

Initial Configuration Tasks

- Configure SNMP MIB, on page 1
- Configure Smart Licensing, on page 16
- Set Backup Parameters, on page 18
- Configure NTP Servers, on page 20
- Configure Sub-interfaces, on page 24

Configure SNMP MIB

About SNMP MIB Support

The CCCSP supports SNMP integration with Cisco-USP-MIB. This MIB is SNMP version 2 (SNMPv2c) compliant.

SIP Proxy integrates an SNMP agent and SNMP MIBs to monitor the health and to conduct performance monitoring and data collection for SIP Proxy. Cisco-USP-MIB and Cisco-Process-MIB monitor the following data:

- Call Statistics
- Server Group Tables
- License State
- Memory and CPU Utilization
- System State

The SNMP integration sends notifications that helps to effectively monitor and manage performance and all the relevant system-specific data. Cisco-Process-MIB is supported in CCCSP for generating traps on configured CPU thresholds.

You can configure SNMP to send notifications to one or more monitoring systems. The maximum number of SNMP trap hosts that you can configure is limited to five.

Definitions

Table 1: Definition of SNMP MIB Related Terms

Term	Definition
Simple Network Management Protocol (SNMP)	It is a common network protocol that describes information passed between SNMP-enabled applications.
SNMP Agent	An SNMP Agent acts as a client to an SNMP management application by providing data values for registered OIDs.
Management Information Base (MIB)	MIBs are a defined hierarchy of data values managed by an SNMP Agent application.
SNMP Notification (Trap)/Informs	Information shared by a network entity with the management station to monitor a fault, exception, or an attribute value change. Traps do not need acknowledgment, but informs request acknowledgment. From SNMPv2, traps are known as notifications.
Object Identifiers (OID)	It is a unique string of digits representing the value defined in an MIB.
SNMP GET	SNMP GET is an SNMP message used to fetch the value for a particular OID.
SNMP SET	SNMP SET is an SNMP request used to modify information on the target agent (controlling agent behavior or configuration of agent).

Prerequisites

For using Cisco-USP-MIB, you must ensure that the following prerequisites are met:

- Configure Community Strings.
- Administrators of the SIP Proxy must be familiar with the Cisco Command-line Interface (CLI) or the Graphical User Interface (GUI).
- Use a MIB browser or Network Management System (NMS) to interact with the SIP Proxy.
- Upload the CISCO-USP-MIB to the NMS.
- Ensure that MIB browser or NMS provides SNMP v2c compliance.

Restrictions

SNMP MIB support is known to have the following limitations or restrictions:

- No Support for SNMP Version 3 (SNMPv3).

- Certain MIB objects in the SIP Proxy MIB tree are not supported. For a list of MIB objects that are not supported, see MIB Objects.
- If both read-only and read-write community strings are same for SNMP MIBs, then read-only takes preference and SET operations are not allowed.
- If the element table contains nested server group as an element, it does not display the partial state. The element state is shown as either up or down.
- The dropped call MIB objects are not updated if the license is in unidentified state.
- If call rate limit is set to a value lesser than license limit, cuspCallsDroppedExceedingLicense MIB counts calls are dropped due to call rate limit.
- There is no CLI and GUI equivalent for the data retrieved through MIB objects related to Calls Per Second (CPS).

Structure

The SNMP MIB structure for SIP Proxy has the following main considerations:

- The SIP Proxy is uniquely identified within the Cisco management (9) group by the number –.1.3.6.1.4.1.9.9.827.
- Use either of the following methods to identify objects in the CISCO-USP-MIB:
  - The object identifier –.1.3.6.1.4.1.9.9.827.<Cisco-USP- MIB-variable>
  - The object name – iso(1).org(3).dod(6).internet(1).private(4).enterprise(1).cisco(9).ciscoMgmt(9).CISCO-USP-MIB(827).<Cisco-USP-MIB-variable>
- The following traps in Cisco-Process-MIB for CPU utilization monitoring is supported:
  - cpmCPURisingThreshold (.1.3.6.1.4.1.9.9.109.2.0.1)
  - cpmCPUFallingThreshold (.1.3.6.1.4.1.9.9.109.2.0.2)

The SIP Proxy MIB structure has the following groups and subgroups:

- MIBNotifs
- MIBObjects
  - cuspScalar
  - cuspTable
  - cuspNotifControlInfo
- MIBConform

SIP Proxy SNMP MIBs

The CCCSP captures the following in a management information base:

- MIB Objects
- MIB Notifications (Traps)

MIB Objects

The supported SIP Proxy MIB Objects are:

- cuspScalar
  - cuspCallStats
  - cuspMessageStats
  - cuspThresholdValues
- cuspTable
- cuspNotifControlInfo

The CCCSP-15.0(1) release does not support the following MIB objects:

