a protocol that provides a secure replacement for the Berkeley r-tools.

The behavior of SCP is similar to that of Remote Copy Protocol (RCP), which comes from the Berkeley r-tools suite (Berkeley university's own set of networking applications), except that SCP relies on SSH for security. In addition, SCP requires authentication, authorization, and accounting (AAA) to be configured to ensure that the device can determine whether a user has the correct privilege level.

SCP allows only users with a privilege level of 15 to copy a file in the Cisco IOS File System (Cisco IFS) to and from a device by using the **copy** command. An authorized administrator can also perform this action from a workstation.

Note

- Enable the SCP option while using the `pscp.exe` file.
- An RSA public-private key pair must be configured on the device for SSH to work.

Similar to SCP, SSH File Transfer Protocol (SFTP) can be used to copy switch configuration or image files. For more information, refer the *Configuring SSH File Transfer Protocol* chapter of the *Security Configuration Guide*.

Secure Copy Performance Improvements

SSH bulk data transfer mode can be used to enhance the throughput performance of SCP that is operating in the capacity of a client or a server. Beginning from Cisco IOS XE Dublin 17.10.1, SSH bulk data transfer mode is enabled by default with default window size of 128KB. TCP selective acknowledgement (SACK) is enabled by default if the bulk mode window size is configured.

The default bulk mode window size of 128 KB is optimal to copy large files in most network settings. However, in long big networks where the round-trip time (RTT) is high, 128 KB is not enough. You can enable the most optimal SCP throughput performance by configuring the bulk mode window size using the **ip ssh bulk-mode** *window-size* command. For example, in an ideal lab testing environment, a window size of 2 MB in a 200-milliseconds round-trip time setting can give around 500 percent improved throughput performance when compared to the default 128-KB window size.

The bulk mode window size must be configured as per the network bandwidth-delay product, that is, a multiple of total available bandwidth in bits per second and the round-trip time in seconds. Because the CPU usage may increase with the increased window size, make sure to balance this by choosing the right window size.

How to Configure Secure Copy

The following sections provide information about the Secure Copy configuration tasks.

Configuring Secure Copy

To configure a Cisco device for SCP server-side functionality, perform the following steps.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Sets AAA authentication at login.
Step 4	**aaa authentication login** {**default** | *list-name*} *method1* [*method2...*] **Example:** `Device(config)# aaa authentication login default group tacacs+`	Enables the AAA access control system.
Step 5	**username** *name* [**privilege** *level*] **password** *encryption-type encrypted-password* **Example:** `Device(config)# username superuser privilege 2 password 0 superpassword`	Establishes a username-based authentication system. **Note** You can omit this step if a network-based authentication mechanism, such as TACACS+ or RADIUS, has been configured.
Step 6	**ip scp server enable** **Example:** `Device(config)# ip scp server enable`	Enables SCP server-side functionality.
Step 7	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 8	**debug ip scp** **Example:** `Device# debug ip scp`	(Optional) Troubleshoots SCP authentication problems.

Configuring SCP Username Password

To configure a username and password for SCP, perform the following steps:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip scp username***username* **Example:** `Device# ip scp username cisco`	Defines the username.
Step 4	**ip scp password***password* **Example:** `Device# ip scp password 0 cisco`	Defines the password. Specify the encryption level. • 0 – Unencrypted password. • 0 – Encrypted password. • Line – Clear text password.
Step 5	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Enabling Secure Copy on the SSH Server

The following task shows how to configure the server-side functionality for SCP. This task shows a typical configuration that allows a device to securely copy files from a remote workstation.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables the Authentication, Authorization, and Accounting (AAA) access control model.
Step 4	**aaa authentication login default local** **Example:** `Device(config)# aaa authentication login default local`	Sets AAA authentication to use the local username database for authentication at login.

	Command or Action	Purpose
Step 5	**aaa authorization exec default local** **Example:** `Device(config)# aaa authorization exec default local`	Sets the parameters that restrict user access to a network, runs the authorization to determine if the user ID is allowed to run an privileged EXEC shell, and specifies that the system must use the local database for authorization.
Step 6	**username** *name* **privilege** *privilege-level* **password** *password* **Example:** `Device(config)# username samplename privilege 15 password password1`	Establishes a username-based authentication system, and specifies the username, privilege level, and an unencrypted password. **Note** The minimum required value for the *privilege-level* argument is 15. A privilege level of less than 15 results in the connection closing.
Step 7	**ip ssh time-out** *seconds* **Example:** `Device(config)# ip ssh time-out 120`	Sets the time interval (in seconds) that the device waits for the SSH client to respond.
Step 8	**ip ssh authentication-retries** *integer* **Example:** `Device(config)# ip ssh authentication-retries 3`	Sets the number of authentication attempts after which the interface is reset.
Step 9	**ip scp server enable** **Example:** `Device(config)# ip scp server enable`	Enables the device to securely copy files from a remote workstation.
Step 10	**ip ssh bulk-mode** *window-size* **Example:** `Device(config)# ip ssh bulk-mode 33107232`	(Optional) Sets the bulk mode window size to enhance the throughput performance of SCP. **Note** Beginning from Cisco IOS XE Dublin 17.10.1, SSH bulk data transfer mode is enabled by default with default window size of 128KB.
Step 11	**exit** **Example:** `Device(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 12	**debug ip scp** **Example:** `Device# debug ip scp`	(Optional) Provides diagnostic information about SCP authentication problems.

Configuration Examples for Secure Copy

The following are examples of the Secure Copy configuration.

Example: Secure Copy Configuration Using Local Authentication

The following example shows how to configure the server-side functionality of Secure Copy. This example uses a locally defined username and password.

```
! AAA authentication and authorization must be configured properly in order for SCP to work.
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authentication login default local
Device(config)# aaa authorization exec default local
Device(config)# username user1 privilege 15 password 0 lab
! SSH must be configured and functioning properly.
Device(config)# ip scp server enable
Device(config)# end
```

Example: Secure Copy Server-Side Configuration Using Network-Based Authentication

The following example shows how to configure the server-side functionality of Secure Copy using a network-based authentication mechanism:

```
! AAA authentication and authorization must be configured properly for SCP to work.
Device> enable
Device# configure terminal
Device(config)# aaa new-model
Device(config)# aaa authentication login default group tacacs+
Device(config)# aaa authorization exec default group tacacs+
! SSH must be configured and functioning properly.
Device(config)# ip ssh time-out 120
Device(config)# ip ssh authentication-retries 3
Device(config)# ip scp server enable
Device(config)# end
```

Additional References for Secure Copy

Related Documents

Related Topic	Document Title
Secure Shell Version 1 and 2 support	*Configuring Secure Shell*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

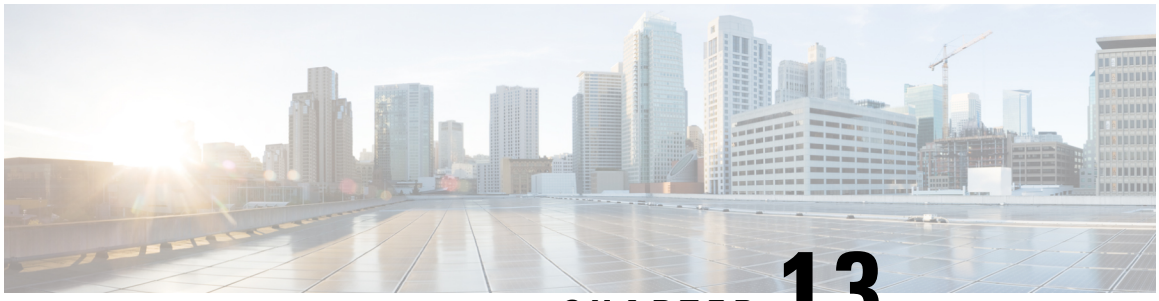
Feature History for Secure Copy

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.11.1	Secure Copy	The Secure Copy feature provides a secure and authenticated method for copying device configurations or device image files. SCP relies on SSH, an application and protocol that provide a secure replacement for the Berkeley r-tools suite. The following commands were introduced or modified: **debug ip scp** and **ip scp server enable**.
Cisco IOS XE Amsterdam 17.2.1	Secure Copy Performance Improvements	SSH bulk mode enables certain optimizations to enhance the throughput performance of procedures involving large amount of data transfer. This mode can be enabled by using the **ip ssh bulk-mode** global configuration command.
Cisco IOS XE Bengaluru 17.6.1	Secure Copy Improvement in Large RTT Scenario	Secure copy in large RTT settings can be configured by using the *window-size* variable option of the **ip ssh bulk-mode** command.
Cisco IOS XE Cupertino 17.7.1	Secure Copy	Support for this feature was introduced on the Cisco Catalyst 9600 Series Supervisor 2 Module.
Cisco IOS XE Dublin 17.10.1	Secure Copy Performance Improvements	SSH bulk mode is enabled by default with the default window size of 128KB.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.



CHAPTER 13

Configuration Replace and Configuration Rollback

Prerequisites for Configuration Replace and Configuration Rollback

The format of the configuration files used as input by the Configuration Replace and Configuration Rollback feature must comply with standard Cisco software configuration file indentation rules as follows:

- Start all commands on a new line with no indentation, unless the command is within a configuration submode.
- Indent commands within a first-level configuration submode one space.
- Indent commands within a second-level configuration submode two spaces.
- Indent commands within subsequent submodes accordingly.

These indentation rules describe how the software creates configuration files for such commands as **show running-config** or **copy running-config** *destination-url*. Any configuration file generated on a Cisco device complies with these rules.

Free memory larger than the combined size of the two configuration files (the current running configuration and the saved replacement configuration) is required.

Restrictions for Configuration Replace and Configuration Rollback

If the device does not have free memory larger than the combined size of the two configuration files (the current running configuration and the saved replacement configuration), the configuration replace operation is not performed.

Certain Cisco configuration commands such as those pertaining to physical components of a networking device (for example, physical interfaces) cannot be added or removed from the running configuration. For example, a configuration replace operation cannot remove the **interface ethernet 0** command line from the current running configuration if that interface is physically present on the device. Similarly, the **interface ethernet 1** command line cannot be added to the running configuration if no such interface is physically present on the device. A configuration replace operation that attempts to perform these types of changes results in error messages indicating that these specific command lines failed.

In very rare cases, certain Cisco configuration commands cannot be removed from the running configuration without reloading the device. A configuration replace operation that attempts to remove this type of command results in error messages indicating that these specific command lines failed.

Information About Configuration Replace and Configuration Rollback

Configuration Archive

The Cisco IOS configuration archive is intended to provide a mechanism to store, organize, and manage an archive of Cisco IOS configuration files to enhance the configuration rollback capability provided by the **configure replace** command. Before this feature was introduced, you could save copies of the running configuration using the **copy running-config** *destination-url* command, storing the replacement file either locally or remotely. However, this method lacked any automated file management. On the other hand, the Configuration Replace and Configuration Rollback feature provides the capability to automatically save copies of the running configuration to the Cisco IOS configuration archive. These archived files serve as checkpoint configuration references and can be used by the **configure replace** command to revert to previous configuration states.

The **archive config** command allows you to save Cisco IOS configurations in the configuration archive using a standard location and filename prefix that is automatically appended with an incremental version number (and optional timestamp) as each consecutive file is saved. This functionality provides a means for consistent identification of saved Cisco IOS configuration files. You can specify how many versions of the running configuration are kept in the archive. After the maximum number of files are saved in the archive, the oldest file is automatically deleted when the next, most recent file is saved. The **show archive** command displays information for all configuration files saved in the Cisco IOS configuration archive.

The Cisco IOS configuration archive, in which the configuration files are stored and available for use with the **configure replace** command, can be located on the following file systems: FTP, HTTP, RCP, TFTP.

Configuration Replace

The **configure replace** privileged EXEC command provides the capability to replace the current running configuration with any saved Cisco IOS configuration file. This functionality can be used to revert to a previous configuration state, effectively rolling back any configuration changes that were made since the previous configuration state was saved.

When using the **configure replace** command, you must specify a saved Cisco IOS configuration as the replacement configuration file for the current running configuration. The replacement file must be a complete configuration generated by a Cisco IOS device (for example, a configuration generated by the **copy running-config** *destination-url* command), or, if generated externally, the replacement file must comply with the format of files generated by Cisco IOS devices. When the **configure replace** command is entered, the current running configuration is compared with the specified replacement configuration and a set of diffs is generated. The algorithm used to compare the two files is the same as that employed by the **show archive config differences** command. The resulting diffs are then applied by the Cisco IOS parser to achieve the replacement configuration state. Only the diffs are applied, avoiding potential service disruption from reapplying configuration commands that already exist in the current running configuration. This algorithm effectively handles configuration changes to order-dependent commands (such as access lists) through a multiple pass process. Under normal circumstances, no more than three passes are needed to complete a configuration replace operation, and a limit of five passes is performed to preclude any looping behavior.

The Cisco IOS **copy** *source-url* **running-config** privileged EXEC command is often used to copy a stored Cisco IOS configuration file to the running configuration. When using the **copy** *source-url* **running-config** command as an alternative to the **configure replace** *target-url* privileged EXEC command, the following major differences should be noted:

- The **copy** *source-url* **running-config** command is a merge operation and preserves all of the commands from both the source file and the current running configuration. This command does not remove commands from the current running configuration that are not present in the source file. In contrast, the **configure replace** *target-url* command removes commands from the current running configuration that are not present in the replacement file and adds commands to the current running configuration that need to be added.
- The **copy** *source-url* **running-config** command applies every command in the source file, whether or not the command is already present in the current running configuration. This algorithm is inefficient and, in some cases, can result in service outages. In contrast, the **configure replace** *target-url* command only applies the commands that need to be applied—no existing commands in the current running configuration are reapplied.
- A partial configuration file may be used as the source file for the **copy** *source-url* **running-config** command, whereas a complete Cisco IOS configuration file must be used as the replacement file for the **configure replace** *target-url* command.

A locking feature for the configuration replace operation was introduced. When the **configure replace** command is used, the running configuration file is locked by default for the duration of the configuration replace operation. This locking mechanism prevents other users from changing the running configuration while the replacement operation is taking place, which might otherwise cause the replacement operation to terminate unsuccessfully. You can disable the locking of the running configuration by using the **no lock** keyword when issuing the **configure replace** command.

The running configuration lock is automatically cleared at the end of the configuration replace operation. You can display any locks that may be currently applied to the running configuration using the **show configuration lock** command.

Configuration Rollback

The concept of rollback comes from the transactional processing model common to database operations. In a database transaction, you might make a set of changes to a given database table. You then must choose whether to commit the changes (apply the changes permanently) or to roll back the changes (discard the changes and revert to the previous state of the table). In this context, rollback means that a journal file containing a log of the changes is discarded, and no changes are applied. The result of the rollback operation is to revert to the previous state, before any changes were applied.

The **configure replace** command allows you to revert to a previous configuration state, effectively rolling back changes that were made since the previous configuration state was saved. Instead of basing the rollback operation on a specific set of changes that were applied, the Cisco IOS configuration rollback capability uses the concept of reverting to a specific configuration state based on a saved Cisco IOS configuration file. This concept is similar to the database idea of saving a checkpoint (a saved version of the database) to preserve a specific state.

If the configuration rollback capability is desired, you must save the Cisco IOS running configuration before making any configuration changes. Then, after entering configuration changes, you can use that saved configuration file to roll back the changes (using the **configure replace** *target-url* command). Furthermore, because you can specify any saved Cisco IOS configuration file as the replacement configuration, you are not limited to a fixed number of rollbacks, as is the case in some rollback models.

Configuration Rollback Confirmed Change

The Configuration Rollback Confirmed Change feature allows configuration changes to be performed with an optional requirement that they be confirmed. If this confirmation is not received, the configuration is returned to the state prior to the changes being applied. The mechanism provides a safeguard against inadvertent loss of connectivity between a network device and the user or management application due to configuration changes.

Benefits of Configuration Replace and Configuration Rollback

- Allows you to revert to a previous configuration state, effectively rolling back configuration changes.
- Allows you to replace the current running configuration file with the startup configuration file without having to reload the device or manually undo CLI changes to the running configuration file, therefore reducing system downtime.
- Allows you to revert to any saved Cisco IOS configuration state.
- Simplifies configuration changes by allowing you to apply a complete configuration file to the device, where only the commands that need to be added or removed are affected.
- When using the **configure replace** command as an alternative to the **copy** *source-url* **running-config** command, increases efficiency and prevents risk of service outages by not reapplying existing commands in the current running configuration.

How to Use Configuration Replace and Configuration Rollback

Creating a Configuration Archive

No prerequisite configuration is needed to use the **configure replace** command. Using the **configure replace** command in conjunction with the Cisco IOS configuration archive and the **archive config** command is optional but offers significant benefit for configuration rollback scenarios. Before using the **archive config** command, the configuration archive must be configured. Perform this task to configure the characteristics of the configuration archive.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**archive** **Example:** `Device(config)# archive`	Enters archive configuration mode.
Step 4	**path** *url* **Example:** `Device(config-archive)# path flash:myconfiguration`	Specifies the location and filename prefix for the files in the Cisco IOS configuration archive. **Note** If a directory is specified in the path instead of file, the directory name must be followed by a forward slash as follows: path flash:/directory/. The forward slash is not necessary after a filename; it is only necessary when specifying a directory.
Step 5	**maximum** *number* **Example:** `Device(config-archive)# maximum 14`	(Optional) Sets the maximum number of archive files of the running configuration to be saved in the Cisco IOS configuration archive. • The *number* argument is the maximum number of archive files of the running configuration to be saved in the Cisco IOS

	Command or Action	Purpose
		configuration archive. Valid values are from 1 to 14. The default is 10. **Note** Before using this command, you must configure the **path** command to specify the location and filename prefix for the files in the Cisco IOS configuration archive.
Step 6	**time-period** *minutes* **Example:** `Device(config-archive)# time-period 1440`	(Optional) Sets the time increment for automatically saving an archive file of the current running configuration in the Cisco IOS configuration archive. • The *minutes* argument specifies how often, in minutes, to automatically save an archive file of the current running configuration in the Cisco IOS configuration archive. **Note** Before using this command, you must configure the **path** command to specify the location and filename prefix for the files in the Cisco IOS configuration archive.
Step 7	**end** **Example:** `Device(config-archive)# end`	Exits to privileged EXEC mode.
Step 8	**archive config** **Example:** `Device# archive config`	Saves the current running configuration file to the configuration archive. **Note** The **path** command must be configured before using this command.

Performing a Configuration Replace or Configuration Rollback Operation

Perform this task to replace the current running configuration file with a saved Cisco IOS configuration file.

Note You must create a configuration archive before performing this procedure. See Creating a Configuration Archive for detailed steps. The following procedure details how to return to that archived configuration in the event of a problem with the current running configuration.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure replace** *target-url* [**nolock**] [**list**] [**force**] [**ignore case**] [**revert trigger** [**error**] [**timer** *minutes*] | **time** *minutes*]] **Example:** `Device# configure replace flash: startup-config time 120`	Replaces the current running configuration file with a saved Cisco IOS configuration file. • The *target - url* argument is a URL (accessible by the Cisco IOS file system) of the saved Cisco IOS configuration file that is to replace the current running configuration, such as the configuration file created using the **archive config** command. • The **list** keyword displays a list of the command lines applied by the Cisco IOS software parser during each pass of the configuration replace operation. The total number of passes performed is also displayed. • The **force** keyword replaces the current running configuration file with the specified saved Cisco IOS configuration file without prompting you for confirmation. • The **time** *minutes* keyword and argument specify the time (in minutes) within which you must enter the **configure confirm** command to confirm replacement of the current running configuration file. If the **configure confirm** command is not entered within the specified time limit, the configuration replace operation is automatically reversed (in other words, the current running configuration file is restored to the configuration state that existed prior to entering the **configure replace** command). • The **nolock** keyword disables the locking of the running configuration file that prevents other users from changing the running configuration during a configuration replace operation.

