CLI Configuration Guide for Cisco Unified SIP Proxy Release 10.0

August 5, 2021

CLI Configuration Guide for Cisco Unified SIP Proxy Release 10.0

CONTENTS

Overview of Cisco Unified SIP Proxy Release 10.0

Last updated: August 5, 2021

About This Document

This document contains information about how to configure the Cisco Unified SIP Proxy system using the CLI. Use it in conjunction with the *CLI Command Reference for Cisco Unified SIP Proxy Release 10.0*, which lists all the CLI commands.

Administration Interfaces

Cisco Unified SIP Proxy Release 10.0 utilizes both a command-line interface (CLI) and a graphical user interface (GUI).

Command-Line Interface

The CLI is a text-based interface that is accessed through a Telnet or SSH session directly to the Cisco Unified SIP Proxy appliance or, via the Console through the hypervisor. Those familiar with Cisco IOS command structure and routers will see similarities.

The Cisco Unified SIP Proxy commands are structured much like the Cisco IOS CLI commands. However, the Cisco Unified SIP Proxy CLI commands do not affect Cisco IOS configurations. After you log in to Cisco Unified SIP Proxy, the command environment is no longer the Cisco IOS environment.

The CLI is accessible from a PC or server anywhere in the IP network.

Graphical User Interface

Cisco Unified SIP Proxy Release 10.0 also has a GUI that is used to configure and operate the Cisco Unified SIP Proxy system.

For information on using the GUI, see the online help in the application or the *GUI Configuration Guide for Cisco Unified SIP Proxy Release 10.0*.

GUI information is not within the scope of this document.

Commercial Open Source Licensing

Some components of the software created for Cisco Unified SIP Proxy Release 10.0 are provided through open source or commercial licensing. These components and the associated copyright statements can be found at http://www.cisco.com/en/US/products/ps10475/products_licensing_information_listing.html.

Obtaining Documentation and Submitting a Service Request

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

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

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

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and RSS Feeds. Access to most tools on the Cisco Support website requires a Cisco.com username and password.	http://www.cisco.com/techsupport
Use the Cisco Feature Navigator website to find information about platform support and Cisco IOS and Catalyst OS software image support. An account on Cisco.com is not required.	http://www.cisco.com/go/cfn

Initial Configuration Tasks

Last updated: August 5, 2021

Configuring SNMP MIB

About SNMP MIB Support

The Cisco Unified SIP Proxy (Unified SIP Proxy) includes SNMP integration for Release 10.0 with support for Cisco-USP-MIB. It is an enhancement from the SNMP MIB basic support introduced in Unified SIP Proxy Release 8.5. The Cisco Unified SIP Proxy Release 10.0 is SNMP version 2 (SNMPv2c) compliant.

Unified SIP Proxy integrates an SNMP agent and SNMP MIBs to monitor the health and to conduct performance monitoring and data collection for Unified 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 Cisco Unified SIP Proxy Release 10.0 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 Unified 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 Cisco Unified SIP Proxy Release 10.0.
- Upload the CISCO-USP-MIB to the NMS.
- Ensure that MIB browser or NMS provides SNMP v2c compliance.

Restrictions

SNMP MIB support in Cisco Unity SIP Proxy Release 10.0 is known to have the following limitations or restrictions:

- No Support for SNMP Version 3 (SNMPv3)
- Certain MIB objects in the Cisco Unified 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.

Structure

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

- The Unified 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>
- Cisco Unified SIP Proxy Release 10.0 supports the following traps in Cisco-Process-MIB for CPU utilization monitoring:
  - 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 Unified SIP Proxy MIB structure has the following groups and subgroups:

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

Cisco Unified SIP Proxy SNMP MIBs

The Cisco Unified SIP Proxy captures the following in a management information base.

- MIB Objects
- MIB Notifications (Traps)

MIB Objects

The supported Cisco Unified SIP Proxy MIB Objects are:

- cuspScalar
  - cuspCallStats
  - cuspMessageStats
  - cuspThresholdValues
- cuspTable

- cuspNotifControlInfo

Cisco Unified SIP Proxy Release 10.0 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 Unified SIP Proxy scalars. An entry in this table represents Unified 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 Cisco Unified 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 Cisco Unified 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.