- cuspMemoryThresholdAlert
- cuspDiskSpaceThresholdAlert
- cuspBackupProcessFailAlert
- cuspConnectionExceptionAlert
- cuspThresholdValues
- cuspDiskSpaceThresholdValue
- cuspMemoryThresholdValue
- cuspMessageStats
- cuspStrayMessageCount
- cuspNoOfMessagesRecieved
- cuspMemoryThresholdAlertEnable
- cuspExtensiveLoggingAlertEnable
- cuspDiskSpaceThresholdAlertEnable
- cuspBackupProcessFailAlertEnable
- cuspConnectionExceptionAlertEnable
- cuspDiskSpaceUsed

cuspScalar

This table contains a list of SIP Proxy scalars. An entry in this table represents SIP Proxy information relevant to licenses, system state, and memory.

Table 2: MIB Description for cuspScalar

MIB	OID	Description
cuspLastCounterResetTime	.1.3.6.1.4.1.9.9.827.1.1.1	Gives the timestamps in date and time when the call counter was last reset. All counters related to calls, Calls Per Second (CPS) and messages are reset when the counter is reset.
cuspSystemState	.1.3.6.1.4.1.9.9.827.1.1.2	Gives the SIP Proxy system state as UP or DOWN.
cuspSystemUpTime	.1.3.6.1.4.1.9.9.827.1.1.3	Gives information on the active time of the SIP Proxy system.
cuspLicenseLimit	.1.3.6.1.4.1.9.9.827.1.1.4	Gives the license limit information. Calls are rejected if the license limit is exceeded.
cuspLicenseState	.1.3.6.1.4.1.9.9.827.1.1.5	Gives the current license state of SIP Proxy.
cuspSmartAgentState	.1.3.6.1.4.1.9.9.827.1.1.6	Gives the current license state of the SmartLicense Agent.
cuspConfiguredMemory	.1.3.6.1.4.1.9.9.827.1.1.7	Gives the total memory (RAM) configured on SIP Proxy in Megabytes.
cuspMemoryUsed	.1.3.6.1.4.1.9.9.827.1.1.8	Gives the SIP Proxy current memory (RAM) usage information in Megabytes.
cuspDiskSpaceUsed	.1.3.6.1.4.1.9.9.827.1.1.9	Gives the current disk utilization of CUSP in MB (Megabytes).

cuspCallStats

This SIP Proxy MIB defines data related to calls.

Table 3: MIB Description for cuspCallStats

MIB	OID	Description
cuspTotalCalls	.1.3.6.1.4.1.9.9.827.1.1.10.1	The total number of calls since the last counter reset.
cuspTotalFailedCalls	.1.3.6.1.4.1.9.9.827.1.1.10.2	The total number of failed calls since last counter reset.
cuspCPS	.1.3.6.1.4.1.9.9.827.1.1.10.3	The current running Calls Per Second (CPS) information.

MIB	OID	Description
cuspAvgCPSOneMin	.1.3.6.1.4.1.9.9.827.1.1.10.4	The average CPS in the last one minute.
cuspMaxCPSOneMin	.1.3.6.1.4.1.9.9.827.1.1.10.5	The Maximum value of CPS in the last one minute.
cuspDroppedCallsOneSec	.1.3.6.1.4.1.9.9.827.1.1.10.6	The count on number of calls dropped in the last one second.
cuspAvgDroppedCallsOneMin	.1.3.6.1.4.1.9.9.827.1.1.10.7	The average of dropped calls per second' in the last one minute.
cuspMaxDroppedCallsOneMin	.1.3.6.1.4.1.9.9.827.1.1.10.8	The Maximum of dropped calls per second' in the last one minute.
cuspCallsRoutedOneSec	.1.3.6.1.4.1.9.9.827.1.1.10.9	The number of calls routed through CUSP in one second.
cuspAvgCallsRoutedOneMin	.1.3.6.1.4.1.9.9.827.1.1.10.10	The average of calls routed per second' in last one minute.
cuspMaxCallsRoutedOneMin	.1.3.6.1.4.1.9.9.827.1.1.10.11	The maximum of 'calls routed per second' in the last one minute.
cuspCallsDroppedExceedingLicense	.1.3.6.1.4.1.9.9.827.1.1.10.12	The total calls dropped due to exceeding license limit.

cuspThresholdValues

The SIP proxy MIB object cuspThresholdValues (.1.3.6.1.4.1.9.9.827.1.1.12) provides threshold value information (as configured by user) on disk space and memory utilization.