	Command or Action	Purpose
		• The **revert trigger** keywords set the following triggers for reverting to the original configuration: • **error**: Reverts to the original configuration upon error. • **timer** *minutes*: Reverts to the original configuration if specified time elapses. **Note** In some cases, while performing the **revert trigger** operation for multiple pass operations, a partial configuration may be missed out causing the revert operation to the original configuration state to fail. • The **ignore case** keyword allows the configuration to ignore the case of the confirmation command.
Step 3	**configure revert** { **now** | **timer** { *minutes* | **idle** *minutes* } } **Example:** `Device# configure revert now`	(Optional) To cancel the timed rollback and trigger the rollback immediately, or to reset parameters for the timed rollback, use the **configure revert**command in privileged EXEC mode. • **now**: Triggers the rollback immediately. • **timer**: Resets the configuration revert timer. • Use the *minutes* argument with the **timer** keyword to specify a new revert time in minutes. • Use the **idle** keyword along with a time in minutes to set the maximum allowable time period of no activity before reverting to the saved configuration.
Step 4	**configure confirm** **Example:** `Device# configure confirm`	(Optional) Confirms replacement of the current running configuration file with a saved Cisco IOS configuration file.

	Command or Action	Purpose
		Note Use this command only if the **time** *seconds* keyword and argument of the **configure replace** command are specified.
Step 5	**exit** **Example:** `Device# exit`	Exits to user EXEC mode.

Monitoring and Troubleshooting the Feature

Perform this task to monitor and troubleshoot the Configuration Replace and Configuration Rollback feature.

Procedure

Step 1 **enable**

Use this command to enable privileged EXEC mode. Enter your password if prompted.

Example:

```
Device> enable
Device#
```

Step 2 **show archive**

Use this command to display information about the files saved in the Cisco IOS configuration archive.

Example:

```
Device# show archive
There are currently 1 archive configurations saved.
The next archive file will be named flash:myconfiguration-2
 Archive #  Name
   0
   1       flash:myconfiguration-1 <- Most Recent
   2
   3
   4
   5
   6
   7
   8
   9
   10
   11
   12
   13
   14
```

The following is sample output from the **show archive** command after several archive files of the running configuration have been saved. In this example, the maximum number of archive files to be saved is set to three.

Example:

```
Device# show archive
There are currently 3 archive configurations saved.
The next archive file will be named flash:myconfiguration-8
 Archive #  Name
   0
   1       :Deleted
   2       :Deleted
   3       :Deleted
   4       :Deleted
   5       flash:myconfiguration-5
   6       flash:myconfiguration-6
   7       flash:myconfiguration-7 <- Most Recent
   8
   9
   10
   11
   12
   13
   14
```

Step 3 debug archive versioning

Use this command to enable debugging of the Cisco IOS configuration archive activities to help monitor and troubleshoot configuration replace and rollback.

Example:

```
Device# debug archive versioning
Jan  9 06:46:28.419:backup_running_config
Jan  9 06:46:28.419:Current = 7
Jan  9 06:46:28.443:Writing backup file flash:myconfiguration-7
Jan  9 06:46:29.547: backup worked
```

Step 4 debug archive config timestamp

Use this command to enable debugging of the processing time for each integral step of a configuration replace operation and the size of the configuration files being handled.

Example:

```
Device# debug archive config timestamp
Device# configure replace flash:myconfiguration force
Timing Debug Statistics for IOS Config Replace operation:
       Time to read file usbflash0:sample_2.cfg = 0 msec (0 sec)
       Number of lines read:55
       Size of file        :1054
Starting Pass 1
       Time to read file system:running-config = 0 msec (0 sec)
       Number of lines read:93
       Size of file        :2539
       Time taken for positive rollback pass = 320 msec (0 sec)
       Time taken for negative rollback pass = 0 msec (0 sec)
       Time taken for negative incremental diffs pass = 59 msec (0 sec)
       Time taken by PI to apply changes = 0 msec (0 sec)
       Time taken for Pass 1 = 380 msec (0 sec)
Starting Pass 2
```

```
        Time to read file system:running-config = 0 msec (0 sec)
        Number of lines read:55
        Size of file        :1054
        Time taken for positive rollback pass = 0 msec (0 sec)
        Time taken for negative rollback pass = 0 msec (0 sec)
        Time taken for Pass 2 = 0 msec (0 sec)
Total number of passes:1
Rollback Done
```

Step 5 **exit**

Use this command to exit to user EXEC mode.

Example:

```
Device# exit
Device>
```

Configuration Examples for Configuration Replace and Configuration Rollback

Creating a Configuration Archive

The following example shows how to perform the initial configuration of the Cisco IOS configuration archive. In this example, flash:myconfiguration is specified as the location and filename prefix for the files in the configuration archive and a value of 10 is set as the maximum number of archive files to be saved.

```
configure terminal
!
archive
 path flash:myconfiguration
 maximum 10
end
```

Replacing the Current Running Configuration with a Saved Cisco IOS Configuration File

The following example shows how to replace the current running configuration with a saved Cisco IOS configuration file named flash:myconfiguration. The **configure replace** command interactively prompts you to confirm the operation.

```
Device# configure replace flash:myconfiguration
This will apply all necessary additions and deletions
to replace the current running configuration with the
contents of the specified configuration file, which is
assumed to be a complete configuration, not a partial
configuration. Enter Y if you are sure you want to proceed. ? [no]: Y
Total number of passes: 1
Rollback Done
```

In the following example, the **list** keyword is specified in order to display the command lines that were applied during the configuration replace operation:

```
Device# configure replace flash:myconfiguration list
This will apply all necessary additions and deletions
to replace the current running configuration with the
contents of the specified configuration file, which is
assumed to be a complete configuration, not a partial
configuration. Enter Y if you are sure you want to proceed. ? [no]: Y
!Pass 1
!List of Commands:
no snmp-server community public ro
snmp-server community mystring ro

end
Total number of passes: 1
Rollback Done
```

Reverting to the Startup Configuration File

The following example shows how to revert to the Cisco IOS startup configuration file using the **configure replace** command. This example also shows the use of the optional **force** keyword to override the interactive user prompt:

```
Device# configure replace flash:startup-config force
Total number of passes: 1
Rollback Done
```

Performing a Configuration Replace Operation with the configure confirm Command

The following example shows the use of the **configure replace** command with the **time** *minutes* keyword and argument. You must enter the **configure confirm** command within the specified time limit to confirm replacement of the current running configuration file. If the **configure confirm** command is not entered within the specified time limit, the configuration replace operation is automatically reversed (in other words, the current running configuration file is restored to the configuration state that existed prior to entering the **configure replace** command).

```
Device# configure replace flash:startup-config time 120
This will apply all necessary additions and deletions
to replace the current running configuration with the
contents of the specified configuration file, which is
assumed to be a complete configuration, not a partial
configuration. Enter Y if you are sure you want to proceed. ? [no]: Y
Total number of passes: 1
Rollback Done
Device# configure confirm
```

The following example shows the use of the **configure revert** command with the **timer** keyword. You must enter the **configure revert** command to cancel the timed rollback and trigger the rollback immediately, or to reset parameters for the timed rollback.

```
Device# configure revert timer 100
```

Performing a Configuration Rollback Operation

The following example shows how to make changes to the current running configuration and then roll back the changes. As part of the configuration rollback operation, you must save the current running configuration before making changes to the file. In this example, the **archive config** command is used to save the current running configuration. The generated output of the **configure replace** command indicates that only one pass was performed to complete the rollback operation.

Note Before using the **archive config** command, you must configure the **path** command to specify the location and filename prefix for the files in the Cisco IOS configuration archive.

You first save the current running configuration in the configuration archive as follows:

```
archive config
```

You then enter configuration changes as shown in the following example:

```
configure terminal
!
user netops2 password rain
user netops3 password snow
exit
```

After having made changes to the running configuration file, assume you now want to roll back these changes and revert to the configuration that existed before the changes were made. The **show archive** command is used to verify the version of the configuration to be used as a replacement file. The **configure replace** command is then used to revert to the replacement configuration file as shown in the following example:

```
Device# show archive
There are currently 1 archive configurations saved.
The next archive file will be named flash:myconfiguration-2
 Archive #  Name
   0
   1       flash:myconfiguration-1 <- Most Recent
   2
   3
   4
   5
   6
   7
   8
   9
   10
Device# configure replace flash:myconfiguration-1
Total number of passes: 1
Rollback Done
```

Additional References for Configuration Replace and Configuration Rollback

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	*Command Reference (Catalyst 9600 Series Switches)*

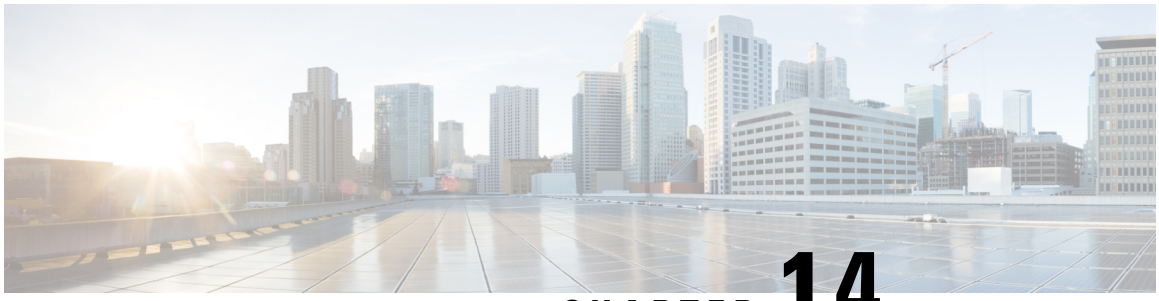
Feature History for Configuration Replace and Configuration Rollback

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.11.1	Configuration Replace and Configuration Rollback	The Cisco IOS configuration archive is intended to provide a mechanism to store, organize, and manage an archive of Cisco IOS configuration files to enhance the configuration rollback capability provided by the **configure replace** command.
Cisco IOS XE Cupertino 17.7.1	Configuration Replace and Configuration Rollback	Support for this feature was introduced on the Cisco Catalyst 9600 Series Supervisor 2 Module.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.



CHAPTER 14

BIOS Protection

Introduction to BIOS Protection

BIOS protection feature enables write-protection and secure upgrade of the golden ROMMON image. ROMMON is a bootstrap program that initializes the hardware and boots the Cisco IOS XE software image when you power on or restart the device. ROMMON upgrades can be required to resolve firmware defects or to support new features. Typically, ROM Monitor upgrades are infrequent and not required for every Cisco IOS XE software upgrade.

Without BIOS protection feature, golden ROMMON may be corrupted by malicious code during software upgrades.

ROMMON Upgrade

ROMMON images are stored on the SPI flash device as primary ROMMON and golden ROMMON. Primary ROMMON boots every time the device is powered on or restarted. If the primary ROMMON gets corrupted, the device uses the golden ROMMON to boot the IOS XE software image. When the device boots from the primary ROMMON, golden ROMMON is locked. With BIOS protection, golden ROMMON is made write-protected and cannot be upgraded using the flash utility upgrade mechanism. Access policies are governed by the FPGA firmware. FPGA blocks the disallowed operations such as write, erase etc on the golden ROMMON SPI flash device.

Note Golden ROMMON upgrade is not enabled without secure-boot FPGA upgrade.

Primary ROMMON, primary FPGA and golden FPGA (secure-boot FPGA) is automatically upgraded when the device boots. Golden ROMMON can only be upgraded using the capsule upgrade.

The upgrade process varies between standalone and high availability systems and is explained below.

Standalone Systems

For a standalone device, when your upgrade the device in install mode, the primary ROMMON is automatically upgraded when the device boots. Golden ROMMON can be upgraded using the capsule upgrade.

High Availability and StackWise Virtual Systems

We recommend that you perfom In-Service-Software-Upgrade (ISSU) for devices in a high availability setup. FPGA upgrades occur as part of ISSU.

If you are performing the upgrade in install mode with reload, do not reload both the supervisors at the same time. With the standby supervisor in ROMMON state, boot the active supervisor. When ROMMON upgrade is completed on each supervisor, FPGA and software image is upgraded.

Boot the standby supervisor and allow the standby supervisor to upgrade and reach standby hot state.

Capsule Upgrade

In a capsule upgrade, a secure update capsule is created and signed which is used by the primary ROMMON after authentication for upgrading the golden ROMMON. The secure update capsule requires a secure flash certificate. Secure flash certificate is created using the product key and added to the primary ROMMON image to verify the authenticity of the update capsule. A capsule is now created using the secure flash certificate and a secure boot 16 MB flash image and signed.

When the device boots, the primary ROMMON triggers the capsule upgrade for the golden ROMMON. To perform capsule upgrade for the golden ROMMON, use the **upgrade rom-monitor capsule golden switch** command in privileged EXEC mode.

The following processes occur in a capsule upgrade:

- The device checks if secure-boot FPGA upgrade is enabled. If not, the process exits.
- The device checks if bootloader protection is enabled. If not, a one-time upgrade of primary ROMMON, golden ROMMON, and primary FPGA is initiated.
- If bootloader protection is already active, IOS copies the secure update capsule to bootflash and the device reboots.
- When the device reboots, secure update capsule is picked for performing the upgrade.

Feature History for BIOS Protection

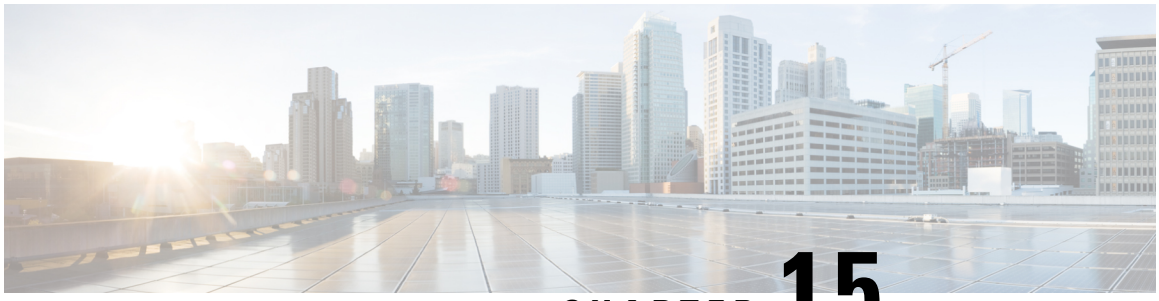
This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.12.1	BIOS Protection	BIOS Protection feature enables write-protection and secure upgrade of the golden ROMMON image.

Release	Feature	Feature Information
Cisco IOS XE Amsterdam 17.1.1	Capsule Upgrade	Support for capsule upgrade for golden ROMMON using **upgrade rom-monitor capsule switch active** command was enabled.
Cisco IOS XE Cupertino 17.7.1	BIOS Protection	Support for this feature was introduced on the C9500X-28C8D model of the Cisco Catalyst 9500 Series Switches. Support for this feature was introduced on the Cisco Catalyst 9600 Series Supervisor 2 Module.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.



CHAPTER 15

Software Maintenance Upgrade

Software Maintenance Upgrade (a SMU), is a package that can be installed on a system to provide a fix or a security resolution to a released image.

Restrictions for Software Maintenance Upgrade

- SMU supports patching using install mode only.
- Prior to Cisco IOS XE Bengaluru 17.9.1, SMU installation was supported both in bundle and install modes. From Cisco IOS XE Bengaluru 17.9.1, SMU installation will be supported in install mode only.

Information About Software Maintenance Upgrade

SMU Overview

An SMU is a package that can be installed on a system, to provide a fix or a security resolution to a released image. An SMU package is provided on a per release and per component basis.

An SMU provides a significant benefit over classic Cisco IOS software because it allows you to address network issues quickly while reducing the time and scope of the testing required. The Cisco IOS XE platform internally validates SMU compatibility and does not allow you to install incompatible SMUs.

All SMUs are integrated into the subsequent Cisco IOS XE software maintenance releases. An SMU is an independent and self-sufficient package and it does not have any prerequisites or dependencies. You can choose which SMUs to install or uninstall in any order.

SMUs are supported only on Extended Maintenance releases and for the full lifecycle of the underlying software release.

Perform these basic steps to install an SMU:

1. Add the SMU to the filesystem.
2. Activate the SMU on the system.
3. Commit the SMU changes so that it is persistent across reloads.

SMU Workflow

The SMU process is initiated with a request to the Cisco Customer Support. Contact your customer support to raise an SMU request.

At release time, the SMU package is posted to the Cisco Software Download page and can be downloaded and installed.

SMU Package

The SMU package contains a small set of files for patching the release along with metadata that describes the contents of the package, and fix for the reported issue that the SMU is requested for.

SMU Reload

The SMU type describes the effect the installed SMU has on the corresponding system. SMUs might not have an impact on traffic, or might result in device restart, reload, or switchover. Run the **show install package flash**: *filename* command to verify whether a reload is required or not.

Hot patching enables SMU to take effect after activation without the system having to be reloaded. After the SMU is committed, the changes are persistent across reloads. In certain cases, SMUs may require a cold (complete) reload of the operating system. This action affects the traffic flow for the duration of the reload. If a cold reload is required, users will be prompted to confirm the action.

Note If the user deletes the SMU file from the directory and performs a bootup, the device displays the error message **%BOOT-3-BOOTTIME_SMU_MISSING_DETECTED: R0/0: install_engine: SMU file /bootflash/cat9k_iosxe-lni.BLD_POLARIS_DEV_LATEST_20210616_160027.SSA.bin missing and system impact will be unknown**. However, this will not lead to any functional error.

How to Manage Software Maintenance Updates

The following sections provide information about managing SMUs.

You can install, activate, and commit an SMU package using a single command (1-step process) or using separate commands (3-step process).

Tip Use the 1-step process when you have to install just one SMU package file and use the 3-step process when you have to install multiple SMUs. The 3-step process minimises the number of reloads required when you have more than one SMU package file to install.

Installing an SMU Package: 1-Step Process

This task shows how to use the single **install add file activate commit** command for installing an SMU package.

Before you begin

Check that the SMU you are about to install corresponds to the software image installed on your device. For example, SMU `cat9k_lite_iosxe.16.09.04.CSCvk70181.SPA.smu.bin` is compatible with software image `cat9k_lite_iosxe.16.09.04.SPA.bin`.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.
Step 2	**install add file flash:** *filename* [**activate commit**] **Example:** `Device# install add file flash:cat9k_lite_iosxe.16.09.04.CSCvk70181.SPA.smu.bin activate commit`	Copies the maintenance update package from flash to the device, performs a compatibility check for the platform and image versions, activates the SMU package, and makes the package persistent across reloads. This command extracts the individual components of the .bin file into the subpackages and packages.conf files. You can also copy the SMU package from from a remote location (through FTP, HTTP, HTTPS, or TFTP). **Note** If the SMU file is copied using TFTP, use bootflash to activate the SMU.
Step 3	**exit** **Example:** `Device# exit`	Exits privileged EXEC mode and returns to user EXEC mode.

Installing an SMU Package: 3-Step Process

This task shows you the 3-step process for installing an SMU package. Use this method to install multiple SMUs and avoid multiple reloads.

Before you begin

Check that the SMU you are about to install corresponds to the software image installed on your device. For example, SMU `cat9k_lite_iosxe.16.09.04.CSCvk70181.SPA.smu.bin` is compatible with software image `cat9k_lite_iosxe.16.09.04.SPA.bin`.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.

Step 2	**install add file** *location filename*
	Example:
	`Device# install add file flash:cat9k_lite_iosxe.16.12.03.CSCvt22238.SPA.smu.bin`
	`Device# install add file flash:cat9k_lite_iosxe.16.12.03.CSCvt72427.SPA.smu.bin`
		Copies the maintenance update package from flash to the device, and then performs a compatibility check for the platform and image versions, and adds the SMU package on all member nodes or FRUs, as applicable. This command also runs base compatibility checks on a file to ensure that the SMU package is supported on the platform. It also adds an entry in the package/SMU.sta file, so that its status can be monitored and maintained.
		You can also copy the SMU package from a remote location (through FTP, HTTP, HTTPS, or TFTP).

Step 3	**install activate file** *location filename* **Example:** `Device# install activate file flash:cat9k_lite_iosxe.16.12.03.CSCvt22238.SPA.smu.bin, cat9k_lite_iosxe.16.12.03.CSCvt72427.SPA.smu.bin`	Activates the SMU package file that was added and updates the package status details. You will be promped to reload the system in order to complete the activation process. When entering multiple SMUs, use a comma (without a space before or after), to separate file names. Also ensure that total number of characters does not exceed 128. This step involves a reload.
Step 4	**install commit** **Example:** `Device# install commit`	Commits the activation changes to be persistent across reloads. The commit can be done after activation while the system is up, or after the first reload. If a package is activated but not committed, it remains active after the first reload, but not after the second reload.

Managing an SMU

This task shows how to rollback the installation state, deactivate, and remove a previously installed SMU package from the device. This can be used for a SMU that has been installed with the 1-step and 3-step process.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**install rollback to** {**base** | **committed** | **id** *commit-ID*} **Example:** `Device# install rollback to committed`	Returns the device to the previous installation state. After the rollback, a reload is required.
Step 3	**install deactivate file** *location filename* **Example:** `Device# install deactivate file flash:cat9k_lite_iosxe.16.09.04.CSCvk70181.SPA.smu.bin`	Deactivates an active package, updates the package status, and triggers a process to restart or reload.
Step 4	**install remove** {**file** *location filename* | **inactive**} **Example:** `Device# install remove file flash:cat9k_lite_iosxe.16.09.04.CSCvk70181.SPA.smu.bin`	Checks if the specified SMU is inactive and if it is, deletes it from the file system. The **inactive** option deletes all the inactive packages from the file system.
Step 5	**show version** **Example:** `Device# show version`	Displays the image version on the device.
Step 6	**show install summary** **Example:** `Device# show install summary`	Displays information about the active package. The output of this command varies according to the **install** commands that are configured.

Configuration Examples for Software Maintenance Upgrade

The following is a list of SMU configuration examples.

Example: Managing an SMU

Note
- The examples used in this section are of hot patching SMU.

The following example shows how to copy an SMU file to flash:

```
Device# copy ftp://172.16.0.10//auto/ftpboot/user/
cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin

flash:
Destination filename
[cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin]?
Accessing ftp://172.16.0.10//auto/ftpboot/folder1/
cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin...
Loading /auto/ftpboot/folder1/
cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin from
172.16.0.10 (via GigabitEthernet0): !
[OK - 17668 bytes]
17668 bytes copied in 0.058 secs (304621 bytes/sec)
```

The following example shows how to add a maintenance update package file:

```
Device# install add file
flash:cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin

install_add: START Mon Mar  5 21:48:51 PST 2018
install_add: Adding SMU

--- Starting initial file syncing ---
Info: Finished copying
flash:cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin to the
 selected switch(es)
Finished initial file syncing

Executing pre scripts....

Executing pre scripts done.
--- Starting SMU Add operation ---
Performing SMU_ADD on all members
  [1] SMU_ADD package(s) on switch 1
  [1] Finished SMU_ADD on switch 1
Checking status of SMU_ADD on [1]
SMU_ADD: Passed on [1]
Finished SMU Add operation

SUCCESS: install_add
/flash/cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin Mon
Mar  5 21:49:00 PST 2018
```

The following is a sample output from the **show install summary** command after adding an SMU package file to the device:

```
Device# show install summary

[ Switch 1 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
```

```
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
SMU   I    flash:cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin
IMG   C    16.9.1.0.43131

--------------------------------------------------------------------------------
Auto abort timer: inactive
--------------------------------------------------------------------------------
```

The following example shows how to activate an added SMU package file:

```
Device# install activate file
flash:cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin

install_activate: START Mon Mar  5 21:49:22 PST 2018
install_activate: Activating SMU
Executing pre scripts....

Executing pre scripts done.

--- Starting SMU Activate operation ---
Performing SMU_ACTIVATE on all members
  [1] SMU_ACTIVATE package(s) on switch 1
  [1] Finished SMU_ACTIVATE on switch 1
Checking status of SMU_ACTIVATE on [1]
SMU_ACTIVATE: Passed on [1]
Finished SMU Activate operation

SUCCESS: install_activate
/flash/cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin Mon
Mar  5 21:49:34 PST 2018
```

The following a sample output from the **show version** command:

```
Device# show version

Cisco IOS XE Software, Version BLD_POLARIS_DEV_LATEST_20180302_085005_2 - SMU-PATCHED
Cisco IOS Software [Fuji], Catalyst L3 Switch Software (CAT9K_IOSXE), Experimental Version
 16.9.20180302:
085957 [polaris_dev-/nobackup/mcpre/BLD-BLD_POLARIS_DEV_LATEST_20180302_085005 166]
Copyright (c) 1986-2018 by Cisco Systems, Inc.
Compiled Fri 02-Mar-18 09:50 by mcpre
...
```

The following is a sample output from the **show install summary** command displays the status of the SMU package as active and uncommitted:

```
Device# show install summary

[ Switch 1 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
SMU   U    flash:cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin
IMG   C    16.9.1.0.43131

--------------------------------------------------------------------------------
Auto abort timer: active on install_activate, time before rollback - 01:59:50
```

```
--------------------------------------------------------------------------------
```

The following is a sample output from the **show install active** command:

```
Device# show install active

[ Switch 1 ] Active Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
SMU   U    flash:cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin
IMG   C    16.9.1.0.43131
```

The following example shows how to execute the **install commit** command:

```
Device# install commit

install_commit: START Mon Mar  5 21:50:52 PST 2018
install_commit: Committing SMU
Executing pre scripts....

Executing pre scripts done.
--- Starting SMU Commit operation ---
Performing SMU_COMMIT on all members
  [1] SMU_COMMIT package(s) on switch 1
  [1] Finished SMU_COMMIT on switch 1
Checking status of SMU_COMMIT on [1]
SMU_COMMIT: Passed on [1]
Finished SMU Commit operation

SUCCESS: install_commit
/flash/cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin Mon
Mar  5 21:51:01 PST 2018
```

The following is a sample output from the **show install summary** command displays that the update package is now committed, and that it will be persistent across reloads:

```
Device# show install summary

[ Switch 1 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
SMU   C    flash:cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin
IMG   C    16.9.1.0.43131

--------------------------------------------------------------------------------
Auto abort timer: inactive
--------------------------------------------------------------------------------
```

The following example shows how to rollback an update package to the committed package:

```
Device# install rollback to committed

install_rollback: START Mon Mar  5 21:52:18 PST 2018
install_rollback: Rolling back SMU
Executing pre scripts....
```

```
Executing pre scripts done.

--- Starting SMU Rollback operation ---
Performing SMU_ROLLBACK on all members
  [1] SMU_ROLLBACK package(s) on switch 1
  [1] Finished SMU_ROLLBACK on switch 1
Checking status of SMU_ROLLBACK on [1]
SMU_ROLLBACK: Passed on [1]
Finished SMU Rollback operation

SUCCESS: install_rollback
/flash/cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin Mon
Mar  5 21:52:30 PST 2018
```

The following is a sample output from the **show install summary** command:

```
Device# show install summary

[ Switch 1 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
IMG   C    16.9.1.0.43131

--------------------------------------------------------------------------------
Auto abort timer: inactive
--------------------------------------------------------------------------------
```

The following example shows how to deactivate an SMU package file:

```
Device# install deactivate file
flash:cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin

install_deactivate: START Mon Mar  5 21:54:06 PST 2018
install_deactivate: Deactivating SMU
Executing pre scripts....

Executing pre scripts done.

--- Starting SMU Deactivate operation ---
Performing SMU_DEACTIVATE on all members
  [1] SMU_DEACTIVATE package(s) on switch 1
  [1] Finished SMU_DEACTIVATE on switch 1
Checking status of SMU_DEACTIVATE on [1]
SMU_DEACTIVATE: Passed on [1]
Finished SMU Deactivate operation

SUCCESS: install_deactivate
/flash/cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin Mon
Mar  5 21:54:17 PST 2018
```

The following is a sample output from the **show install summary** command:

```
Device# show install summary

[ Switch 1 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
```

```
SMU   D    flash:cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin
IMG   C    16.9.1.0.43131

--------------------------------------------------------------------------------
Auto abort timer: active on install_deactivate, time before rollback - 01:59:50
--------------------------------------------------------------------------------
```

The following example shows how to remove an SMU from the device:

```
Device# install remove file
flash:cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin

install_remove: START Mon Mar  5 22:03:50 PST 2018
install_remove: Removing SMU
Executing pre scripts....

Executing pre scripts done.

--- Starting SMU Remove operation ---
Performing SMU_REMOVE on all members
  [1] SMU_REMOVE package(s) on switch 1
  [1] Finished SMU_REMOVE on switch 1
Checking status of SMU_REMOVE on [1]
SMU_REMOVE: Passed on [1]
Finished SMU Remove operation

SUCCESS: install_remove
/flash/cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin Mon
Mar  5 22:03:58 PST 2018
```

The following is a sample output from the **show install summary** command:

```
Device# show install summary

[ Switch 1 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
            C - Activated & Committed, D - Deactivated & Uncommitted
--------------------------------------------------------------------------------
Type  St   Filename/Version
--------------------------------------------------------------------------------
IMG   C    16.9.1.0.43131

--------------------------------------------------------------------------------
Auto abort timer: inactive
--------------------------------------------------------------------------------
```

Additional References for Software Maintenance Upgrade

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	*Command Reference (Catalyst 9600 Series Switches)*

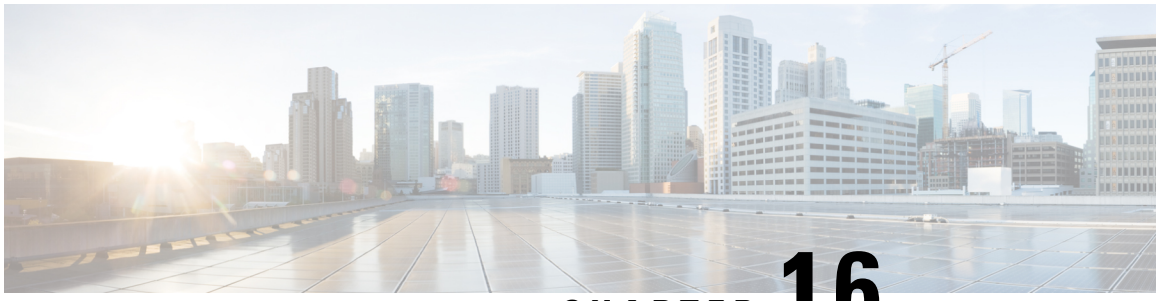
Feature History for Software Maintenance Upgrade

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.11.1	Software Maintenance Upgrade (SMU)	An SMU is a package that can be installed on a system to provide a fix or a security resolution to a released image. Feature support includes hot patching and PKI patching support.
Cisco IOS XE Cupertino 17.7.1	Software Maintenance Upgrade (SMU)	Support for this feature was introduced on the Cisco Catalyst 9600 Series Supervisor 2 Module.
Cisco IOS XE Cupertino 17.9.1	Software Maintenance Upgrade (SMU)	SMU installation is supported in install mode only.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.



CHAPTER 16

Working with the Flash File System

Information About the Flash File System

The flash file system is a single flash device on which you can store files. It also provides several commands to help you manage software bundles and configuration files. The default flash file system on the device is named flash:.

As viewed from the active device, flash: refers to the local flash device, which is the device attached to the same device on which the file system is being viewed.

Only one user at a time can manage the software bundles and configuration files.

Displaying Available File Systems

To display the available file systems on your device, use the **show file systems** privileged EXEC command as shown in this example for a standalone device:

```
Device# show file systems
File Systems:
Size(b) Free(b) Type Flags Prefixes
- - opaque rw system:
- - opaque rw tmpsys:
1651314688 1467920384 disk rw crashinfo:
* 11353194496 6942072832 disk rw flash:
7723847680 7646384128 disk ro webui:
```

```
- - opaque rw null:
- - opaque ro tar:
- - network rw tftp:
2097152 2089932 nvram rw nvram:
- - network rw rcp:
- - network rw http:
- - network rw ftp:
- - network rw scp:
- - network rw https:
- - opaque ro cns:
118014062592 111933124608 disk rw usbflash1:
```

This example displays the usbflash1 filesystem format.

```
Device#show usbflash1: filesys
Filesystem: usbflash1
Filesystem Path: /vol/usb1
Filesystem Type: ext4
Mounted: Read/Write
```

This example shows a device stack. In this example, the active device is stack member 2; the file system on stack member 1 is displayed as flash-1:,the file system on stack member 2 is displayed as flash-2:, the file system on stack member 3 is displayed as flash-3: and so on up to . The example also shows the crashinfo directories and a USB flash drive plugged into the active device:

```
Device# show file systems
File Systems:

       Size(b)       Free(b)      Type  Flags  Prefixes
             -             -    opaque     rw   system:
             -             -    opaque     rw   tmpsys:
    1651314688    1565089792      disk     rw   crashinfo: crashinfo-2:
    1651507200    1560281088      disk     rw   crashinfo-1:
    1651507200    1562378240      disk     rw   crashinfo-3: stby-crashinfo:
*  11353194496   10735611904      disk     rw   flash: flash-2:
   11353980928   10152312832      disk     rw   flash-1:
   11353980928    2161115136      disk     rw   flash-3: stby-flash:
   15243046912   14423638016      disk     rw   usbflash0: usbflash0-2:
     520093696     520093696      disk     rw   usbflash0-1:
    3497074688    3417554944      disk     ro   webui:
             -             -    opaque     rw   null:
             -             -    opaque     ro   tar:
             -             -   network     rw   tftp:
       2097152       2085334     nvram     rw   nvram:
             -             -   network     rw   rcp:
             -             -   network     rw   http:
             -             -   network     rw   ftp:
             -             -   network     rw   scp:
             -             -   network     rw   https:
             -             -    opaque     ro   cns:
   21003628544   19867037696      disk     rw   usbflash1: usbflash1-2:
  118014083072  111933390848      disk     rw   usbflash1-3: stby-usbflash1:
       2097152       2085334     nvram     rw   stby-nvram:
             -             -     nvram     rw   stby-rcsf:
             -             -    opaque     rw   revrcsf:
```

Table 28: show file systems Field Descriptions

Field	Value
Size(b)	Amount of memory in the file system in bytes.
Free(b)	Amount of free memory in the file system in bytes.
Type	Type of file system. **disk**—The file system is for a flash memory device, USB flash, and crashinfo file. **network**—The file system for network devices; for example, an FTP server or and HTTP server. **nvram**—The file system is for a NVRAM device. **opaque**—The file system is a locally generated pseudo file system (for example, the system) or a download interface, such as brimux. **unknown**—The file system is an unknown type.
Flags	Permission for file system. **ro**—read-only. **rw**—read/write. **wo**—write-only.
Prefixes	Alias for file system. **crashinfo:**—Crashinfo file. **flash:**—Flash file system. **ftp:**—FTP server. **http:**—HTTP server. **https:**—Secure HTTP server. **nvram:**—NVRAM. **null:**—Null destination for copies. You can copy a remote file to null to find its size. **rcp:**—Remote Copy Protocol (RCP) server. **scp:**—Session Control Protocol (SCP) server. **system:**—Contains the system memory, including the running configuration. **tftp:**—TFTP network server. **usbflash0:**—USB flash memory. **usbflash1:**—External USB flash memory. **ymodem:**—Obtain the file from a network machine by using the Ymodem protocol.

Setting the Default File System

You can specify the file system or directory that the system uses as the default file system by using the **cd** *filesystem:* privileged EXEC command. You can set the default file system to omit the *filesystem:* argument from related commands. For example, for all privileged EXEC commands that have the optional *filesystem:* argument, the system uses the file system specified by the **cd** command.

By default, the default file system is *flash:*.

You can display the current default file system as specified by the **cd** command by using the **pwd** privileged EXEC command.

Displaying Information About Files on a File System

You can view a list of the contents of a file system before manipulating its contents. For example, before copying a new configuration file to flash memory, you might want to verify that the file system does not already contain a configuration file with the same name. Similarly, before copying a flash configuration file to another location, you might want to verify its filename for use in another command. To display information about files on a file system, use one of the privileged EXEC commands listed in the following table.

Table 29: Commands for Displaying Information About Files

Command	Description
dir [/all] **[***filesystem:filename***]**	Displays a list of files on a file system.
show file systems	Displays more information about each of the files on a file system.
show file information *file-url*	Displays information about a specific file.
show file descriptors	Displays a list of open file descriptors. File descriptors are the internal representations of open files. You can use this command to see if another user has a file open.

For example, to display a list of all files in a file system, use the **dir** privileged EXEC command:

```
Device# dir flash:
DDirectory of bootflash:/

616513  drwx             4096  Jul 15 2015 07:11:35 +00:00  .installer
608402  -rw-            33818  Sep 25 2015 11:41:35 +00:00  bootloader_evt_handle.log
608403  drwx             4096  Feb 27 2017 13:56:47 +00:00  .ssh
608410  -rw-                0   Jun 5 2015 10:16:17 +00:00  dc_stats.txt
608411  drwx            20480  Sep 23 2015 11:50:13 +00:00  core
624625  drwx             4096  Sep 23 2015 12:29:27 +00:00  .prst_sync
640849  drwx             4096  Feb 27 2017 13:57:30 +00:00  .rollback_timer
608412  drwx             4096  Jun 17 2015 18:12:47 +00:00  orch_test_logs
608413  -rw-         33554432  Sep 25 2015 11:43:15 +00:00  nvram_config
608417  -rw-               35  Sep 25 2015 20:17:42 +00:00  pnp-tech-time
608439  -rw-           214054  Sep 25 2015 20:17:48 +00:00  pnp-tech-discovery-summary
608419  drwx             4096  Jul 23 2015 07:50:25 +00:00  util
```

```
616514  drwx             4096  Mar 18 2015 11:09:04 +00:00  onep
608442  -rw-              556  Mar 18 2015 11:19:34 +00:00  vlan.dat
608448  -rw-          1131779  Mar 28 2015 13:13:48 +00:00  log.txt
616516  drwx             4096   Apr 1 2015 09:34:56 +00:00  gs_script
616517  drwx             4096   Apr 6 2015 09:42:38 +00:00  tools
608440  -rw-              252  Sep 25 2015 11:41:52 +00:00  boothelper.log
624626  drwx             4096  Apr 17 2015 06:10:55 +00:00  SD_AVC_AUTO_CONFIG
608488  -rw-            98869  Sep 25 2015 11:42:15 +00:00  memleak.tcl
608437  -rwx            17866  Jul 16 2015 04:01:10 +00:00  ardbeg_x86
632745  drwx             4096  Aug 20 2015 11:35:09 +00:00  CRDU
632746  drwx             4096  Sep 16 2015 08:57:44 +00:00  ardmore
608418  -rw-          1595361   Jul 8 2015 11:18:33 +00:00
system-report_RP_0_20150708-111832-UTC.tar.gz
608491  -rw-         67587176  Aug 12 2015 05:30:35 +00:00  mcln_x86_kernel_20170628.SSA
608492  -rwx         74880100  Aug 12 2015 05:30:57 +00:00  stardust.x86.idprom.0718B

11250098176 bytes total (9128050688 bytes free)
Device#
```

Changing Directories and Displaying the Working Directory

Follow these steps to change directories and to display the working directory:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**dir** *filesystem:* **Example:** `Device# dir flash:`	Displays the directories on the specified file system. For *filesystem:*, use flash: for the system board flash device.
Step 3	**cd** *directory_name* **Example:** `Device# cd new_configs`	Navigates to the specified directory. The command example shows how to navigate to the directory named *new_configs*.
Step 4	**pwd** **Example:** `Device# pwd`	Displays the working directory.
Step 5	**cd** **Example:** `Device# cd`	Navigates to the default directory.