Table 4: MIB Description for cuspThresholdValues

MIB	OID	Description
cuspDiskSpaceThresholdValue	.1.3.6.1.4.1.9.9.827.1.1.12.1	The percentage threshold value configured by the user. If the percentage disk space utilization exceeds this limit, then cuspDiskSpaceThresholdAlert notification is sent.
cuspMemoryThresholdValue	.1.3.6.1.4.1.9.9.827.1.1.12.2	The percentage threshold value configured by the user. If the percentage memory utilization exceeds this limit, then cuspMemoryThresholdAlert notification is sent.

cuspTable

The SIP proxy MIB object cuspTable (.1.3.6.1.4.1.9.9.827.1.2) consists of two main subgroups of objects:

- cuspServerGroupTable (OID:.1.3.6.1.4.1.9.9.827.1.2.1)
- cuspElementTable (OID:.1.3.6.1.4.1.9.9.827.1.2.2)



Note If data is retrieved from multiple network elements using cuspTable MIBs, the CPU utilization can spike beyond the optimum levels.

cuspServerGroupTable

The MIB cuspServerGroupTable represents a list of server groups that are part of active configuration. Server groups define the elements with which the SIP Proxy system interacts for each network.

Table 5: MIB Description for cuspServerGroupTable

MIB	OID	Description
cuspServerGroupEntry	.1.3.6.1.4.1.9.9.827.1.2.1.1	An entry (conceptual row) in the ServerGroup Table.
cuspServerGroupIndex	.1.3.6.1.4.1.9.9.827.1.2.1.1.1	A unique value, greater than zero, for each server group.
cuspServerGroupName	.1.3.6.1.4.1.9.9.827.1.2.1.1.2	The name of the server group.
cuspServerGroupNetwork	.1.3.6.1.4.1.9.9.827.1.2.1.1.3	The network to which the server group belongs.
cuspServerGroupStatus	.1.3.6.1.4.1.9.9.827.1.2.1.1.4	The Server group status is given as up, partial down, and down.
cuspServerGroupPingStatus	.1.3.6.1.4.1.9.9.827.1.2.1.1.5	Server group ping status.
cuspServerGroupLBType	.1.3.6.1.4.1.9.9.827.1.2.1.1.6	The load balancing algorithm for the server group.



Note The CuspservergroupPingStatus MIB object retrieves the information of a group, irrespective of the global ping status.

cuspElementTable

The MIB cuspElementTable provides a list of elements in a server group table. Also, the table contains information on status (up or down) of the element, its Q-value, weight, and transport type.

Table 6: MIB Description for cuspElementTable

MIB	OID	Description
cuspElementEntry	.1.3.6.1.4.1.9.9.827.1.2.2.1	An entry (conceptual row) in the cuspElementTable.
cuspElementIndex	.1.3.6.1.4.1.9.9.827.1.2.2.1.1	A unique value, greater than zero, for each element.
cuspElementName	.1.3.6.1.4.1.9.9.827.1.2.2.1.2	The Server group element ID.
cuspElementStatus	.1.3.6.1.4.1.9.9.827.1.2.2.1.3	The server group element status as up or down.
cuspElementQValue	.1.3.6.1.4.1.9.9.827.1.2.2.1.4	The Q value of the server group element. Q value range is 0.0 to 1.0.
cuspElementWeight	.1.3.6.1.4.1.9.9.827.1.2.2.1.5	The weight of the server group element. Weight is used for load balancing between server group elements.
cuspElementPort	.1.3.6.1.4.1.9.9.827.1.2.2.1.6	Gives the port number of the server group element.
cuspElementTransport	.1.3.6.1.4.1.9.9.827.1.2.2.1.7	The transport type of the server group element. Transport type can be udp, tcp, or tls.
cuspElementTotalCalls	.1.3.6.1.4.1.9.9.827.1.2.2.1.8	The total routed calls to the server group element.
cuspElementFailedCalls	.1.3.6.1.4.1.9.9.827.1.2.2.1.9	The total failed calls on the server group element.

cuspNotifControlInfo

The MIB cuspNotifControlInfo (OID is.1.3.6.1.4.1.9.9.827.1.3) contains object that manages (enabling and disabling) the traps defined in CiscoUspMIBNotifs.

Table 7: MIB Description for cuspNotifControlInfo

MIB	OID	Description
cuspNotifSeverity	.1.3.6.1.4.1.9.9.827.1.3.1	The classification on the event severity.
cuspNotifDetail	.1.3.6.1.4.1.9.9.827.1.3.2	The detailed information on error encountered.
cuspSystemStateAlertEnable	.1.3.6.1.4.1.9.9.827.1.3.3	Controls generation of cuspSystemStateAlert, cuspConnectionExceptionAlert.