Creating Directories

Beginning in privileged EXEC mode, follow these steps to create a directory:

Procedure

	Command or Action	Purpose
Step 1	**dir** *filesystem:* **Example:** `Device# dir flash:`	Displays the directories on the specified file system. For *filesystem:*, use flash: for the system board flash device.
Step 2	**mkdir** *directory_name* **Example:** `Device# mkdir new_configs`	Creates a new directory. Directory names are case sensitive and are limited to 45 characters between the slashes (/); the name cannot contain control characters, spaces, slashes, quotes, semicolons, or colons.
Step 3	**dir** *filesystem:* **Example:** `Device# dir flash:`	Verifies your entry.

Removing Directories

To remove a directory with all its files and subdirectories, use the **delete /force /recursive** *filesystem:/file-url* privileged EXEC command.

Use the **/recursive** keyword to delete the named directory and all subdirectories and the files contained in it. Use the **/force** keyword to suppress the prompting that confirms a deletion of each file in the directory. You are prompted only once at the beginning of this deletion process.

For *filesystem*, use **flash:** for the system board flash device. For *file-url*, enter the name of the directory to be deleted. All of the files in the directory and the directory are removed.

Caution When directories are deleted, their contents cannot be recovered.

Copying Files

To copy a file from a source to a destination, use the **copy** *source-url destination-url* privileged EXEC command. For the source and destination URLs, you can use **running-config** and **startup-config** keyword shortcuts. For example, the **copy running-config startup-config** command saves the currently running configuration file to the NVRAM section of flash memory to be used as the configuration during system initialization.

You can also copy from special file systems (**xmodem:**, **ymodem:**) as the source for the file from a network machine that uses the Xmodem or Ymodem protocol. SSH File Transfer Protocol (SFTP) is also another option to copy switch configuration or image files. For more information, refer the *Configuring SSH File Transfer Protocol* chapter of the *Security Configuration Guide*.

Network file system URLs include ftp:, rcp:, tftp:, scp:, http:, and https: and have these syntaxes:

- FTP—ftp:[[//username [:password]@location]/directory]/filename
- RCP—rcp:[[//username@location]/directory]/filename
- TFTP—tftp:[[//location]/directory]/filename
- SCP—scp:[[//username [:password]@location]/directory]/filename
- HTTP—http:[[//username [:password]@location]/directory]/filename
- HTTPS—https:[[//username [:password]@location]/directory]/filename

Note The password must not contain the special character '@'. If the character '@' is used, the copy fails to parse the IP address of the server.

Local writable file systems include flash:.

Some invalid combinations of source and destination exist. Specifically, you cannot copy these combinations:

- From a running configuration to a running configuration
- From a startup configuration to a startup configuration

Deleting Files

When you no longer need a file on a flash memory device, you can permanently delete it. To delete a file or directory from a specified flash device, use the **delete** [**/force**] [**/recursive**] [*filesystem:*]*/file-url* privileged EXEC command.

Use the **/recursive** keyword for deleting a directory and all subdirectories and the files contained in it. Use the **/force** keyword to suppress the prompting that confirms a deletion of each file in the directory. You are prompted only once at the beginning of this deletion process. Use the **/force** and **/recursive** keywords for deleting old software images that were installed by using the **archive download-sw** command but are no longer needed.

If you omit the *filesystem:* option, the device uses the default device specified by the **cd** command. For *file-url*, you specify the path (directory) and the name of the file to be deleted.

When you attempt to delete any files, the system prompts you to confirm the deletion.

Caution When files are deleted, their contents cannot be recovered.

This example shows how to delete the file *myconfig* from the default flash memory device:

```
Device# delete myconfig
```

Creating, Displaying and Extracting Files

You can create a file and write files into it, list the files in a file, and extract the files from a file as described in the next sections.

Beginning in privileged EXEC mode, follow these steps to create a file, display the contents, and extract it:

Procedure

	Command or Action	Purpose
Step 1	**archive tar /create** *destination-url* **flash:** */file-url* **Example:** `Device# archive tar /create tftp:172.20.10.30/saved. flash:/new-configs`	Creates a file and adds files to it. For destination-url, specify the destination URL alias for the local or network file system and the name of the file to create: • Local flash file system syntax: **flash:** • FTP syntax: **ftp**:[[*//username*[*:password*][*@location*]*/directory*]*/-filename*. • RCP syntax: **rcp**:[[*//username@location*]*/directory*]*/-filename*. • TFTP syntax: **tftp**:[[*//location*]*/directory*]*/-filename*. For **flash:***/file-url*, specify the location on the local flash file system in which the new file is created. You can also specify an optional list of files or directories within the source directory to add to the new file. If none are specified, all files and directories at this level are written to the newly created file.
Step 2	**archive tar /table** *source-url* **Example:** `Device# archive tar /table flash: /new_configs`	Displays the contents of a file. For *source-url*, specify the source URL alias for the local or network file system. The *-filename.* is the file to display. These options are supported: • Local flash file system syntax: **flash:** • FTP syntax: **ftp**:[[*//username*[*:password*][*@location*]*/directory*]*/-filename*. • RCP syntax: **rcp**:[[*//username@location*]*/directory*]*/-filename*. • TFTP syntax:

	Command or Action	Purpose
		tftp:[[//*location*]/*directory*]/*-filename*. You can also limit the file displays by specifying a list of files or directories after the file. Only those files appear. If none are specified, all files and directories appear.
Step 3	**archive tar /xtract** *source-url* **flash:**/*file-url* [*dir/file...*] **Example:** `Device# archive tar /xtract tftp:/172.20.10.30/saved. flash:/new-configs`	Extracts a file into a directory on the flash file system. For *source-url*, specify the source URL alias for the local file system. The *-filename.* is the file from which to extract files. These options are supported: • Local flash file system syntax: **flash:** • FTP syntax: **ftp**:[[//*username*[:*password*]@*location*]/*directory*]/*-filename*. • RCP syntax: **rcp**:[[//*username@location*]/*directory*]/*-filename*. • TFTP syntax: **tftp**:[[//*location*]/*directory*]/*-filename*. For **flash:**/*file-url* [*dir/file...*], specify the location on the local flash file system from which the file is extracted. Use the *dir/file...* option to specify a list of files or directories within the file to be extracted. If none are specified, all files and directories are extracted.
Step 4	**more** [/**ascii** | /**binary** | /**ebcdic**] /*file-url* **Example:** `Device# more flash:/new-configs`	Displays the contents of any readable file, including a file on a remote file system.

Additional References for Flash File System

Related Documents

Related Topic	Document Title
Commands for managing flash: file systems	*Cisco IOS Configuration Fundamentals Command Reference*

Feature History for Flash File System

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.11.1	Flash File System	The flash file system is a single flash device on which you can store files. It also provides several commands to help you manage software bundles and configuration files.
Cisco IOS XE Cupertino 17.7.1	Flash File System	Support for this feature was introduced on the Cisco Catalyst 9600 Series Supervisor 2 Module.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.

CHAPTER 17

Performing Factory Reset

Prerequisites for Performing a Factory Reset

- Ensure that all the software images, including the current image, configurations, and personal data are backed up before you begin the factory reset process.
- Ensure that there is uninterrupted power supply when the factory reset process is in progress.
- Ensure that In-Service Software Upgrade (ISSU) or In-Service Software Downgrade (ISSD) are not in progress before you begin the factory reset process.

Restrictions for Performing a Factory Reset

- Software patches, if installed on the device, will not be restored after the factory reset process.
- If the **factory-reset** command is issued through a VTY session, the session is not restored after completion of the factory reset process.
- The **config** keyword of the **factory-reset** command is not supported when the switch is in stacking or Stackwise Virtual Link (SVL) mode.
- For modular chassis devices configured in high-availability (HA) mode, factory reset must be applied on each supervisor module.

Information About Performing a Factory Reset

Factory reset erases all the customer-specific data stored in a device and restores the device to its original configuration at the time of shipping. Data that is erased includes configurations, log files, boot variables, core files, and credentials such as Federal Information Processing Standard-related (FIPS-related) keys. The erasure is consistent with the clear method, as described in NIST SP 800-88 Rev. 1.

The factory reset process is used in the following scenarios:

- Return Material Authorization (RMA) for a device: If you have to return a device to Cisco for RMA, remove all the customer-specific data before obtaining an RMA certificate for the device.
- Recovering a compromised device: If the key material or credentials that are stored on a device are compromised, reset the device to the factory configuration, and then reconfigure the device.

During a factory reset, the device reloads and enters ROMMON mode. After the factory reset, the device removes all its environment variables, including the **MAC_ADDRESS** and the **SERIAL_NUMBER** variables, which are required to locate and load the software. Perform a reset in ROMmon mode to automatically set the environment variables. The BAUD rate environment variable returns to its default value after a factory reset. Make sure that the BAUD rate and the console speed are the same at all times. Otherwise, the console becomes unresponsive.

After the system reset in ROMmon mode is complete, add the Cisco IOS image either through an USB or TFTP.

The following table provides details about the data that is erased and retained during the factory reset process:

Table 30: Data Erased and Retained During Factory Reset

Data Erased	Data Retained
All Cisco IOS images, including the current boot image	Data from remote field-replaceable units (FRUs)
Crash information and logs	Value of the configuration register.
User data, startup and running configuration, and contents of removable storage devices, such as Serial Advanced Technology Attachment (SATA), Solid State Drive (SSD), or USB	—
Credentials such as FIPS-related keys	Credentials such as Secure Unique Device Identifier (SUDI) certificates, and public key infrastructure (PKI) keys.
Onboard Failure Logging (OBFL) logs	
ROMmon variables added by a user.	—
Licenses	—

Secure Data Wipe

The device storage is used to maintain software images, device configuration, software logs and operational history. Customer-specific data can be contained in any of these areas. The information can include network architecture and design used by customers.

The **all secure** option in the **factory-reset** command performs data sanitization and securely resets the device. After data sanitization, the device reloads and boots with the software image present in flash.

Secure data wipe feature implements guidelines for media sanitization as described in NIST SP 800-88 Rev. 1.

How to Perform a Factory Reset

To perform a factory reset, complete this procedure:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password, if prompted.

Step 2	• For a standalone device:
	factory-reset {**all** [**secure**] [**3-pass**] | **config** | **boot-vars**}
	• For Cisco StackWise Virtual enabled devices:
	factory-reset {**all** [**secure 3-pass**] | **config** | **boot-vars** | **switch** {*switch-number* | **all** {**all** [**secure 3-pass**] | **config** | **boot-vars**}}
	Example:
	`Device# factory-reset all`
	OR
	`Device# factory-reset switch 1 all config`
		Resets the device to its configuration at the time of its shipping.
		No system configuration is required to use the **factory reset** command.
		The following options are available:
		• **all**: Erases all the content from the NVRAM, all the Cisco IOS images, including the current boot image, boot variables, startup and running configuration data, and user data. We recommend that you use this option.
		• **all secure**: Performs data sanitization and securely resets the device.

	Command or Action	Purpose
	OR `Device# factory-reset all secure`	**Note** • You can use the **all secure** option only on standalone devices. • This option implements guidelines for media sanitization as described in NIST SP 800-88 Rev. 1. • The **factory-reset all secure** command initiates data sanitization. The booted image of the device is retained. • When data sanitization is completed, the device reloads, and the device image is retained in flash if it was booted with an image from the flash. • **secure 3-pass**: Erases all the content from the device with 3-pass overwrite. • Pass 1: Overwrites all addressable locations with binary zeroes. • Pass 2: Overwrites all addressable locations with binary ones. • Pass 3: Overwrites all addressable locations with a random bit pattern. **Note** This option takes approximately thrice the time taken to perform any other option. • **config**: Resets the startup configurations. • **boot-vars**: Resets the user-added boot variables. • **switch** {*switch-number* | **all**}: • *switch-number*: Specifies the switch number. The range is from 1 to 16.

	Command or Action	Purpose
		• **all**: Selects all the switches in the stack. After the factory reset process is successfully completed, the device reboots and enters ROMmon mode.

Configuration Examples for Performing a Factory Reset

The following example shows how to perform a factory reset on a standalone switch:

```
Device> enable
Device# factory-reset all

The factory reset operation is irreversible for all operations. Are you sure? [confirm]
The following will be deleted as a part of factory reset:
1: Crash info and logs
2: User data, startup and running configuration
3: All IOS images, including the current boot image
4: OBFL logs
5: User added rommon variables
6: Data on Field Replaceable Units(USB/SSD/SATA)
The system will reload to perform factory reset.
It will take some time to complete and bring it to rommon.
You will need to load IOS image using USB/TFTP from rommon after
this operation is completed.
DO NOT UNPLUG THE POWER OR INTERRUPT THE OPERATION
Are you sure you want to continue? [confirm]
```

The following examples show how to perform a factory reset on Cisco StackWise Virtual enabled devices:

```
Device> enable
Device# factory-reset switch 2 all
The factory reset operation is irreversible for all operations. Are you sure? [confirm]
 The following will be deleted as a part of factory reset:
 1: Crash info and logs
 2: User data, startup and running configuration
 3: All IOS images, including the current boot image
 4: OBFL logs
 5: User added rommon variables
 6: Data on Field Replaceable Units(USB/SSD/SATA)
 The system will reload to perform factory reset.
 It will take some time to complete and bring it to rommon.
 You will need to load IOS image using USB/TFTP from rommon after
 this operation is completed.
 DO NOT UNPLUG THE POWER OR INTERRUPT THE OPERATION
 Are you sure you want to continue? [confirm]
Switch#
*Sep 23 18:10:42.739:  Successfully sent switch reload message for switch num: 2 and reason
 Factory Reset
*Sep 23 18:10:42.740: %STACKMGR-1-RELOAD: Chassis 2 R0/0: stack_mgr: Reloading due to reason
```

```
 Factory Reset
*Sep 23 18:10:43.158: NGWC_FACTORYRESET: Switch 2, cmd: reset-all success

Original standby Switch 2:
Chassis 2 reloading, reason - Factory Reset
Sep 23 18:11:03.199: %PMAN-5-EXITACTION: R0/0: pvp: Process manager is exiting: process
exit with reload fru code


Enabling factory reset for this reload cycle
 Switch booted with tftp://172.19.72.26/tftpboot/thpaliss/trial.bin
% FACTORYRESET - Started Cleaning Up...

% FACTORYRESET - Unmounting flash1
% FACTORYRESET - Cleaning Up flash1
% FACTORYRESET - In progress.. please wait for completion...

% FACTORYRESET - write zero...
% FACTORYRESET - finish erase

Creating filesystem with 2790400 4k blocks and 697632 inodes
Filesystem UUID: 6a8ec2fb-4602-41b3-9c5c-ed59039d7480
Superblock backups stored on blocks:
 32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208

Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done

% FACTORYRESET - Mounting Back flash1
% FACTORYRESET - Handling Mounted flash1
% FACTORYRESET - Factory Reset Done for flash1

% FACTORYRESET - Unmounting flash2
% FACTORYRESET - Cleaning Up flash2
% FACTORYRESET - In progress.. please wait for completion...

% FACTORYRESET - write zero...
% FACTORYRESET - finish erase

Creating filesystem with 409600 4k blocks and 102544 inodes
Filesystem UUID: e2f2280f-245a-4232-b0a8-edbf590a3107
Superblock backups stored on blocks:
 32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done

% FACTORYRESET - Mounting Back flash2
% FACTORYRESET - Handling Mounted flash2
% FACTORYRESET - Factory Reset Done for flash2

% FACTORYRESET - Unmounting flash3
% FACTORYRESET - Cleaning Up flash3
% FACTORYRESET - In progress.. please wait for completion...

% FACTORYRESET - write zero...
% FACTORYRESET - finish erase

Creating filesystem with 131072 1k blocks and 32768 inodes
Filesystem UUID: 3c548955-16f5-4db5-a1c3-9a956248ccac
Superblock backups stored on blocks:
 8193, 24577, 40961, 57345, 73729
```

```
Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done

% FACTORYRESET - Mounting Back flash3
% FACTORYRESET - Handling Mounted flash3
% FACTORYRESET - Factory Reset Done for flash3

% FACTORYRESET - Unmounting flash7
% FACTORYRESET - Cleaning Up flash7
% FACTORYRESET - In progress.. please wait for completion...

% FACTORYRESET - write zero...
% FACTORYRESET - finish erase

Creating filesystem with 514811 4k blocks and 128768 inodes
Filesystem UUID: 9fe5a9db-263e-4303-825f-78ce815835c2
Superblock backups stored on blocks:
 32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done

% FACTORYRESET - Mounting Back flash7
% FACTORYRESET - Handling Mounted flash7
% FACTORYRESET - Factory Reset Done for flash7
% FACTORYRESET - Lic Clean UP
% FACTORYRESET - Lic Clean Successful...
% FACTORYRESET - Clean Up Successful...


watchdog: watchdog0: watchdog did not stop!
systemd-shutdown[1]: Failed to parse (null): No such file or directory
systemd-shutdown[1]: Failed to deactivate swaps: No such file or directory
```

The following examples show how to perform a factory reset on stacked devices:

```
Device> enable
Device# factory-reset switch all all
The factory reset operation is irreversible for all operations. Are you sure? [confirm]
 The following will be deleted as a part of factory reset:
 1: Crash info and logs
 2: User data, startup and running configuration
 3: All IOS images, including the current boot image
 4: OBFL logs
 5: User added rommon variables
 6: Data on Field Replaceable Units(USB/SSD/SATA)
 The system will reload to perform factory reset.
 It will take some time to complete and bring it to rommon.
 You will need to load IOS image using USB/TFTP from rommon after
 this operation is completed.
 DO NOT UNPLUG THE POWER OR INTERRUPT THE OPERATION
 Are you sure you want to continue? [confirm]
Chassis 1 reloading, reason - Factory Reset

 Protection key not found
9300L#Oct 25 09:53:05.740: %PMAN-5-EXITACTION: F0/0: pvp: Process manager is exiting: reload
 fp action requested
Oct 25 09:53:07.277: %PMAN-5-EXITACTION:vp: Process manager is exiting: rp processes exit
with reload switch code
```

```
Enabling factory reset for this reload cycle
 Switch booted with
tftp://10.5.40.45/cat9k_iosxe.BLD_POLARIS_DEV_LATEST_20191007_224933_V17_2_0_21_2.SSA.bin
 Switch booted via
//10.5.40.45/cat9k_iosxe.BLD_POLARIS_DEV_LATEST_20191007_224933_V17_2_0_21_2.SSA.bin
% FACTORYRESET - Started Cleaning Up...

% FACTORYRESET - Unmounting sd1
% FACTORYRESET - Cleaning Up sd1 [0]
% FACTORYRESET - erase In progress.. please wait for completion...
% FACTORYRESET - write zero...
% FACTORYRESET - finish erase

% FACTORYRESET - Making File System sd1 [0]
Discarding device blocks: done
Creating filesystem with 409600 4k blocks and 102544 inodes
Filesystem UUID: fcf01664-7c6f-41ce-99f0-6df1d941701e
Superblock backups stored on blocks:
 32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done

% FACTORYRESET - Mounting Back sd1 [0]
% FACTORYRESET - Handling Mounted sd1
% FACTORYRESET - Factory Reset Done for sd1

% FACTORYRESET - Unmounting sd3
% FACTORYRESET - Cleaning Up sd3 [0]
% FACTORYRESET - erase In progress.. please wait for completion...
% FACTORYRESET - write zero...



Chassis 2 reloading, reason - Factory Reset
Dec 12 01:02:12.500: %PMAN-5-EXITACTION: F0/0: pvp: Process manager is exiting: reload fp
action requested
De
Enabling factory reset for this reload cycle
 Switch booted with
tftp://10.5.40.45/cat9k_iosxe.BLD_POLARIS_DEV_LATEST_20191007_224933_V17_2_0_21_2.SSA.bin
 Switch booted via
//10.5.40.45/cat9k_iosxe.BLD_POLARIS_DEV_LATEST_20191007_224933_V17_2_0_21_2.SSA.bin
% FACTORYRESET - Started Cleaning Up...
% FACTORYRESET - Unmounting sd1
% FACTORYRESET - Cleaning Up sd1 [0]
% FACTORYRESET - erase In progress.. please wait for completion...
% FACTORYRESET - write zero...



After this the switch will come to boot prompt. Then the customer has to boot the device
from TFTP.
```

The following sample output from the **show platform software factory-reset secure log** command displays the data sanitization report:

```
Device# show platform software factory-reset secure log
Factory reset log:
#CISCO C9200 DATA SANITIZATION REPORT#
START : 18-09-2022, 06:18:44
```

```
  END : 18-09-2022, 06:23:36
-MTD-
PNM : nor
NIST : PURGE
-eMMC-
MID : 'Micron'
PNM : 'Q2J55L'
SN : 0x00000001
NIST : PURGE
```

Additional References for Performing a Factory Reset

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	Command Reference

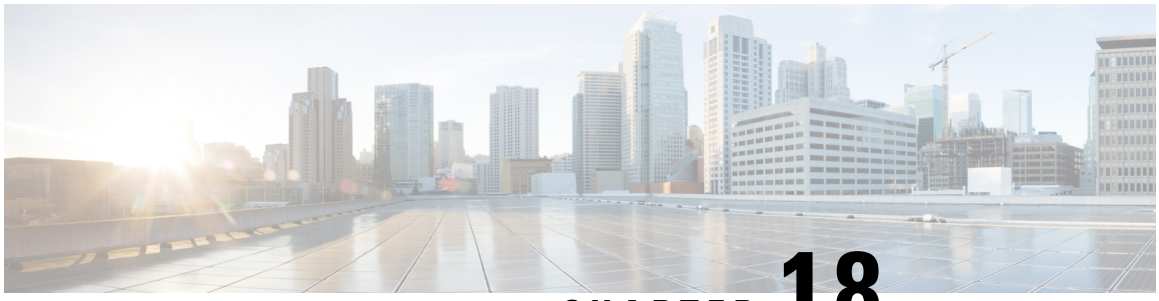
Feature History for Performing a Factory Reset

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.11.1	Factory Reset	Factory reset erases all the customer-specific data stored in a device and restores the device to its original configuration at the time of shipping
Cisco IOS XE Gibraltar 16.12.1	Factory Reset for Removable Storage Devices	Performing a factory reset erases the contents of removable storage devices, such as SATA, SSD, or USB.
Cisco IOS XE Amsterdam 17.2.1	Factory Reset with 3-pass Overwrite	A factory reset can be performed to erase all the content from the device securely with 3-pass overwrite. The **secure 3-pass** keyword was introduced.
	Enhanced Factory Reset Option for Stack and Cisco StackWise Virtual	Support for factory reset on stacked devices and for Cisco StackWise Virtual enabled devices is introduced.
Cisco IOS XE Cupertino 17.7.1	Factory Reset	Support for this feature was introduced only on the Cisco Catalyst 9600 Series Supervisor 2 Module.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.



CHAPTER 18

Configuring Secure Storage

Information About Secure Storage

Secure Storage feature allows you to secure critical configuration information by encrypting it. It encrypts asymmetric key-pairs, pre-shared secrets, the type 6 password encryption key and certain credentials. An instance-unique encryption key is stored in the hardware trust anchor to prevent it from being compromised.

Enabling Secure Storage

Before you begin

By default, this feature is enabled. Perform this procedure only after disabling secure storage on the device.

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters the global configuration mode.
Step 2	**service private-config-encryption** **Example:** `DEvice(config)# service private-config-encryption`	Enables the Secure Storage feature on your device.
Step 3	**end** **Example:**	Returns to privileged EXEC mode.

	Command or Action	Purpose
	`Device(config)# end`	
Step 4	**write memory** **Example:** `Device# write memory`	Encrypts the private-config file and saves the file in an encrypted format.

Disabling Secure Storage

Before you begin

To disable Secure Storage feature on a device, perform this task:

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Device# configure terminal`	Enters the global configuration mode.
Step 2	**no service private-config-encryption** **Example:** `Device(config)# no service private-config-encryption`	Disables the Secure Storage feature on your device. When secure storage is disabled, all the user data is stored in plain text in the NVRAM.
Step 3	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.
Step 4	**write memory** **Example:** `Device# write memory`	Decrypts the private-config file and saves the file in plane format.

Verifying the Status of Encryption

Use the **show parser encrypt file status** command to verify the status of encryption. The following command output indicates that the feature is available but the file is not encrypted. The file is in 'plain text' format.

```
Device#show parser encrypt file status
Feature: Enabled
File Format: Plain Text
Encryption Version: Ver1
```

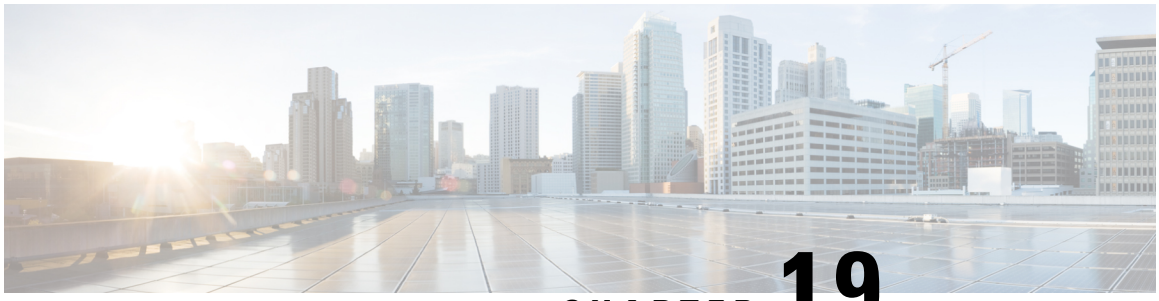
Feature History for Secure Storage

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.11.1	Secure Storage	Secure Storage feature allows you to secure critical configuration information by encrypting it. It encrypts asymmetric key-pairs, pre-shared secrets, the type 6 password encryption key and certain credentials. An instance-unique encryption key is stored in the hardware trust anchor to prevent it from being compromised.
Cisco IOS XE Cupertino 17.7.1	Secure Storage	Support for this feature was introduced on the Cisco Catalyst 9600 Series Supervisor 2 Module (C9600X-SUP-2).

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.



CHAPTER 19

Trace Management

Information About Trace Management

The tracing functionality logs internal events. Trace files are automatically created and saved on the persistent storage device of specific platforms.

If the device has issues, the contents of the trace files are useful to troubleshoot the issue. The trace file outputs provide logs that are used to locate and solve the issue, and helps to get a detailed view of system actions and operations.

To view the recent trace information for a specific process, use the **show logging [process | Profile | process-helper]** command. The **process** keyword uses the first few letters of the name of a process and provides trace logs of the process that starts or matches with the entered string, the **profile** keyword lists the predefined set of process names, and the **profile-helper** keyword displays the available names.

To change the verbosity in a trace message output, you can adjust the trace level of processes using the **set platform software trace level** command. You can choose the **all** keyword to adjust the trace level for all the processes listed or you can select a specific process. When you select a specific process, there's also the option to adjust the trace level for a specific module, or you can use the **all-modules** keyword to adjust all the modules of processes.

Introduction to Binary Tracing

Binary tracing is helpful in gathering trace information with a minimal impact on performance. In binary tracing, the tracing is always on for the system components and a basic level of trace is collected on all the time; thus, the data necessary for troubleshooting a problem has been captured the first time it occurs.

Introduction to Conditional Debugging and Radioactive Tracing

The Conditional Debugging feature allows you to enable debugging and logging for specific features based on the set of conditions you define. This feature is useful in systems where many features are supported.

The Conditional debug allows granular debugging in a network that is operating at a large scale with a large number of features. It allows you to observe detailed debugs for granular instances within the system. This type of debugging is useful when we need to debug only a particular session among thousands of sessions. It's also possible to specify multiple conditions.

A condition refers to a feature or identity, where an identity could be an interface, IP Address, or a MAC address and so on.

Conditional debugging is in contrast to the general debug command, that produces its output without discriminating on the feature objects that are being processed. General debug command consumes numerous system resources and impacts the system performance.

Radioactive tracing provides the ability to form a chain of execution for operations of interest across the system, at an increased verbosity level. This provides a way to print conditionally debug information (up to DEBUG Level or a specified level) across threads, processes, and function calls.

Radioactive Tracing when coupled with Conditional Debugging, provides a single debug command to debug all execution contexts related to the condition. You can execute this command without being aware of the various control flow processes of the feature within the box and without having to issue debugs at these processes individually.

Tracing Levels

Trace level determines the types of traces outputted. Each trace message is assigned a trace level. If the trace level of a process or its module it set as greater than or equal to the level as the trace message, the trace message is displayed otherwise, it's skipped. For example, the default trace level is **Notice** level, so all traces with the **Notice** level and below the notice level are included while the traces above the **Notice** level are excluded.

The following table shows the available tracing levels, and provides descriptions of the message that are displayed with each tracing level. The tracing levels listed in the table are from the lowest to the highest order. The default trace level is **Notice**.

Table 31: Tracing Levels and Descriptions

Tracing Level	Description
Fatal	The message stating the process is aborted.
Emergency	The message is regarding an issue that makes the system unusable.
Alert	The message indicating that an action must be taken immediately.
Critical	The message is regarding a critical event causing loss of important functions.
Error	The message is regarding a system error.
Warning	The message is regarding a system warning.
Notice	The message is regarding a significant event.
Informational	The message is useful for informational purposes only.

Tracing Level	Description
Debug	The message provides debug-level output.
Verbose	All possible trace messages are sent.
Noise	All possible trace messages for the module are logged. The noise level is always equal to the highest possible tracing level. Even if a future enhancement to tracing introduces a higher tracing level, the noise level will become equal to the level of that new enhancement.

Payload Filter

This feature is used to filter trace messages. Trace messages contain actual debug information such as text strings, special characters, and variable arguments (strings), integers, long, IPv4/IPv6/MAC addresses, and so on. Using the payload feature, the trace messages can be filtered based on the selected criteria and without string operations.

You can use the following set and show commands to configure a payload filter and to view the applied filters.

Table 32: Set Commands for Payload Filter

set platform software btrace-manager ... utm-pf enable	Enables and disables the payload filtering feature.
set platform software btrace-manager ... utm-pf disable	
set platform software btrace-manager ... consumer-name <input> create	Creates and deletes consumer/stream.
set platform software btrace-manager ... consumer-name <input> delete	
set platform software btrace-manager ... consumer-name <input> filter <input> add	Applies and removes filter on stream/consumer
set platform software btrace-manager ... consumer-name <input> filter <input> remove	

Table 33: Show Commands for Payload Filter

#show platform software btrace-manager ... utm-pf	Shows the current status of the payload feature and other additional details
show platform software btrace-manager ... utm-pf consumer-name <input> all-filters	Shows all filters currently applied on consumer/stream.

show platform software btrace-manager ... utm-pf consumer-name <input> all-luids	Shows all or selected LUID of consumer for the applied filter.
show platform software btrace-manager ... utm-pf consumer-name <input> filter <input>	
show platform software btrace-manager ... utm-pf message	Shows consumer/stream messages.

How to Configure Conditional Debugging

Conditional Debugging and Radioactive Tracing

Radioactive Tracing when coupled with Conditional Debugging, prpvides a single debug command to debug all execution contexts related to the condition. You can execute this command without being aware of the various control flow processes of the feature within the box and without having to issue debugs at these processes individually.

Configuring Conditional Debugging

Follow the steps to configure conditional debugging:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug platform condition mac** {*mac-address*} **Example:** `Device# debug platform condition mac bc16.6509.3314`	Configures conditional debugging for the MAC Address specified.
Step 3	**debug platform condition start** **Example:** `Device# debug platform condition start`	Starts conditional debugging (this step starts radioactive tracing if there's a match on one of the preceding conditions).

Step 4	**show platform condition** OR **show debug**
	Example:
	`Device# show platform condition`
	`Device# show debug`
		Displays the current conditions set.

	Command or Action	Purpose
Step 5	**debug platform condition stop** **Example:** `Device# debug platform condition stop`	Stops conditional debugging (this step stops radioactive tracing).
Step 6	**request platform software trace archive** [**last** {*number*} **days**] [**target** {*crashinfo:* | *flashinfo:*}] **Example:** `# request platform software trace archive last 2 days`	(Optional) Displays historical logs of merged tracefiles on the system. Filter on any combination of number of days or location.
Step 7	**show platform software trace** [**filter-binary** | **level** | **message**] **Example:** `Device# show platform software trace message`	(Optional) Displays logs merged from the latest trace file. Filter on any combination of application condition, trace module name, and trace level. • **filter-binary** - Filter the modules to be collated • **level** - Show trace levels • **message** - Show trace message ring contents **Note** On the device: • Available from IOS console in addition to linux shell. • Generates a file with merged logs • Displays merged logs only from staging area.
Step 8	**clear platform condition all** **Example:** `Device# clear platform condition all`	Clears all conditions.

What to do next

Note The commands **request platform software trace filter-binary** and **show platform software trace filter-binary** work in a similar way. The only difference is:

- **request platform software trace filter-binary** - Sources the data from historical logs.
- **show platform software trace filter-binary** – Sources the data from the flash Temp directory.

The *mac_log <..date..>* is the important file, as it provides messages for the MAC that is being debugged. The command **show platform software trace filter-binary** also generates the same flash files, and also prints the mac_log on the screen.

Collecting Trace Files

To collect trace files from a device, follow these steps:

1. To request the tracelogs for a specific time period (For example: Five days), use the command:

   Device# **request platform software trace archive last 5 day**

2. The system generates a tar ball (.gz file) of the tracelogs in the location **/flash**:

Copying Archived Trace Files

The following is an example of the trace file for a switching device:

```
Device# dir crashinfo:/tracelogs
Directory of crashinfo:/tracelogs/

50664 -rwx 760 Sep 22 2015 11:12:21 +00:00 plogd_F0-0.bin_0.gz
50603 -rwx 991 Sep 22 2015 11:12:08 +00:00 fed_pmanlog_F0-0.bin_0.9558.20150922111208.gz
50610 -rw- 11 Nov 2 2015 00:15:59 +00:00 timestamp
50611 -rwx 1443 Sep 22 2015 11:11:31 +00:00
auto_upgrade_client_sh_pmanlog_R0-.bin_0.3817.20150922111130.gz
50669 -rwx 589 Sep 30 2015 03:59:04 +00:00 cfgwr-8021_R0-0.bin_0.gz
50612 -rwx 1136 Sep 22 2015 11:11:46 +00:00 reflector_803_R0-0.bin_0.1312.20150922111116.gz
50794 -rwx 4239 Nov 2 2015 00:04:32 +00:00 IOSRP_R0-0.bin_0.14239.20151101234827.gz
50615 -rwx 131072 Nov 2 2015 00:19:59 +00:00 linux_iosd_image_pmanlog_R0-0.bin_0
--More—
```

You can copy the trace files using one of the following options:

```
Device# copy crashinfo:/tracelogs ?
  crashinfo: Copy to crashinfo: file system
  flash: Copy to flash: file system
  ftp: Copy to ftp: file system
  http: Copy to http: file system
  https: Copy to https: file system
  null: Copy to null: file system
  nvram: Copy to nvram: file system
  rcp: Copy to rcp: file system
  running-config Update (merge with) current system configuration
  scp: Copy to scp: file system
  startup-config Copy to startup configuration
  syslog: Copy to syslog: file system
  system: Copy to system: file system
  tftp: Copy to tftp: file system
  tmpsys: Copy to tmpsys: file system
```

The general syntax for copying onto a TFTP server is as follows:

```
Device# copy source: tftp:
Device# copy crashinfo:/tracelogs/IOSRP_R0-0.bin_0.14239.20151101234827.gz tftp:
Address or name of remote host []? 2.2.2.2
Destination filename [IOSRP_R0-0.bin_0.14239.20151101234827.gz]?
```

Note It's important to clear the generated report or archive files off the device so that there's flash space available for tracelog and other purposes.

Configuring Payload Filter

To configure a payload filter, you must create a consumer and add the relevant payload filter data.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

Step 2	**set platform software btrace-manager utm-pf enable**
	Example:
	`Device# set platform software btrace-manager chassis active r0 utm-pf enable`
	`Device# set platform software btrace-manager chassis active r0 utm-pf disable`
		Enables or disables the payload filter.

Step 3	**set platform software btrace-manager** {*consumer-name*} **create** **Example:** `Device# set platform software btrace-manager chassis active r0 consumer-name utm_pf_test create`	Creates a consumer name.
Step 4	**set platform software btrace-manager consumer** {*consumer-name*} **filter** {*input*} **add** **Example:** `Device# set platform software btrace-manager chassis active r0 consumer-name utm_pf_test filter "Failed to retrieve an interface" add`	Add a filter data.

Configuration Examples for Trace Management

The following is an output example of the *show platform condition* command.