MIB	OID	Description
cuspServerGroupAlertEnable	.1.3.6.1.4.1.9.9.827.1.3.4	Controls the generation of cuspServerGroupElementAlert and cuspServerGroupAlert.
cuspServerGroupElementAlertEnable	.1.3.6.1.4.1.9.9.827.1.3.5	Controls the generation of cuspServerGroupElementAlert.
cuspLicenseExceededAlertEnable	.1.3.6.1.4.1.9.9.827.1.3.6	Controls the generation of cuspLicenseExceededAlert.
cuspLicenseStateAlertEnable	.1.3.6.1.4.1.9.9.827.1.3.7	Controls the generation of cuspLicenseStateAlert.
cuspExtensiveLoggingAlertEnable	.1.3.6.1.4.1.9.9.827.1.3.8	Controls the generation of cuspExtensiveLoggingAlert.
cuspDiskSpaceThresholdAlertEnable	.1.3.6.1.4.1.9.9.827.1.3.9	Controls the generation of cuspDiskSpaceThresholdAlert.
cuspMemoryThresholdAlertEnable	.1.3.6.1.4.1.9.9.827.1.3.10	Controls the generation of cuspMemoryThresholdAlert.
cuspBackupProcessFailAlertEnable	.1.3.6.1.4.1.9.9.827.1.3.11	Controls the generation of cuspBackupProcessFailAlert notification.
cuspConnectionExceptionAlertEnable	.1.3.6.1.4.1.9.9.827.1.3.12	Controls the generation of cuspConnectionExceptionAlert.
cuspSIPMessageQueueOverflowAlertEnable	.1.3.6.1.4.1.9.9.827.1.3.13	Controls the generation of cuspSIPMessageQueueOverflowAlert.

MIB Notifications (Traps)

SIP Proxy generates trap notifications when the Network Management Station (NMS) or the administrator has to be informed about an event. The notification describes the operation state information of a service when a condition occurs. Traps provide information on issues that occur in the network element without polling for SNMP objects.

The administrator can control traps using the Command-line Interface (CLI), the Graphical User Interface (GUI), or through SNMP. By default, the traps are set to disabled state.

CCCSP-15.0(1) supports a generic trap and raises SNMP traps on the following events:

- License Limit is exceeded
- System Failure
- Change in Server element state
- Change in Server group element state

CCCSP-15.0(1) does not support SNMP traps on the following events:

- Backup Process Failure
- Memory threshold is exceeded
- Disk space threshold is exceeded
- Extensive Debug level logging
- Connection Exception

Table 8: MIB Description for MIB Traps

MIB	OID	Description
cuspSystemStateAlert	.1.3.6.1.4.1.9.9.827.0.1	Generated when the SIP Proxy system goes up or down. This notification can be enabled or disabled by setting cuspSystemStateAlertEnable. CLI command to configure the trap: **snmp-server enable traps System-State**
cuspServerGroupElementAlert	.1.3.6.1.4.1.9.9.827.0.2	Generated when the status of server group element changes. This notification can be enabled or disabled by setting cuspServerGroupAlertEnable. CLI command to configure the trap: **snmp-server enable traps SG-Element**
cuspServerGroupAlert	.1.3.6.1.4.1.9.9.827.0.3	Generated when all the elements in the server group go down. Also, it is generated when any one element in the server group comes up after all the elements in the group were down. This notification is enabled or disabled by setting cuspServerGroupAlertEnable. CLI command to configure the trap: **snmp-server enable traps Server-Group**
cuspMemoryThresholdAlert	.1.3.6.1.4.1.9.9.827.0.4	Generated when SIP Proxy memory usage exceeds the cuspMemoryThresholdValue. This notification can be enabled or disabled by setting cuspThresholdAlertEnable.

MIB	OID	Description
cuspLicenseExceededAlert	.1.3.6.1.4.1.9.9.827.0.5	Generated when average CPS exceeds cuspLicenseLimit. This notification can be enabled or disabled by setting cuspLicenseExceededAlertEnable. CLI command to configure the trap: **snmp-server enable traps License-Exceeded**
cuspLicenseStateAlert	.1.3.6.1.4.1.9.9.827.0.6	Generated when SIP Proxy license state changes. This notification is enabled or disabled by setting cuspLicenseStateAlertEnable CLI command to configure the trap: **snmp-server enable traps License-State**
cuspExtensiveLoggingAlert	.1.3.6.1.4.1.9.9.827.0.7	Generated when extensive debug level logging is enabled in SIP Proxy. Extensive logging has an impact on performance and system stability. This notification can be enabled or disabled by setting cuspExtensiveLoggingAlertEnable. CLI Command to configure the trap: **snmp-server enable traps Extensive-Logging**
cuspDiskSpaceThresholdAlert	.1.3.6.1.4.1.9.9.827.0.8	Generated when the SIP Proxy Disk usage exceeds the cuspDiskSpaceThresholdValue. This notification can be enabled or disabled by setting cuspThresholdAlertEnable.
cuspBackupProcessFailAlert	.1.3.6.1.4.1.9.9.827.0.9	Generated when backup process fails. This notification is enabled or disabled by setting cuspBackupProcessFailAlertEnabl.
cuspConnectionExceptionAlert	.1.3.6.1.4.1.9.9.827.0.10	Generated when a connection exception occurs. This notification can be enabled or disabled by setting cuspSystemStateAlertEnable.