The following is a sample of the *debug platform condition stop* command.

```
Device# debug platform condition stop
Conditional Debug Global State: Stop
```

The following is an example of the **show logging** command for the **ios** process.

```
Device# show logging process ios
Logging display requested on 2022/10/27 09:32:06 (PDT) for Hostname: [vwlc_1_9222], Model:
 [C9800-CL-K9], Version: [17.11.01], SN: [9ZY0U03YBM0], MD_SN: [9ZY0U03YBM0]

Displaying logs from the last 0 days, 0 hours, 10 minutes, 0 seconds
executing cmd on chassis 1 ...
Unified Decoder Library Init .. DONE
Found 1 UTF Streams

2022/10/27 09:31:52.835197577 {iosrp_R0-0}{1}: [parser_cmd] [26471]: (note): id=
console@console:user= cmd: 'show logging process ios' SUCCESS 2022/10/27 08:31:48.762 PST
2022/10/27 09:31:59.651965736 {iosrp_R0-0}{1}: [parser_cmd] [26471]: (note): id=
console@console:user= cmd: 'show logging process ios internal' SUCCESS 2022/10/27 08:31:56.485
 PST
============================================================
======= Unified Trace Decoder Information/Statistics ======
============================================================
----------------- Decoder Input Information ---------------
============================================================
 Num of Unique Streams .. 1
 Total UTF To Process ... 1
 Total UTM To Process ... 75403
 UTM Process Filter ..... ios
 MRST Filter Rules ...... 4
============================================================
----------------- Decoder Output Information --------------
============================================================
 First UTM TimeStamp ............... 2022/10/27 02:21:47.048461994
 Last UTM TimeStamp ................ 2022/10/27 09:32:04.919540850
 UTM [Skipped / Rendered / Total] .. 75401 / 2 / 75403
 UTM [ENCODED] ..................... 75266
 UTM [PLAIN TEXT] .................. 94
 UTM [DYN LIB] ..................... 0
 UTM [MODULE ID] ................... 0
 UTM [TDL TAN] ..................... 43
 UTM [APP CONTEXT] ................. 0
 UTM [MARKER] ...................... 0
 UTM [PCAP] ........................ 0
 UTM [LUID NOT FOUND] .............. 0
============================================================
```

The following is an example of the **show logging profile wireless** command.

```
Device# show logging profile wireless
Logging display requested on 2023/03/13 09:07:09 (UTC) for Hostname: [FABRIEK], Model:
[C8300-1N1S-4T2X], Version: [17.12.01], SN: [FDO24190V85], MD_SN: [FDO2451M13G]

Displaying logs from the last 0 days, 0 hours, 10 minutes, 0 seconds
executing cmd on chassis local ...
Unified Decoder Library Init .. DONE
Found 1 UTF Streams

2023/03/13 08:57:34.084609935 {iosrp_R0-0}{255}: [parser_cmd] [3793]: (note): id=
10.68.219.145@vty0:user= cmd: 'show logging profile wireless level info' SUCCESS 2023/03/13
 08:57:31.376 UTC
```

```
2023/03/13 09:07:03.562290152 {iosrp_R0-0}{255}: [parser_cmd] [3793]: (note): id=
10.68.219.145@vty0:user= cmd: 'show logging profile wireless internal ' SUCCESS 2023/03/13
 08:58:51.922 UTC
============================================================
======= Unified Trace Decoder Information/Statistics ======
============================================================
----------------- Decoder Input Information ---------------
============================================================
 Num of Unique Streams .. 1
 Total UTF To Process ... 1
 Total UTM To Process ... 55410
 UTM Process Filter .....
qpp_cp,fman_fp,fman_rp,fman-rp,fman-fp,fed,mobilityd,mspd,odm_proxy,rogued,rrm,repm,wncd,wncmgrd,wncd_x,IOSRP,snd,odm,wstatsd,linux-iosd-image,wncloudn,locationd

 MRST Filter Rules ...... 24
============================================================
----------------- Decoder Output Information --------------
============================================================
 First UTM TimeStamp ............... 2023/03/13 08:13:19.321653302
 Last UTM TimeStamp ................ 2023/03/13 09:07:08.462269864
 UTM [Skipped / Rendered / Total] .. 55408 / 2 / 55410
 UTM [ENCODED] ..................... 2
 UTM [PLAIN TEXT] .................. 0
 UTM [DYN LIB] ..................... 0
 UTM [MODULE ID] ................... 0
 UTM [TDL TAN] ..................... 0
 UTM [APP CONTEXT] ................. 0
 UTM [MARKER] ...................... 0
 UTM [PCAP] ........................ 0
 UTM [LUID NOT FOUND] .............. 0
 UTM Level [EMERGENCY / ALERT / CRITICAL / ERROR] .. 0 / 0 / 0 / 0
 UTM Level [WARNING / NOTICE / INFO / DEBUG] ....... 0 / 2 / 0 / 0
 UTM Level [VERBOSE / NOISE / INVALID] ............. 0 / 0 / 0
============================================================
```

The following is an example of the **show logging process-helper** command.

```
Device# show logging process-helper ios
Logging display requested on 2023/03/13 10:30:29 (UTC) for Hostname: [FABRIEK], Model:
[C8300-1N1S-4T2X], Version: [17.12.01], SN: [FDO24190V85], MD_SN: [FDO2451M13G]

Displaying logs from the last 0 days, 0 hours, 10 minutes, 0 seconds
executing cmd on chassis local ...
Unified Decoder Library Init .. DONE
Found 1 UTF Streams

2023/03/13 10:30:16.884663022 {iosrp_R0-0}{255}: [parser_cmd] [3793]: (note): id=
10.68.219.145@vty0:user= cmd: 'enable' SUCCESS 2023/03/13 10:30:10.721 UTC
============================================================
======= Unified Trace Decoder Information/Statistics ======
============================================================
----------------- Decoder Input Information ---------------
============================================================
 Num of Unique Streams .. 1
 Total UTF To Process ... 1
 Total UTM To Process ... 88985
 UTM Process Filter ..... IOSRP
 MRST Filter Rules ...... 1
============================================================
----------------- Decoder Output Information --------------
============================================================
 First UTM TimeStamp ............... 2023/03/13 08:13:19.321653302
 Last UTM TimeStamp ................ 2023/03/13 10:30:27.267645695
```

```
UTM [Skipped / Rendered / Total] .. 88984 / 1 / 88985
UTM [ENCODED] ..................... 1
UTM [PLAIN TEXT] .................. 0
UTM [DYN LIB] ..................... 0
UTM [MODULE ID] ................... 0
UTM [TDL TAN] ..................... 0
UTM [APP CONTEXT] ................. 0
UTM [MARKER] ...................... 0
UTM [PCAP] ........................ 0
UTM [LUID NOT FOUND] .............. 0
UTM Level [EMERGENCY / ALERT / CRITICAL / ERROR] .. 0 / 0 / 0 / 0
UTM Level [WARNING / NOTICE / INFO / DEBUG] ....... 0 / 1 / 0 / 0
UTM Level [VERBOSE / NOISE / INVALID] ............. 0 / 0 / 0
============================================================
```

Additional References for Trace Management

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	Command Reference Guide for catalyst 9K platforms.

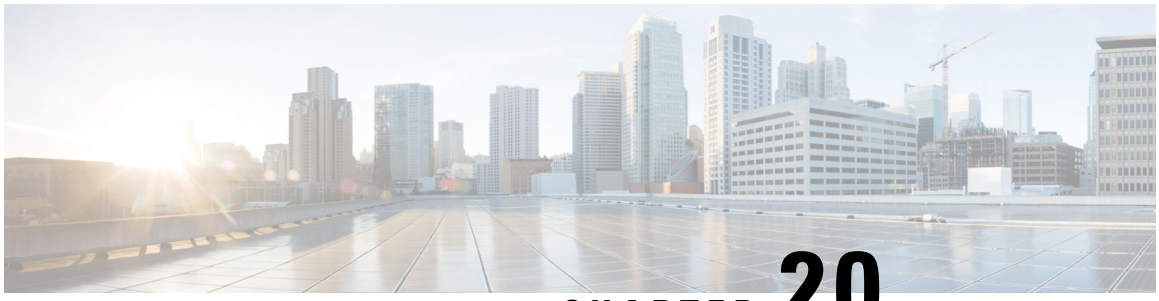
Feature History for Trace Management

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Fuji 16.9.2	Conditional Debugging and Radioactive Tracing	The Conditional Debugging feature allows you to selectively enable debugging and logging for specific features based on the set of conditions you define.
Cisco IOS XE Cupertino 17.7.x	Binary Tracing	Binary tracing helps in gathering of trace information with a minimal impact on performance.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.



CHAPTER 20

Consent Token

Restrictions for Consent Token

- Consent Token is enabled by default and cannot be disabled.
- After the challenge has been sent from the device, the response needs to be entered within 30 minutes. If it is not entered, the challenge expires and a new challenge must be requested.
- A single response is valid only for one time for a corresponding challenge.
- The maximum authorization timeout for root-shell access is seven days.
- After a switchover event, all the existing Consent Token based authorizations would be treated as expired. You must then restart a fresh authentication sequence for service access.
- Only Cisco authorized personnel have access to Consent Token response generation on Cisco's challenge signing server.
- In System Shell access scenario, exiting the shell does not terminate authorization until the authorization timeout occurs or the shell authorization is explicitly terminated by the consent token terminate authorization command.

  We recommend that you force terminate System Shell authorization by explicitly issuing the Consent Token terminate command once the purpose of System Shell access is complete.

Information About Consent Token

Consent Token is a security feature that is used to authenticate the network administrator of an organization to access system shell with mutual consent from the network administrator and Cisco Technical Assistance Centre (Cisco TAC).

In some debugging scenarios, the Cisco TAC engineer may have to collect certain debug information or perform live debug on a production system. In such cases, the Cisco TAC engineer will ask you (the network

administrator) to access system shell on your device. Consent Token is a lock, unlock and re-lock mechanism that provides you with privileged, restricted, and secure access to the system shell.

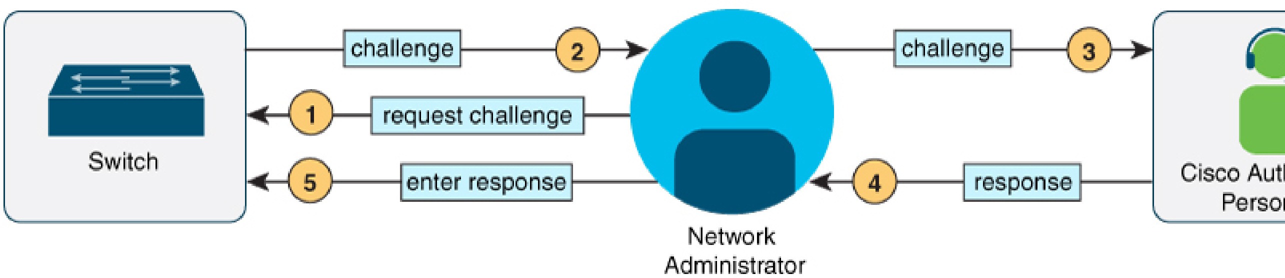
When you request access to system shell, you need to be authorized. You must first run the command to generate a challenge using the Consent Token feature on your device. The device generates a unique challenge as output. You must then copy this challenge string and send it to a Cisco Authorized Personnel through e-mail or Instant Message.

The Cisco Authorized Personnel processes the unique challenge string and generates a response that is unique. The Cisco Authorized Personnel copies this response string and sends it to you through e-mail or Instant Message.

You must then input this response string into your device. If the challenge-response pair match, you are authorized to access system shell. If not, an error is displayed and you are required to repeat the authentication process.

Once you gain access to system shell, collect the debug information required by the Cisco TAC engineer. After you are done accessing system shell, terminate the session and continue the debugging process.

Figure 13: Consent Token



Consent Token Authorization Process for System Shell Access

This section describes the process of Consent Token authorization to access system shell:

Procedure

Step 1 Generate a challenge requesting for access to system shell for the specified time period.

Example:

```
Device# request consent-token generate-challenge shell-access auth-timeout 900
zSSdrAAAAQEBAAQAAAABAgAEAAAAAAMACH86csUmD10BAAQ0Fvd7CxqRYUeoD7B4AwW7QUABAAAAG8GAAhDVEFfREVNTwcAGENUQV9ERU1PX0NUQV9TSUdOSU5HX0tFWQgAC0M5ODAwLUNMUs5CQALOVpQUEVESE5KRkI=
Device#
*Jan 18 02:47:06.733: %CTOKEN-6-AUTH_UPDATE: Consent Token Update (challenge generation
attempt: Shell access 0).
```

Send a request for a challenge using the **request consent-token generate-challenge shell-access** *time-validity-slot* command. The duration in minutes for which you are requesting access to system shell is the time-slot-period.

In this example, the time period is 900 minutes after which the session expires.

The device generates a unique challenge as output. This challenge is a base-64 format string.

Step 2 Send the challenge string to a Cisco Authorized Personnel.

Send the challenge string generated by the device to a Cisco Authorized Personnel through e-mail or Instant Message.

The Cisco Authorized Personnel processes the unique challenge string and generates a response. The response is also a base-64 string that is unique. The Cisco Authorized Personnel copies this response string and sends it to you through e-mail or Instant Message.

Step 3 Input the response string onto your device.

Example:

```
Device# request consent-token accept-response shell-access
[redacted]
% Consent token authorization success
*Jan 18 02:51:37.807: %CTOKEN-6-AUTH_UPDATE: Consent Token Update (authentication success:
 Shell access 0).

Device# request platform software system shell
Activity within this shell can jeopardize the functioning of the system.
Are you sure you want to continue? [y/n] y
Device#
*Jan 18 02:56:59.714: %CTOKEN-6-AUTH_UPDATE: Consent Token Update (authorization for Shell
 access 0 will expire in 10 min).
```

Input the response string sent to you by the Cisco Authorized Personnel using the **request consent-token accept-response shell-access** *response-string* command.

If the challenge-response pair match, you are authorized to access system shell. If the challenge-response pair do not match, an error is displayed and you are required to repeat steps 1 to 3.

After you are authorized, you can access system shell for the requested time-slot.

The device sends a message when there is ten minutes remaining of the authorization session.

Step 4 Terminate the session.

Example:

```
Device# request consent-token terminate-auth
% Consent token authorization termination success

Device#
*Jan 18 23:33:02.937: %CTOKEN-6-AUTH_UPDATE: Consent Token Update (terminate authentication:
 Shell access 0).
Device#
```

When you finish accessing system shell, you can end the session using the **request consent-token terminate-auth** command. You can also force terminate the session prior to the authorization timeout using this command. The session also gets terminated automatically when the requested time slot expires.

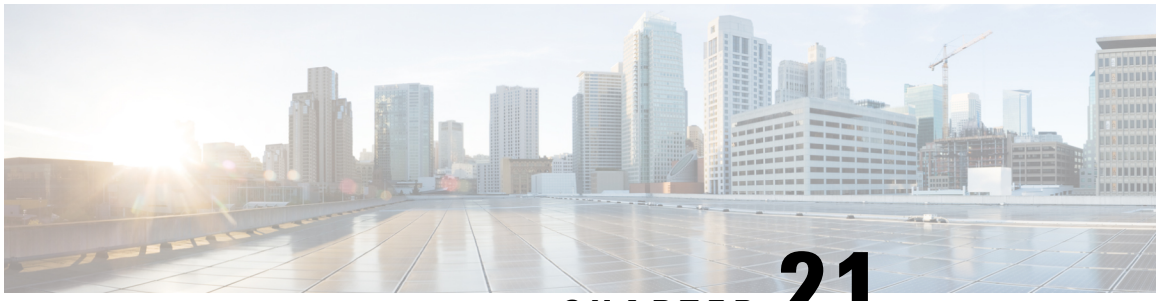
Feature History for Consent Token

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.11.1	Consent Token	Consent Token is a security feature that is used to authenticate the network administrator of an organization to access system shell with mutual consent from the network administrator and Cisco Technical Assistance Centre (Cisco TAC).
Cisco IOS XE Cupertino 17.7.1	Consent Token	Support for this feature was introduced on the Cisco Catalyst 9600 Series Supervisor 2 Module.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.



CHAPTER 21

Troubleshooting the Software Configuration

This chapter describes how to identify and resolve software problems related to the Cisco IOS software on the switch. Depending on the nature of the problem, you can use the command-line interface (CLI), Device Manager, or Network Assistant to identify and solve problems.

Additional troubleshooting information, such as LED descriptions, is provided in the hardware installation guide.

Information About Troubleshooting the Software Configuration

Software Failure on a Switch

Switch software can be corrupted during an upgrade by downloading the incorrect file to the switch, and by deleting the image file. In all of these cases, there is no connectivity.

Lost or Forgotten Password on a Device

The default configuration for the device allows an end user with physical access to the device to recover from a lost password by interrupting the boot process during power-on and by entering a new password. These recovery procedures require that you have physical access to the device.

Note On these devices, a system administrator can disable some of the functionality of this feature by allowing an end user to reset a password only by agreeing to return to the default configuration. If you are an end user trying to reset a password when password recovery has been disabled, a status message reminds you to return to the default configuration during the recovery process.

Note You cannot recover encryption password key, when Cisco WLC configuration is copied from one Cisco WLC to another (in case of an RMA).

Follow the steps described in the section Recovering from a Lost or Forgotten Password, on page 436 to recover from a lost or forgotten password.

Ping

The device supports IP ping, which you can use to test connectivity to remote hosts. Ping sends an echo request packet to an address and waits for a reply. Ping returns one of these responses:

- Normal response—The normal response (*hostname* is alive) occurs in 1 to 10 seconds, depending on network traffic.
- Destination does not respond—If the host does not respond, a *no-answer* message is returned.
- Unknown host—If the host does not exist, an *unknown host* message is returned.
- Destination unreachable—If the default gateway cannot reach the specified network, a *destination-unreachable* message is returned.
- Network or host unreachable—If there is no entry in the route table for the host or network, a *network or host unreachable* message is returned.

Refere the section Executing Ping, on page 440 to understand how **ping** works.

Layer 2 Traceroute

The Layer 2 traceroute feature allows the switch to identify the physical path that a packet takes from a source device to a destination device. Layer 2 traceroute supports only unicast source and destination MAC addresses. Traceroute finds the path by using the MAC address tables of the devices in the path. When the Device detects a device in the path that does not support Layer 2 traceroute, the Device continues to send Layer 2 trace queries and lets them time out.

The Device can only identify the path from the source device to the destination device. It cannot identify the path that a packet takes from source host to the source device or from the destination device to the destination host.

Layer 2 Traceroute Guidelines

- Cisco Discovery Protocol (CDP) must be enabled on all the devices in the network. For Layer 2 traceroute to function properly, do not disable CDP.

  If any devices in the physical path are transparent to CDP, the switch cannot identify the path through these devices.

- A device is reachable from another device when you can test connectivity by using the **ping** privileged EXEC command. All devices in the physical path must be reachable from each other.
- The maximum number of hops identified in the path is ten.

- You can enter the **traceroute mac** or the **traceroute mac ip** privileged EXEC command on a device that is not in the physical path from the source device to the destination device. All devices in the path must be reachable from this switch.
- The **traceroute mac** command output shows the Layer 2 path only when the specified source and destination MAC addresses belong to the same VLAN. If you specify source and destination MAC addresses that belong to different VLANs, the Layer 2 path is not identified, and an error message appears.
- If you specify a multicast source or destination MAC address, the path is not identified, and an error message appears.
- If the source or destination MAC address belongs to multiple VLANs, you must specify the VLAN to which both the source and destination MAC addresses belong. If the VLAN is not specified, the path is not identified, and an error message appears.
- The **traceroute mac ip** command output shows the Layer 2 path when the specified source and destination IP addresses belong to the same subnet. When you specify the IP addresses, the device uses the Address Resolution Protocol (ARP) to associate the IP addresses with the corresponding MAC addresses and the VLAN IDs.
  - If an ARP entry exists for the specified IP address, the device uses the associated MAC address and identifies the physical path.
  - If an ARP entry does not exist, the device sends an ARP query and tries to resolve the IP address. If the IP address is not resolved, the path is not identified, and an error message appears.
- When multiple devices are attached to one port through hubs (for example, multiple CDP neighbors are detected on a port), the Layer 2 traceroute feature is not supported. When more than one CDP neighbor is detected on a port, the Layer 2 path is not identified, and an error message appears.
- This feature is not supported in Token Ring VLANs.
- Layer 2 traceroute opens a listening socket on the User Datagram Protocol (UDP) port 2228 that can be accessed remotely with any IPv4 address, and does not require any authentication. This UDP socket allows to read VLAN information, links, presence of particular MAC addresses, and CDP neighbor information, from the device. This information can be used to eventually build a complete picture of the Layer 2 network topology.
- Layer 2 traceroute is enabled by default and can be disabled by running the **no l2 traceroute** command in global configuration mode. To re-enable Layer 2 traceroute, use the **l2 traceroute** command in global configuration mode.

IP Traceroute

You can use IP traceroute to identify the path that packets take through the network on a hop-by-hop basis. The command output displays all network layer (Layer 3) devices, such as routers, that the traffic passes through on the way to the destination.

Your Device can participate as the source or destination of the **traceroute** privileged EXEC command and might or might not appear as a hop in the **traceroute** command output. If the Device is the destination of the traceroute, it is displayed as the final destination in the traceroute output. Intermediate devices do not show up in the traceroute output if they are only bridging the packet from one port to another within the same VLAN. However, if the intermediate Device is a multilayer Device that is routing a particular packet, this device shows up as a hop in the traceroute output.

The **traceroute** privileged EXEC command uses the Time To Live (TTL) field in the IP header to cause routers and servers to generate specific return messages. Traceroute starts by sending a User Datagram Protocol (UDP) datagram to the destination host with the TTL field set to 1. If a router finds a TTL value of 1 or 0, it drops the datagram and sends an Internet Control Message Protocol (ICMP) time-to-live-exceeded message to the sender. Traceroute finds the address of the first hop by examining the source address field of the ICMP time-to-live-exceeded message.

To identify the next hop, traceroute sends a UDP packet with a TTL value of 2. The first router decrements the TTL field by 1 and sends the datagram to the next router. The second router sees a TTL value of 1, discards the datagram, and returns the time-to-live-exceeded message to the source. This process continues until the TTL is incremented to a value large enough for the datagram to reach the destination host (or until the maximum TTL is reached).

To learn when a datagram reaches its destination, traceroute sets the UDP destination port number in the datagram to a very large value that the destination host is unlikely to be using. When a host receives a datagram destined to itself containing a destination port number that is unused locally, it sends an ICMP *port-unreachable* error to the source. Because all errors except port-unreachable errors come from intermediate hops, the receipt of a port-unreachable error means that this message was sent by the destination port.

Go to Example: Performing a Traceroute to an IP Host, on page 445 to see an example of IP traceroute process.

Debug Commands

Caution Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco technical support staff. It is best to use **debug** commands during periods of lower network traffic and fewer users. Debugging during these periods decreases the likelihood that increased **debug** command processing overhead will affect system use.

All **debug** commands are entered in privileged EXEC mode, and most **debug** commands take no arguments.

System Report

System reports or crashinfo files save information that helps Cisco technical support representatives to debug problems that caused the Cisco IOS image to fail (crash). It is necessary to quickly and reliably collect critical crash information with high fidelity and integrity. Further, it is necessary to collect this information and bundle it in a way that it can be associated or identified with a specific crash occurrence.

System reports are generated in case of a switchover: System reports are generated only on high availability (HA) member switches. Reports are not generated for non-HA members.

The system does not generate reports in case of a reload.

During a process crash, the following is collected locally from the switch:

1. Full process core
2. Tracelogs
3. IOS syslogs (not guaranteed in case of non-active crashes)
4. System process information

5. Bootup logs
6. Reload logs
7. Certain types of /proc information

This information is stored in separate files which are then archived and compressed into one bundle. This makes it convenient to get a crash snapshot in one place, and can be then moved off the box for analysis. This report is generated before the switch goes down to rommon/bootloader.

Except for the full core and tracelogs, everything else is a text file.

Use the **request platform software process core fed switch active** command to generate the core dump.

```
Device# request platform software process core fed switch active
SUCCESS: Core file generated.

Device# dir bootflash:/core
Directory of bootflash:/core/
16430   -rw-         10941657   Apr 6 2022 00:15:20 +00:00
Switch_1_RP_0_fed_18469_20220406-001511-UTC.core.gz
16812   -rw-               1   Apr 6 2022 00:01:48 +00:00  .callhome
16810   drwx            4096  Jan 18 2022 21:10:35 +00:00  modules
```

Crashinfo Files

By default the system report file will be generated and saved into the /crashinfo directory. Ifit cannot be saved to the crashinfo partition for lack of space, then it will be saved to the /flash directory.

To display the files, enter the **dir crashinfo:** command. The following is sample output of a crashinfo directory:

System reports are located in the crashinfo directory in the following format:

```
system-report_[switch number]_[date]-[timestamp]-UTC.gz
```

After a switch crashes, check for a system report file. The name of the most recently generated system report file is stored in the last_systemreport file under the crashinfo directory. The system report and crashinfo files assist TAC while troubleshooting the issue.

The system report generated can be further copied using TFTP, HTTP and few other options.

```
Device# copy crashinfo: ?
crashinfo:     Copy to crashinfo: file system
flash:         Copy to flash: file system
ftp:           Copy to ftp: file system
http:          Copy to http: file system
https:         Copy to https: file system
null:          Copy to null: file system
nvram:         Copy to nvram: file system
rcp:           Copy to rcp: file system
running-config Update (merge with) current system configuration
scp:           Copy to scp: file system
startup-config Copy to startup configuration
syslog:        Copy to syslog: file system
system:        Copy to system: file system
tftp:          Copy to tftp: file system
tmpsys:        Copy to tmpsys: file system
```

The general syntax for copying onto TFTP server is as follows:

```
Device# copy crashinfo: tftp:
Source filename [system-report_1_20150909-092728-UTC.gz]?
Address or name of remote host []? 1.1.1.1
Destination filename [system-report_1_20150909-092728-UTC.gz]?
```

The tracelogs can be collected by issuing a trace archive command. This command provides time period options. The command syntax is as follows:

```
Device# request platform software trace archive ?
last     Archive trace files of last x days
target   Location and name for the archive file
```

The tracelogs stored in crashinfo: or flash: directory from within the last 3650 days can be collected.

```
Device# request platform software trace archive last ?
<1-3650> Number of days (1-3650)
Switch#request platform software trace archive last 3650 days target ?
crashinfo: Archive file name and location
flash:     Archive file name and location
```

Note It is important to clear the system reports or trace archives from flash or crashinfo directory once they are copied out, in order to have space available for tracelogs and other purposes.

Onboard Failure Logging on the Switch

You can use the onboard failure logging (OBFL) feature to collect information about the device. The information includes uptime, temperature, and voltage information and helps Cisco technical support representatives to troubleshoot device problems. We recommend that you keep OBFL enabled and do not erase the data stored in the flash memory.

By default, OBFL is enabled. It collects information about the device and small form-factor pluggable (SFP) modules. The device stores this information in the flash memory:

- CLI commands—Record of the OBFL CLI commands that are entered on a standalone device.
- Message—Record of the hardware-related system messages generated by a standalone device .
- Power over Ethernet (PoE)—Record of the power consumption of PoE ports on a standalone device .
- Temperature—Temperature of a standalone deicev .
- Uptime data—Time when a standalone device starts, the reason the device restarts, and the length of time the device has been running since it last restarted.
- Voltage—System voltages of a standalone device .

You should manually set the system clock or configure it by using Network Time Protocol (NTP).

When the device is running, you can retrieve the OBFL data by using the **show logging onboard** privileged EXEC commands. If the device fails, contact your Cisco technical support representative to find out how to retrieve the data.

When an OBFL-enabled device is restarted, there is a 10-minute delay before logging of new data begins.

Fan Failures

By default, the feature is disabled. When more than one of the fans fails in a field-replaceable unit (FRU) or in a power supply, the device does not shut down, and this error message appears:

The device might overheat and shut down.

To restart the device, it must be power cycled.

Possible Symptoms of High CPU Utilization

Excessive CPU utilization might result in these symptoms, but the symptoms might also result from other causes, some of which are the following:

- Spanning tree topology changes
- EtherChannel links brought down due to loss of communication
- Failure to respond to management requests (ICMP ping, SNMP timeouts, slow Telnet or SSH sessions)
- UDLD flapping
- IP SLAs failures because of SLAs responses beyond an acceptable threshold
- DHCP or IEEE 802.1x failures if the switch does not forward or respond to requests

How to Troubleshoot the Software Configuration

Booting from the Recovery Partition

Cisco Catalyst 9200CX Series Switches support booting from the recovery partition. This is beneficial to end users if they face an issue while trying to boot the switch from Flash or an external device, such as USB or SDflash. The recovery image is the same as the recommended Cisco IOS image for the switch, and is stored in a partition named **drec0**.

Note You can not access recovery partition when the switch is in Cisco IOS prompt. Note that the factory-reset process does not erase this image.

To check the partition image name, enter **dir drec0:**

```
switch: dir drec0:

Attributes        Size         Name

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

-rw-r--r--   490586943   cat9k_lite_iosxe.17.09.01.SPA.bin

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

switch:
```

To boot from the recovery partition, enter **boot drec0:<image name>**:

```
switch: boot drec0:cat9k_lite_iosxe.17.09.01.SPA.bin

boot: attempting to boot from [drec0:cat9k_lite_iosxe.17.09.01prd9.SPA.bin]
boot: reading file cat9k_lite_iosxe.17.09.01.SPA.bin
##############################
```

Recovering from a Lost or Forgotten Password

The default configuration for the switch allows an end user with physical access to the switch to recover from a lost password by interrupting the boot process during power-on and by entering a new password. These recovery procedures require that you have physical access to the switch.

Note On these switches, a system administrator can disable some of the functionality of this feature by allowing an end user to reset a password only by agreeing to return to the default configuration. If you are an end user trying to reset a password when password recovery has been disabled, a status message shows this during the recovery process.

Procedure

Step 1 Connect a terminal or PC to the switch.

- Connect a terminal or a PC with terminal-emulation software to the switch console port.
- Connect a PC to the Ethernet management port.

Step 2 Set the line speed on the emulation software to 9600 baud.

Step 3 Power off the standalone switch or the entire switch stack.

Step 4 Reconnect the power cord to the switch or the active switch. For a device with dual supervisor module, remove the standy supervisor from the chassis before the password recovery procedure. Reconnect the power cord to the switch or the active supervisor module. Press Ctrl-C to prevent autoboot and to get into ROMMON mode while the switch or the active supervisor module is booting up.

Proceed to the *Procedure with Password Recovery Enabled* section, and follow the steps.

Step 5 After recovering the password, reload the switch or the active switch.

On a switch:

```
Switch> reload
Proceed with reload? [confirm] y
```

Procedure with Password Recovery Enabled

Procedure

Step 1 Enable manual boot mode.

```
Device: MANUAL_BOOT=yes
```

Step 2 Ignore the startup configuration with the following command:

```
Device: SWITCH_IGNORE_STARTUP_CFG=1
```

Step 3 Boot the switch with the *packages.conf* file from flash.

```
Device: boot flash:packages.conf
```

Step 4 Terminate the initial configuration dialog by answering **No**.

```
Would you like to enter the initial configuration dialog? [yes/no]: No
```

Step 5 At the switch prompt, enter privileged EXEC mode.

```
Device> enable
Device#
```

Step 6 Copy the startup configuration to running configuration.

```
Device# copy startup-config running-config Destination filename [running-config]?
```

Press Return in response to the confirmation prompts. The configuration file is now reloaded, and you can change the password.

Step 7 Enter global configuration mode and change the **enable** password.

```
Device# configure terminal
Device(config)# enable secret password
```

Step 8 Set the SWITCH_IGNORE_STARTUP_CFG parameter to 0.

```
Device(config)# no system ignore startupconfig switch all
Device(config)# end
```

Step 9 Write the running configuration to the startup configuration file and save the configuration.

```
Device# copy running-config startup-config

Device# write memory
```

Step 10 Confirm that manual boot mode is enabled.

```
Device# show boot

 BOOT variable = flash:packages.conf;
 Manual Boot = yes
 Enable Break = yes
```

Step 11 Reload the device.

```
Device# reload
```

Step 12 Boot the device with the *packages.conf* file from flash.

```
Device: boot flash:packages.conf
```

Step 13 After the device boots up, disable manual boot on the device.

```
Device(config)# no boot manual
```

Procedure with Password Recovery Disabled

If the password-recovery mechanism is disabled, this message appears:

```
The password-recovery mechanism has been triggered, but
is currently disabled.  Access to the boot loader prompt
through the password-recovery mechanism is disallowed at
this point.  However, if you agree to let the system be
reset back to the default system configuration, access
to the boot loader prompt can still be allowed.

Would you like to reset the system back to the default configuration (y/n)?
```

Caution Returning the device to the default configuration results in the loss of all existing configurations. We recommend that you contact your system administrator to verify if there are backup device and VLAN configuration files.

- If you enter **n** (no), the normal boot process continues as if the **Mode** button had not been pressed; you cannot access the boot loader prompt, and you cannot enter a new password. You see the message:

  ```
  Press Enter to continue........
  ```

- If you enter **y** (yes), the configuration file in flash memory and the VLAN database file are deleted. When the default configuration loads, you can reset the password.

Procedure

Step 1 Choose to continue with password recovery and delete the existing configuration:

```
Would you like to reset the system back to the default configuration (y/n)? Y
```

Step 2 Display the contents of flash memory:

```
Device: dir flash:
```

The device file system appears.

Step 3 Boot up the system:

```
Device: boot
```

You are prompted to start the setup program. To continue with password recovery, enter **N** at the prompt:

```
Continue with the configuration dialog? [yes/no]: N
```

Step 4 At the device prompt, enter privileged EXEC mode:

```
Device> enable
```

Step 5 Enter global configuration mode:

```
Device# configure terminal
```

Step 6 Change the password:

```
Device(config)# enable secret password
```

The secret password can be from 1 to 25 alphanumeric characters, can start with a number, is case sensitive, and allows spaces but ignores leading spaces.

Step 7 Return to privileged EXEC mode:

```
Device(config)# exit
Device#
```

Step 8 Write the running configuration to the startup configuration file:

```
Device# copy running-config startup-config
```

The new password is now in the startup configuration.

Step 9 You must now reconfigure the device. If the system administrator has the backup device and VLAN configuration files available, you should use those.

Preventing Autonegotiation Mismatches

The IEEE 802.3ab autonegotiation protocol manages the device settings for speed (10 Mb/s, 100 Mb/s, and 1000 Mb/s, excluding SFP module ports) and duplex (half or full). There are situations when this protocol can incorrectly align these settings, reducing performance. A mismatch occurs under these circumstances:

- A manually set speed or duplex parameter is different from the manually set speed or duplex parameter on the connected port.

- A port is set to autonegotiate, and the connected port is set to full duplex with no autonegotiation.

To maximize the device performance and ensure a link, follow one of these guidelines when changing the settings for duplex and speed:

- Let both ports autonegotiate both speed and duplex.
- Manually set the speed and duplex parameters for the ports on both ends of the connection.

Note If a remote device does not autonegotiate, configure the duplex settings on the two ports to match. The speed parameter can adjust itself even if the connected port does not autonegotiate.

Troubleshooting SFP Module Security and Identification

Cisco small form-factor pluggable (SFP) modules have a serial EEPROM that contains the module serial number, the vendor name and ID, a unique security code, and cyclic redundancy check (CRC). When an SFP module is inserted in the device, the device software reads the EEPROM to verify the serial number, vendor name and vendor ID, and recompute the security code and CRC. If the serial number, the vendor name or vendor ID, the security code, or CRC is invalid, the software generates a security error message and places the interface in an error-disabled state.

Note The security error message references the GBIC_SECURITY facility. The device supports SFP modules and does not support GBIC modules. Although the error message text refers to GBIC interfaces and modules, the security messages actually refer to the SFP modules and module interfaces.

If you are using a non-Cisco SFP module, remove the SFP module from the device, and replace it with a Cisco module. After inserting a Cisco SFP module, use the **errdisable recovery cause gbic-invalid** global configuration command to verify the port status, and enter a time interval for recovering from the error-disabled state. After the elapsed interval, the device brings the interface out of the error-disabled state and retries the operation. For more information about the **errdisable recovery** command, see the command reference for this release.

If the module is identified as a Cisco SFP module, but the system is unable to read vendor-data information to verify its accuracy, an SFP module error message is generated. In this case, you should remove and reinsert the SFP module. If it continues to fail, the SFP module might be defective.

Executing Ping

If you attempt to ping a host in a different IP subnetwork, you must define a static route to the network or have IP routing configured to route between those subnets.

IP routing is disabled by default on all devices.

Note Though other protocol keywords are available with the **ping** command, they are not supported in this release.

Use this command to ping another device on the network from the device:

Command	Purpose
ping ip *host* | *address* `Device# ping 172.20.52.3`	Pings a remote host through IP or by supplying the hostname or network address.

Monitoring Temperature

The Device monitors the temperature conditions and uses the temperature information to control the fans.

Monitoring the Physical Path

You can monitor the physical path that a packet takes from a source device to a destination device by using one of these privileged EXEC commands:

Table 34: Monitoring the Physical Path

Command	Purpose
tracetroute mac [**interface** *interface-id*] {*source-mac-address*} [**interface** *interface-id*] {*destination-mac-address*} [**vlan** *vlan-id*] [**detail**]	Displays the Layer 2 path taken by the packets from the specified source MAC address to the specified destination MAC address.
tracetroute mac ip {*source-ip-address* | *source-hostname*} {*destination-ip-address* | *destination-hostname*} [**detail**]	Displays the Layer 2 path taken by the packets from the specified source IP address or hostname to the specified destination IP address or hostname.

Executing IP Traceroute

Note Though other protocol keywords are available with the **traceroute** privileged EXEC command, they are not supported in this release.

Command	Purpose
traceroute ip *host* `Device# traceroute ip 192.51.100.1`	Traces the path that packets take through the network.

Redirecting Debug and Error Message Output

By default, the network server sends the output from **debug** commands and system error messages to the console. If you use this default, you can use a virtual terminal connection to monitor debug output instead of connecting to the console port .

Possible destinations include the console, virtual terminals, internal buffer, and UNIX hosts running a syslog server. The syslog format is compatible with 4.3 Berkeley Standard Distribution (BSD) UNIX and its derivatives.

Note Be aware that the debugging destination you use affects system overhead. When you log messages to the console, very high overhead occurs. When you log messages to a virtual terminal, less overhead occurs. Logging messages to a syslog server produces even less, and logging to an internal buffer produces the least overhead of any method.

For more information about system message logging, see *Configuring System Message Logging*.

Using the show platform Command

The output from the **show platform** privileged EXEC command provides some useful information about the forwarding results if a packet entering an interface is sent through the system. Depending upon the parameters entered about the packet, the output provides lookup table results and port maps used to calculate forwarding destinations, bitmaps, and egress information.

Most of the information in the output from the command is useful mainly for technical support personnel, who have access to detailed information about the device application-specific integrated circuits (ASICs). However, packet forwarding information can also be helpful in troubleshooting.

Using the show debug command

The **show debug** command is entered in privileged EXEC mode. This command displays all debug options available on the switch.

To view all conditional debug options run the command **show debug condition** The commands can be listed by selecting either a condition identifier <*1-1000*> or *all* conditions.

To disable debugging, use the **no debug all** command.

Caution Because debugging output is assigned high priority in the CPU process, it can render the system unusable. For this reason, use **debug** commands only to troubleshoot specific problems or during troubleshooting sessions with Cisco technical support staff. Moreover, it is best to use **debug** commands during periods of lower network traffic and fewer users. Debugging during these periods decreases the likelihood that increased **debug** command processing overhead will affect system use.

Verifying Troubleshooting of the Software Configuration

Displaying OBFL Information

Table 35: Commands for Displaying OBFL Information - Cisco Catalyst 9600 Series Switches

Command	Purpose
show logging onboard RP active clilog [*continuous* | *detail* | *summary*] `Device# show logging onboard RP active clilog`	Displays the OBFL CLI commands that were entered on a module.

Command	Purpose
show logging onboard RP active environment [*continuous* | *detail* | *summary*] `Device# show logging onboard RP active environmentt`	Displays the UDI information for a module and for all the connected FRU devices: the PID, the VID, and the serial number.
show logging onboard RP active message [*continuous* | *detail* | *summary*] `Device# show logging onboard RP active message`	Displays the hardware-related messages generated by a module.
show logging onboard RP active counter [*continuous* | *detail* | *summary*] `Device# show logging onboard RP active counter`	Displays the counter information on a module.
show logging onboard RP active temperature [*continuous* | *detail* | *summary*] `Device# show logging onboard RP active temperature`	Displays the temperature information of a module.
show logging onboard RP active uptime [*continuous* | *detail* | *summary*] `Device# show logging onboard RP active uptime`	Displays the time when a module start, the reason the module restart, and the length of time that the module have been running since they last restarted.
show logging onboard RP active voltage [*continuous* | *detail* | *summary*] `Device# show logging onboard RP active voltage`	Displays the system voltages of a module.
show logging onboard RP active status [*continuous* | *detail* | *summary*] `Device# show logging onboard RP active status`	Displays the status of each OBFL application of a module.