MIB	OID	Description
cuspSIPMessageQueueOverflowAlert	.1.3.6.1.4.1.9.9.827.0.11	Generated when the SIP Proxy system queue is full. Queue full indicates that either SIP Proxy is overloaded or encountering network issues. The time interval between two successive notifications is 5 minutes. Notification is not sent within this time frame even if the queue is full. This back-off timer of 5 minutes prevents the SIP Proxy overload. This notification can be enabled or disabled by setting cuspSIPMessageQueueOverflowAlertEnable. CLI command to configure the trap: **snmp-server enable traps SIP-Message-Queue-Overflow**
cpmCPURisingThreshold	.1.3.6.1.4.1.9.9.109.2.0.1	Sent when configured rising CPU utilization threshold is reached and CPU utilization remains above the threshold for configured interval, and such a notification is requested. CLI command to configure the trap: **snmp-server enable traps CPU-Rising**
cpmCPUFallingThreshold	.1.3.6.1.4.1.9.9.109.2.0.2	Sent when the configured falling threshold is reached and CPU utilization remains under threshold for configured interval, and such a notification is requested. CLI command to configure the trap: **snmp-server enable traps CPU-Falling**

Note cuspLicenseExceededAlert is not generated if the license is in unidentified state.

Configure Community String

Configure community string to poll data using MIB objects.

SUMMARY STEPS

1. **config**
2. **snmp-server community** *community string {RO| RW}*
3. **end**
4. **write memory**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**config** **Example:** `Hostname# config terminal`	Enables privileged EXEC mode.
Step 2	**snmp-server community** *community string {RO| RW}* **Example:** `Hostname(config)# snmp-server community public RW`	Configures the community string. The access could be read-only or read-write based on the selected configuration.
Step 3	**end** **Example:** `Hostname(config)# end`	Exits the privileged EXEC mode.
Step 4	**write memory** **Example:** `Hostname# write memory`	Stores the configuration in the startup configuration file.

Examples

Example

The following example configures Community Strings on the CCCSP:

```
Hostname# config terminal
Hostname(config)# snmp-server community public RW
Hostname(config)# end
Hostname# write memory
```

Configure SNMP Traps

SUMMARY STEPS

1. **config** *terminal*

2. **snmp-server host** *IP Address community string*
3. **snmp-server enable traps [All | System-State | Server-Group | SG-Element | CPU-Rising | CPU-Falling | License-State | License-Exceeded | Extensive-Logging | SIP-Message-Queue-Overflow]**
4. **snmp-server enable traps**
5. **end**
6. **write memory**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**config** *terminal* **Example:** `Hostname# config terminal`	Enables privileged EXEC mode.
Step 2	**snmp-server host** *IP Address community string* **Example:** `Hostname(config)# snmp-server host 10.104.54.108 public`	Specifies the host that receives SNMP notifications.
Step 3	**snmp-server enable traps [All | System-State | Server-Group | SG-Element | CPU-Rising | CPU-Falling | License-State | License-Exceeded | Extensive-Logging | SIP-Message-Queue-Overflow]** **Example:** `Hostname(config)# snmp-server enable traps SG-Element`	Activates the traps selected. The command **snmp-server enable traps all** activates all traps. To activate a specific trap, follow **snmp-server enable traps** with the subcommand specific to that trap.
Step 4	**snmp-server enable traps** **Example:** `Hostname(config)# snmp-server enable traps`	Enables trap generation from CCCSP to the configured hosts. Traps are sent to the host only when this global command is enabled.
Step 5	**end** **Example:** `Hostname(config)# end`	Exits the privileged EXEC mode.
Step 6	**write memory** **Example:** `Hostname(config)# write memory`	Stores the configuration in the startup configuration file.

Example

The following example configures SNMP Traps on the CCCSP:

```
Hostname# config terminal
 Hostname(config)# snmp-server host 10.104.54.108 public
 Hostname(config)# snmp-server enable traps SG-Element
 Hostname(config)# snmp-server enable traps
 Hostname(config)# end
Hostname# write memory
```

Note The trap body information can only be seen on the trap listener host. However, a generic trap notification will be logged in the vCCCSP atrace.log logs.

Configure CPU Threshold Values for Traps

To define rising and falling CPU threshold values for traps, perform these steps:

SUMMARY STEPS

1. **config** *terminal*
2. **process cpu threshold type {total} rising** *percentage* **interval** *seconds* **falling** *percentage* **interval** *seconds*

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**config** *terminal* **Example:** `Hostname# config terminal`	Enables privileged EXEC mode.
Step 2	**process cpu threshold type {total} rising** *percentage* **interval** *seconds* **falling** *percentage* **interval** *seconds* **Example:** `Hostname(config)# process cpu threshold type {total} rising 80 interval 300 falling 5 interval 300`	Sets the CPU threshold notifications types and values. In this example, the CPU utilization threshold is set to 80 percent for a rising threshold notification and 5 percent for a falling threshold notification. The polling interval is set as 300 seconds.

Examples

Example

The following example configures CPU thresholding values for SNMP traps on the CCCSP:

```
Hostname# config terminal
Hostname(config)# process cpu threshold type {total} rising 80 interval 300 falling 5
interval 300
```

Configure Smart Licensing

About Smart Licensing

Cisco Smart Software Licensing is a standardized licensing platform that facilitates you to deploy and manage Cisco software licenses easily and quickly. Cisco Smart Software Licensing establishes a pool of software licenses that can be used across your network in a flexible and automated manner. It also provides visibility to your purchased and deployed licenses in your network. Cisco Smart Software Licensing removes the need for Product Activation Keys (PAKs) and reduces your license activation and registration time.



Note For more information on Smart Licensing, see http://www.cisco.com/go/smartlicensing.

Before you begin

Log in to your Smart Account and generate a valid token and make sure that the required licenses for CCCSP is available in your Account.

SUMMARY STEPS

1. **enable**
2. **license smart activate cccsp** *count*
3. **license smart destinationAddr** *server url*
4. **license smart httpProxyAddr** *ip address / FQDN* **port** *port*
5. **license smart register token_id** *token*

DETAILED STEPS

Procedure

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

	Command or Action	Purpose
Step 2	**license smart activate cccsp** *count* **Example:** `Hostname# license smart activate cccsp 100`	Activates the request number of licenses. The count must be a multiple of 5. In the example, the count is configured as 100, so the actual license count that is consumed on the Smart Account is 20 (100/5).
Step 3	**license smart destinationAddr** *server url* **Example:** `Hostname# license smart destinationAddr https://smartreceiver.cisco.com/licservice/license`	Connects to the central licensing server.

Step 4	**license smart httpProxyAddr** *ip address / FQDN* **port** *port*
	Example:
	`Hostname# license smart httpProxyAddr http://10.1.1.1 port 3128`
	Example:
	`Hostname# license smart httpProxyAddr https://10.1.1.1 port 3128`
		Sets the HTTP(S) proxy server IP address or FQDN and port address for smart licensing.

Step 5	**license smart register token_id** *token* **Example:** `Hostname# license smart register token_id` [redacted]	Registers the device instance with the Cisco licensing cloud. This step is performed only once per device instance. The license agent registers the product with Cisco and receives back an identity certificate. This certificate is saved and automatically used for all future communications with Cisco. The license agent automatically renews the registration information with Cisco after one year.

Example

The following example configures Smart License on the CCCSP:

```
Hostname# enable
Hostname# license smart activate cccsp 100
Hostname# license smart destinationAddr https://smartreceiver.cisco.com/licservice/license
Hostname# license smart httpProxyAddr https://10.1.1.1 port 3128
Hostname# license smart register token_id
```
[redacted]

Verify Smart Licensing

Use the **show license smart summary** command to verify if the Smart License Agent is successfully configured.

```
Hostname# show license smart summary
Smart Agent is Enabled: true
Current State of the Agent: OUT_OF_COMPLIANCE
Evaluation Mode: NO
Registration Successful: YES
Authorization Successful: YES

CPS Count Requested: 5
Configured destination address: https://smartreceiver.cisco.com/licservice/license
HTTP Proxy Address: http://10.10.10.39:3128
License UDI: CCCSP:JZp7KbrPwS6
Product Serial Number: JZp7KbrPwS6
Product ID: CCCSP
Product License Version: 1.0
Licensing State: OutOfCompliance
Registration expiry period: Tue Aug 11 17:57:33 IST 2026
Latest Failure Reason String Notification: Successful
Auth period: Tue Aug 12 18:02:37 IST 2025
```

Set Backup Parameters

About Backup Parameters

CCCSP backup and restore functions use an SFTP server to store and retrieve data. The backup function copies the files from CCCSP to the SFTP server and the restore function copies the files from the SFTP server to CCCSP. The SFTP server can reside anywhere in the network as long as the backup and restore functions can access it with an IP address or hostname.

All CCCSP backup files are stored on the specified server. You can copy the backup files to other locations or servers, if necessary.

The backup parameters specify the SFTP server to use for storing CCCSP backup files and the number of backups that are stored before the system overwrites the oldest one.