Example: Verifying the Problem and Cause for High CPU Utilization

To determine if high CPU utilization is a problem, enter the **show processes cpu sorted** privileged EXEC command. Note the underlined information in the first line of the output example.

```
Device# show processes cpu sorted
CPU utilization for five seconds: 8%/0%; one minute: 7%; five minutes: 8%
PID Runtime(ms) Invoked uSecs 5Sec 1Min 5Min TTY Process
309 42289103 752750 56180 1.75% 1.20% 1.22% 0 RIP Timers
140 8820183 4942081 1784 0.63% 0.37% 0.30% 0 HRPC qos request
100 3427318 16150534 212 0.47% 0.14% 0.11% 0 HRPC pm-counters
192 3093252 14081112 219 0.31% 0.14% 0.11% 0 Spanning Tree
143 8 37 216 0.15% 0.01% 0.00% 0 Exec
...
```

```
<output truncated>
```

This example shows normal CPU utilization. The output shows that utilization for the last 5 seconds is 8%/0%, which has this meaning:

- The total CPU utilization is 8 percent, including both time running Cisco IOS processes and time spent handling interrupts.
- The time spent handling interrupts is zero percent.

Table 36: Troubleshooting CPU Utilization Problems

Type of Problem	Cause	Corrective Action
Interrupt percentage value is almost as high as total CPU utilization value.	The CPU is receiving too many packets from the network.	Determine the source of the network packet. Stop the flow, or change the switch configuration. See the section on "Analyzing Network Traffic."
Total CPU utilization is greater than 50% with minimal time spent on interrupts.	One or more Cisco IOS process is consuming too much CPU time. This is usually triggered by an event that activated the process.	Identify the unusual event, and troubleshoot the root cause. See the section on "Debugging Active Processes."

Configuration Examples for Troubleshooting Software

Example: Pinging an IP Host

This example shows how to ping an IP host:

```
Device# ping 172.20.52.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echoes to 172.20.52.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Device#
```

Table 37: Ping Output Display Characters

Character	Description
!	Each exclamation point means receipt of a reply.
.	Each period means the network server timed out while waiting for a reply.
U	A destination unreachable error PDU was received.
C	A congestion experienced packet was received.
I	User interrupted test.

Character	Description
?	Unknown packet type.
&	Packet lifetime exceeded.

To end a ping session, enter the escape sequence (**Ctrl-^ X** by default). Simultaneously press and release the **Ctrl**, **Shift**, and **6** keys and then press the **X** key.

Example: Performing a Traceroute to an IP Host

This example shows how to perform a **traceroute** to an IP host:

```
Device# traceroute ip 192.0.2.10

Type escape sequence to abort.
Tracing the route to 192.0.2.10

  1 192.0.2.1 0 msec 0 msec 4 msec
  2 192.0.2.203 12 msec 8 msec 0 msec
  3 192.0.2.100 4 msec 0 msec 0 msec
  4 192.0.2.10 0 msec 4 msec 0 msec
```

The display shows the hop count, the IP address of the router, and the round-trip time in milliseconds for each of the three probes that are sent.

Table 38: Traceroute Output Display Characters

Character	Description
*	The probe timed out.
?	Unknown packet type.
A	Administratively unreachable. Usually, this output means that an access list is blocking traffic.
H	Host unreachable.
N	Network unreachable.
P	Protocol unreachable.
Q	Source quench.
U	Port unreachable.

To end a trace in progress, enter the escape sequence (**Ctrl-^ X** by default). Simultaneously press and release the **Ctrl**, **Shift**, and **6** keys and then press the **X** key.

Additional References for Troubleshooting Software Configuration

Related Documents

Related Topic	Document Title
For complete syntax and usage information for the commands used in this chapter.	*Command Reference (Catalyst 9600 Series Switches)*

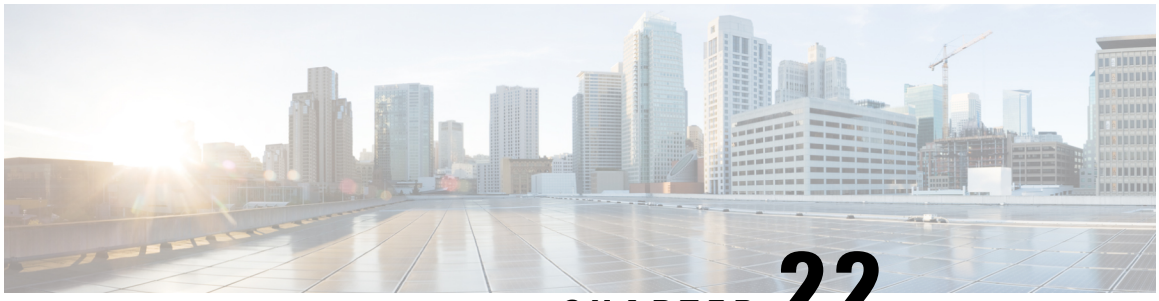
Feature History for Troubleshooting Software Configuration

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Gibraltar 16.11.1	Troubleshooting Software Configuration	Troubleshooting software configuration describes how to identify and resolve software problems related to the Cisco IOS software on the switch.
Cisco IOS XE Amsterdam 17.3.1	System-Report Files	The hostname is prepended to the system-report files. This makes the system-report files uniquely identifiable.
Cisco IOS XE Cupertino 17.7.1	Troubleshooting Software Configuration	Support for this feature was introduced on the Cisco Catalyst 9600 Series Supervisor 2 Module.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.



CHAPTER 22

Line Auto Consolidation

Line Auto Consolidation

Cisco IOS XE software runs a nonvolatile generation (NVGEN) process to retrieve the configuration state of the device. During the NVGEN process, the system auto consolidates the LINE commands based on common parameters.

When the device connects to Cisco Digital Network Architecture (DNA) Center or Cisco vManage and the center sends a line configuration through the Yet Another Next Generation (YANG) interface the resulting configuration is auto consolidated. This can cause a mismatch between the device and the DNA Center. The mismatch in configurations can lead to reverse sync from the device to the DNA Center. The device will be locked from any other configuration changes during this reverse sync. This can affect the performance of the device.

Starting with Cisco IOS XE 17.4.1 release, you can use the **no line auto-consolidation** command, in the global configuration mode, to disable the auto consolidation of LINE commands. Auto consolidation is enabled by default. To disable it use the no form of the command.

You can use the **show running-configuration all** command to display the configuration on the device. In the following example line auto-consolidation is enabled.

```
Device#sh running-config all | i auto-consolidation
line auto-consolidation
```

After auto consolidation is disabled the **show run** command output will be lengthy. This will impact the sizes of the running configuration and start-up configuration files. If you disable auto consolidation you will observe the following behaviors:

- Contiguous groups of lines that belong to the same configuration in a sub-mode will not be combined into a single range.

```
Device#show run | sec line
line con 0
stopbits 1
line vty 0 4
transport input ssh
line vty 5 9
transport input all
Device#configure terminal
Device(config)#no line auto-consolidation
```

```
Device(config)#line vty 10 15
Device(config-line)#transport input all
Device(config-line)#end
Device#show run | sec line
no line auto-consolidation
line con 0
stopbits 1
line vty 0 4
transport input ssh
line vty 5 9
transport input all
line vty 10 15
transport input all
```

- If you disable auto consolidation after configuring some lines with auto consolidation enabled, only the lines which were configured after auto consolidation was disabled will not be consolidated.

```
Device#show run | sec line
line con 0
stopbits 1
line vty 0 4
transport input ssh
line vty 5 9
transport input all
Device#configure terminal
Device(config)#line vty 10 15
Device(config-line)#transport input all
Device(config-line)#end
Device#show run | sec line
line con 0
stopbits 1
consolidated line vty 0 4
transport input ssh
line vty 5 15
transport input all
Device#configure terminal
Device(config)#no line auto-consolidation
Device(config)#line vty 16 20
Device(config-line)#transport input all
Device(config-line)#end
Device#show run | sec line
no line auto-consolidation
line con 0
stopbits 1
consolidated line vty 0 4
transport input ssh
line vty 5 15
transport input all
line vty 16 20
transport input all
```

- If you enable auto consolidation after it has been disabled, lines that were not consolidated will be auto consolidated.

```
Device#sh running-config | sec line
no line auto-consolidation
line con 0
exec-timeout 0 0
logging synchronous
stopbits 1
line vty 0 4
transport input ssh
line vty 5 15
transport input ssh
```

```
line vty 16 19
transport input ssh
Device#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)#line vty 20 25
Device(config-line)#transport input ssh
Device(config-line)#end
Device#sh running-config | sec line
no line auto-consolidation
line con 0
exec-timeout 0 0
logging synchronous
stopbits 1
line vty 0 4
transport input ssh
line vty 5 15
transport input ssh
line vty 16 19
transport input ssh
line vty 20 25
transport input ssh
Device#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)#line auto-consolidation
Device(config)#end
Device#show running-config | sec line
line con 0
exec-timeout 0 0
logging synchronous
stopbits 1
line vty 0 4
transport input ssh
line vty 5 25
transport input ssh
```

- You can configure lines with contiguous ranges. The configuration will be permitted.

```
Device#show run | sec line
no line auto-consolidation
line con 0
stopbits 1
line vty 0 4
transport input ssh
Device#configure terminal
Device(config)#line vty 5  20
Device(config)#transport input all
Device(config-line)#end
Device#show run | sec line
line con 0
stopbits 1
line vty 0 4
transport input ssh
line vty 5 20
transport input all
```

- You can't configure lines with non-contiguous ranges. The configuration is rejected.

```
Device#show run | sec line
no line auto-consolidation
line con 0
logging synchronous
line aux 0
line vty 0 4
transport input none
```

```
Device# configure terminal
Device(config)# line vty 10 20
% Bad line number - VTY line number is not contiguous.
```

- You can delete lines which are contiguous and at the end of the list. In the controller mode, you can delete one line at a time. You cannot delete lines in bulk. In autonomous mode, you can delete lines in bulk.

```
Device# show run | sec line
no line auto-consolidation
line con 0
stopbits 1
line vty 0 4
transport input ssh
line vty 5 20
transport input all
Device# configure terminal
Device(config)# no line vty 5 20
Device(config)# end
Device#show run | sec line
line con 0
stopbits 1
line vty 0 4
transport input ssh
```

- You can't delete lines which are not contiguous and at the end of the list. You can't delete a line that will result in a non-contiguous range when it is deleted. This will generate an error stating the line cannot be deleted.

```
Device# show run | sec line
no line auto-consolidation
line con 0
stopbits 1
line vty 0 4
transport input ssh
line vty 5 9
transport input none
line vty 10 20
transport input all
Device# configure terminal
Device(config)# no line vty 5 9
% Cannot delete the 9 line number as it is not the last VTY line number
```

- You can't delete lines that are in use or are default lines.

```
Device#show run | sec line
no line auto-consolidation
line con 0
stopbits 1
line vty 0 4
transport input ssh
line vty 5 20
transport input ssh
Device#configure terminal
Router(config)#no line vty 15
% Can't delete last 16 VTY lines, lines in use, statbit: 0x10C40, tiptop: 590
% process name: SSH Process
```

- You can modify subranges in autonomous mode. This will cause the lines to split which will cause a reverse sync of the configuration. You can’t modify subranges in the controller mode. This is a behavioural change between the controller and autonomous modes. In the controller mode, any modification of subranges is rejected to avoid discrepancy with the configuration pushed from a controller.

The following examples shows how you can modify subranges in autonomous mode.

```
Device#show run | sec line
no line auto-consolidation
line con 0
stopbits 1
line vty 0 4
transport input ssh
line vty 5 9
transport input none
Device#configure terminal
Device(config)#line vty 7 8
Device(config-line)#transport input telnet
Device(config-line)#end
Device#show run | sec line
line con 0
 stopbits 1
line vty 0 4
 transport input ssh
line vty 5 6
 transport input none
line vty 7 8
 transport input telnet
line vty 9
 transport input none
```

- The following example shows that modification of subranges is not supported in controller mode

```
Device#show run | sec line
no line auto-consolidation
line con 0
stopbits 1
line vty 0 4
transport input ssh
line vty 5 9
transport input none
Device#configure terminal
Device(config)# line vty 5 8
Device(config-line)# end
Uncommitted changes found, commit them? [yes/no/CANCEL] yes
Aborted: inconsistent value: Device refused one or more commands:
line vty 5 8
      ^
% Invalid input detected at '^' marker.
Component Response: "
% Modifications of overlapping/sub range is not allowed in controller mode"
Error executing command: CLI command error -
Device(config)# end
```

- You can modify overlapping ranges in autonomous mode. This will cause the lines to split which will cause a reverse sync of the configuration. You cannot modify overlapping ranges in the controller mode. In the controller mode, any modification of overlapping ranges is rejected to avoid discrepancy with the configuration pushed from a controller.

  The following example shows how you can modify overlapping ranges in autonomous mode.

```
Device#show run | sec line
no line auto-consolidation
line con 0
stopbits 1
line vty 0 4
transport input ssh
line vty 5 10
transport input none
```

```
line vty 11 20
transport input all
Device#configure terminal
Device(config)#line vty 8 12
Device(config-line)#transport input ssh
Device(config-line)#end
Device#show run | sec line
line con 0
stopbits 1
line vty 0 4
transport input ssh
line vty 5 7
transport input none
line vty 8 10
transport input ssh
line vty 11 12
transport input ssh
line vty 13 20
transport input all
```

- The following example shows that modification of overlapping ranges is not supported in controller mode.

```
Device#show run | sec line
no line auto-consolidation
line con 0
stopbits 1
line vty 0 4
transport input ssh
line vty 5 10
transport input none
line vty 11 20
transport input all
Device(config)# line vty 5 11
Device(config-line)# end
Uncommitted changes found, commit them? [yes/no/CANCEL] yes
Aborted: inconsistent value: Device refused one or more commands:
line vty 5 11
      ^
% Invalid input detected at '^' marker.
Component Response: "
% Modifications of overlapping/sub range is not allowed in controller mode"
Error executing command: CLI command error -
Device(config)# end
```

- You can replace a configuration from an auto consolidation enabled state to an auto consolidation disabled state.

```
Device#show run | sec line
line con 0
stopbits 1
line vty 0 4
transport input all
line vty 5 9
transport input ssh
line vty 10 15
transport input telnet
line vty 16 20
transport input ssh

Device#configure replace bootflash:cfg2.txt
This will apply all necessary additions and deletions
to replace the current running configuration with the
contents of the specified configuration file, which is
```

```
assumed to be a complete configuration, not a partial
configuration. Enter Y if you are sure you want to proceed. ? [no]: y
Total number of passes: 1
Rollback Done

Device#show run | sec line
no line auto-consolidation
line con 0
stopbits 1
line vty 0 4
transport input all
line vty 5 20
transport input ssh
```

- You can replace a configuration from an auto consolidation disabled state to an auto consolidation enabled state

```
Device#show run | sec line
no line auto-consolidation
line vty 0 4
transport input all
line vty 5 20
transport input ssh

Device#configure replace bootflash:cfg1.txt
This will apply all necessary additions and deletions
to replace the current running configuration with the
contents of the specified configuration file, which is
assumed to be a complete configuration, not a partial
configuration. Enter Y if you are sure you want to proceed. ? [no]: y
Total number of passes: 1
Rollback Done

Device#show run | sec line
line con 0
stopbits 1
line vty 0 4
transport input all
line vty 5 9
transport input ssh
line vty 10 15
transport input telnet
line vty 16 20
transport input ssh
```

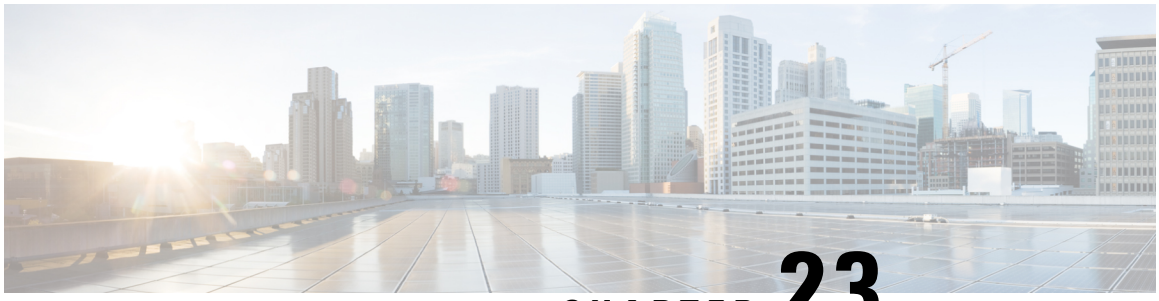
Feature History for Line Auto Consolidation

This table provides release and related information for features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Bengaluru 17.4.1	Line Auto Consolidation	Auto Consolidation of Line commands is enabled by default. The **no line auto-consolidation** command can be used to disable the auto consolidation of Line commands. The **line auto-consolidation** command was introduced.
Cisco IOS XE Cupertino 17.7.1	Line Auto Consolidation	Support for this feature was introduced on the Cisco Catalyst 9600 Series Supervisor 2 Module.

Use Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn.



CHAPTER 23

Troubleshooting System Management

Overview

This chapter provides links to documents authored by Cisco subject matter experts (SMEs). They aim to help you resolve technical issues without requiring a support ticket. If these documents are unable to resolve your issue, we recommend visiting the applicable Cisco Community. There is a wealth of information and advice available from fellow Cisco customers who may have experienced this issue already and provided a solution. If you are not able to find a resolution on the Community, it may be best that you raise a support ticket at Cisco Support. In cases where a support ticket has to be raised, these documents provide guidance about the data that should be collected and added to the support ticket. Specify the support document you referred, and TAC can create an improvement request with the document owner.

Support Articles

The documents in this section were created using specific software and hardware listed in the Components Used section of each article. However, this does not mean that they are limited to what is listed in Components Used, and generally remain relevant for later versions of software and hardware. Note that there could be some changes in the software or hardware that can cause commands to stop working, the syntax to change, or GUIs and CLIs to look different from one release to another.

The following are the support articles associated with this technology:

Document	Description
Cisco Smart Licensing - Troubleshooting Steps and Considerations on Catalyst platforms	This document describes how to work with Cisco Smart Licensing (cloud-based system) to manage software licenses on Catalyst switches.
Configure a Catalyst 9600 Switch	This document describes the initial configuration and verification procedure to set up the Catalyst 9600 switch.

Document	Description
Upgrade Guide for Cisco Catalyst 9000 Switches	This document describes upgrade methods for Catalyst9000 (Cat9K) switches.
Recommended Releases for Catalyst 9200/9300/9400/9500/9600 and Catalyst 3650/3850 Platforms	This document is to help customers find a stable software release for the enterprise switching platforms running Catalyst 9000 series switches.
Migrate Catalyst License to Smart Licensing Using Policy	This document describes what to expect after migration from an older license mechanism to the new "Smart Licensing Using Policy" mechanism in Cisco IOS XE 17.3.2 release and future releases.
Smart Licensing using Policy on Catalyst Switching Platforms	This document describes the Smart Licensing feature using Policy on Catalyst Switching Platforms and its various supported deployment mechanisms, from Cisco IOS XE 17.3.2 release and future releases.
Troubleshoot and Recover Catalyst 9000 Switches from Upgrade Failure Scenarios	This document describes the common failure scenarios that occur when Catalyst 9000 series devices are upgraded along with the procedure to recover them.
Configuration Register equivalent CLIs in IOS-XE	This document describes how to modify certain system parameters using CLI commands on Catalyst 9000 switches running Cisco IOS XE. These commands are an alternative to changing the configuration-register value on Cisco IOS.
Understand Hardware Resources on Catalyst 9000 Switches	This document describes how to understand and troubleshoot hardware resources on Catalyst 9000 series switches.
Understand IPv4 Hardware Resources on Catalyst 9000 Switches	This document describes how to understand and verify IPv4 Forwarding Information Base (FIB) hardware usage on Catalyst 9000 series switches.

Feedback Request

Your input helps. A key aspect to improving these support documents is customer feedback. Note that these documents are owned and maintained by multiple teams within Cisco. If you find an issue specific to the document (unclear, confusing, information missing, etc):

- Provide feedback using the **Feedback** button located at the right panel of the corresponding article. The document owner will be notified, and will either update the article, or flag it for removal.
- Include information regarding the section, area, or issue you had with the document and what could be improved. Provide as much detail as possible.

Disclaimer and Caution

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.