Before you begin

- Verify that an SFTP administrator or other user who can log in to the SFTP server has full permission on the SFTP server, such as read, write, overwrite, create, and delete permissions for files and directories.
- Gather the SFTP server URL and the username and password of the SFTP server login.
- Make sure that the SFTP server URL is pointing to the absolute path of the backup directory. For example, sftp://<hostname>/full/path/from/root/to/backup_directory.
- Determine the number of revisions to save before the oldest backup is overwritten.

SUMMARY STEPS

1. **configure terminal**
2. **backup server url** *sftp-url* { **username** *sftp-username* **password** *sftp-password* }
3. **backup revisions** *number*
4. **end**
5. **show backup**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Hostname# config terminal`	Enters configuration mode.

Step 2	**backup server url** *sftp-url* { **username** *sftp-username* **password** *sftp-password* }
	Example:
	`Hostname(config)# backup server url sftp://main/backups username "admin" password "wxyz"`
	`Hostname(config)# backup server url sftp://192.0.2.15/backups username "admin" password "wxyz"`
		Sets the backup parameters.
		Note
		You must configure the backup server before you can configure the backup revisions.
		• **server url**—The *sftp-url* value is the URL to the network SFTP server where the backup files will be stored.
		• The *sftp-username* and *sftp-password* values are the username and password for the network SFTP server.
		In the example, **main** is the hostname of the SFTP server and **backup** is the directory where backup files are stored.
		Note
		/backups is an absolute path to the backups directory.

Step 3	**backup revisions** *number* **Example:** `Hostname(config)# backup revisions 5`	Sets the number of backup files that will be stored. When the system reaches this number of backups, it deletes the oldest stored file.
Step 4	**end** **Example:** `Hostname(config)# end`	Exists configuration mode.
Step 5	**show backup** **Example:** `Hostname# show backup`	Displays the backup server configuration information, including the SFTP server URL and the maximum number of backup files available.

Example

The following example configures a backup server and displays the **show backup** output:

```
Hostname# enable
```

```
Hostname# configure terminal
Hostname(config)# backup revisions 5
Hostname(config)# backup server url sftp://10.12.0.1/sftp username “admin” password “wxyz”
Hostname(config)# end
Hostname# show backup
Server URL: sftp://10.12.0.1/sftp
User Account on Server:

Number of Backups to Retain: 5
Hostname#
```

What to do next

Related Topics

- For information about the CLI commands, see the CLI Command Reference for .
- For information about back up and restore configuration, see Backing Up and Restoring Data .

Configure NTP Servers

When you install the Cisco Contact Center SIP Proxy (CCCSP) application, the system gives you the option of adding up to five Network Time Protocol (NTP) servers. You can add, modify, or delete up to five NTP servers in CCCSP application using the CLI.

Add NTP Servers

You can specify an NTP server using its IP address or its hostname.

CCCSP uses the DNS server to resolve the hostname to an IP address and stores the IP address as an NTP server. If DNS resolves the hostname to more than one IP address, CCCSP randomly chooses one of the IP addresses that is not already designated as an NTP server. If you do not want to go with the random choice, set the **prefer** attribute for one server.

To configure an NTP server with multiple IP addresses for a hostname, repeat the configuration steps using the same hostname. Each iteration assigns the NTP server to its remaining IP addresses.

SUMMARY STEPS

1. **configure terminal**
2. **ntp server** { *hostname* | *ip-address* } [**prefer**]
3. **end**
4. **show ntp status**
5. **show ntp configuration**
6. **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Hostname# config terminal`	Enters configuration mode.

Step 2	**ntp server** { *hostname* | *ip-address* } [**prefer**]
	Example:
	`Hostname(config)# ntp server 192.0.2.14`
	`Hostname(config)# ntp server 192.0.2.17 prefer`
		Specifies the hostname or IP address of the NTP server.
		If more than one server is configured, the server with the **prefer** attribute is used before the others.

Step 3	**end** **Example:** `Hostname(config)# end`	Exists configuration mode.
Step 4	**show ntp status** **Example:** `Hostname# show ntp status`	Displays statistics about the NTP server.
Step 5	**show ntp configuration** **Example:** `Hostname# show ntp configuration`	Displays the configured NTP servers.
Step 6	**copy running-config startup-config** **Example:** `Hostname# copy running-config startup-config`	Copies the configuration changes to the startup configuration.

Example

The following commands configure the NTP server:

```
Hostname# configure terminal
Hostname(config)# ntp server 192.0.2.14
Hostname(config)# exit
Hostname#
```

The output from the **show ntp status** command looks similar to the following:

```
Hostname# show ntp status
```

```
NTP reference server 1: 192.0.2.14
Status: sys.peer
Time difference (secs): 3.268110099434328E8
Time jitter (secs): 0.1719226837158203
```

Remove NTP Server

You can remove a NTP server using its IP address or hostname.

SUMMARY STEPS

1. **configure terminal**
2. **no ntp server** { *hostname* | *ip-address*
3. **exit**
4. **show ntp status**
5. **show ntp configuration**
6. **copy running-config startup-config**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Hostname# configure terminal`	Enters the configuration mode.

Step 2	**no ntp server** { *hostname* | *ip-address*
	Example:
	`Hostname(config)# no ntp server 192.0.2.14`
	`Hostname(config)# no ntp server myhost`
		Specifies the hostname or IP address of the NTP server to remove.

Step 3	**exit** **Example:** `Hostname(config)# exit`	Exits configuration mode.
Step 4	**show ntp status** **Example:** `Hostname# show ntp status`	Displays statistics about the NTP server.
Step 5	**show ntp configuration** **Example:**	Displays the configured NTP servers.

	Command or Action	Purpose
	`Hostname# show ntp configuration`	
Step 6	**copy running-config startup-config** **Example:** `Hostname# copy running-config startup-config`	Copies the configuration changes to the startup configuration.

Display NTP Server Information

Commands to Display NTP Server Information

The following commands are available to display NTP server configuration information and status:

- **show ntp associations**
- **show ntp servers**
- **show ntp source**
- **show ntp status**

Examples of Showing NTP Server Information

The following is a sample output for the **show ntp associations** command:

```
Hostname# show ntp associations

MS Name/IP address  Stratum Poll Reach LastRx  Last sample
===============================================================================
^* 10.10.10.39         3     6    377   25     -55us[ -294us] +/-  103ms
```

The following is a sample output for the **show ntp servers** command:

```
Hostname# show ntp servers

Name/IP Address    NP  NR  Span  Frequency  Freq Skew  Offset  Std Dev
===============================================================================
nemv6.cisco.com     6   3   324     -1.449     5.560    -56us   178us
```

The following is a sample output for the **show ntp source** command:

```
Hostname# show ntp source
MS  Name/IP address  Stratum Poll Reach LastRx Last sample
===============================================================================
^*  nemv6.cisco.com   3       6   377     1    +12us[ +174us] +/-  103ms
```

The following is a sample output for the **show ntp status** command:

```
Hostname# show ntp status
```

```
Remote address  : 10.10.10.39 (0A405627)
Remote port     : 123
Local address   : 10.10.10.199 (0A4E68C7)
Leap status     : Normal
Version         : 4
Mode            : Server
Stratum         : 3
Poll interval   : 6 (64 seconds)
Precision       : -24 (0.000000060 seconds)
Root delay      : 0.189423 seconds
Root dispersion : 0.007996 seconds
Reference ID    : 417A8361 ()
Reference time  : Tue Aug 12 06:03:58 2025
Offset          : +0.000048062 seconds
Peer delay      : 0.000625227 seconds
Peer dispersion : 0.000000083 seconds
Response time   : 0.000123992 seconds
Jitter asymmetry: +0.50
NTP tests       : 111 111 1111
Interleaved     : No
Authenticated   : No
TX timestamping : Daemon
RX timestamping : Kernel
Total TX        : 211
Total RX        : 211
Total valid RX  : 211
Total good RX   : 203
```

Related Topics

- For information about the CLI commands, see the *CLI Command Reference Guide*.
- For information about the initial installation of the CCCSP system and the post installation configuration tool, see the *Installation Guide for CCCSP*. For information about copying the configuration, see Copying Configurations.

Configure Sub-interfaces

You can define multiple sub-interfaces on Cisco Contact Center SIP Proxy (CCCSP) and there is no specific restriction on the number of sub-interfaces from CCCSP.

Note Ensure that all the sub-interfaces are configured with IP addresses from the same subnet, as the trunk port config with sub-interfaces are not supported CCCSP-15.0(1).

SUMMARY STEPS

1. **configure terminal**
2. **interface FastEthernet**
3. **ip address**
4. **end**

DETAILED STEPS

Procedure

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `Hostname# conf t`	Enters configuration mode.
Step 2	**interface FastEthernet** **Example:** `Hostname(config)# interface FastEthernet 0.11`	Enters interface FastEthernet.
Step 3	**ip address** **Example:** `Hostname(config-subif)# ip address 10.10.10.19 255.255.255.0`	Configures subinterface for Fastethernet 0.11 in configuration mode.
Step 4	**end** **Example:** `Hostname(config-subif)# end`	Exits network command mode.

Examples

Example

The following example configures subinterface for Fastethernet:

```
Hostname# config
Hostname(config)# interface FastEthernet 0.11
Hostname(config-subif)# ip address 10.10.10.19 255.255.255.0
Hostname(config-subif)# end
```