MIB	OID	Description
cuspLicenseState	.1.3.6.1.4.1.9.9.827.1.1.5	Gives the current license state of Cisco Unified 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 Cisco Unified SIP Proxy in Megabytes.
cuspMemoryUsed	.1.3.6.1.4.1.9.9.827.1.1.8	Gives the Cisco Unified 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 (Mega Byte).

cuspCallStats

This Unified 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.
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.

Note There is no CLI and GUI equivalent for the data retrieved through MIB objects related to Calls Per Second (CPS) such as cuspCPS, cuspAvgCPSOneMin, cuspMaxCPSOneMin, cuspDroppedCallsOneSec, cuspAvgDroppedCPSOneMin, cuspMaxDroppedCPSOneMin, cuspCallsRoutedOneSec, cuspAvgCallsRoutedOneMin, and cuspMaxCallsRoutedOneMin. For example, GUI provides data for a five-minute average CPS while the MIB object cuspCPS retrieves CPS data only for the last second.

Note CUSP dropped call MIB objects are not updated if the license is in unidentified state.

Note If call rate limit is set to a value lesser than license limit, cuspCallsDroppedExceedingLicense MIB counts calls that are dropped due to call rate limit.

cuspThresholdValues

The Unified 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 Unified 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 Cisco Unified 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 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.

MIB	OID	Description
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.
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.

MIB	OID	Description
cuspSIPMessageQueueOverflowAlertEnable	.1.3.6.1.4.1.9.9.827.1.3.13	Controls the generation of cuspSIPMessageQueueOverflowAlert.

MIB Notifications (Traps)

Unified 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.

Unified SIP Proxy Release 10.0 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

Unified SIP Proxy Release 10.0 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 Cisco Unified 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**

MIB	OID	Description
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 Cisco Unified SIP Proxy memory usage exceeds the cuspMemoryThresholdValue. This notification can be enabled or disabled by setting cuspThresholdAlertEnable.
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 Cisco Unified 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 Cisco Unified 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**

MIB	OID	Description
cuspDiskSpaceThresholdAlert	.1.3.6.1.4.1.9.9.827.0.8	Generated when the Cisco Unified 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.
cuspSIPMessageQueueOverflowAlert	.1.3.6.1.4.1.9.9.827.0.11	Generated when the Cisco Unified SIP Proxy system queue is full. Queue full indicates that either Cisco Unified 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 Cisco Unified 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**

MIB	OID	Description
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.

Configuring Community String

Configure community string to poll data using MIB objects.

Summary Steps

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

Detailed Steps

	Command or Action	Purpose
Step 1	**config** **Example:** `se-10-0-0-0# config terminal`	Enables privileged EXEC mode.
Step 2	**snmp-server community** *community string {RO| RW}* **Example:** `se-10-0-0-0(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:** `se-10-0-0-0(config)# end`	Exits the privileged EXEC mode.
Step 4	**write memory** **Example:** `se-10-0-0-0# write memory`	Stores the configuration in the startup configuration file.

Example

The following example configures Community Strings on the Cisco Unified SIP Proxy:

```
se-10-0-0-0# config terminal
se-10-0-0-0(config)# snmp-server community public RW
se-10-0-0-0(config)# end
se-10-0-0-0# write memory
```

Configuring 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-Exceeded | Extensive-Logging | SIP-Message-Queue-Overflow]**
4. **snmp-server enable traps**
5. **end**
6. **write memory**

Detailed Steps

	Command or Action	Purpose
Step 1	**config** **Example:** `se-10-0-0-0# config terminal`	Enables privileged EXEC mode.
Step 2	**snmp-server host** *IP Address community string* **Example:** `se-10-0-0-0(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:** `se-10-0-0-0(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:** `se-10-0-0-0(config)# snmp-server enable traps`	Enables trap generation from Cisco Unified SIP Proxy to the configured hosts. Traps are sent to the host only when this global command is enabled.
Step 5	**end** **Example:** `se-10-0-0-0(config)# end`	Exits the privileged EXEC mode.
Step 6	**write memory** **Example:** `se-10-0-0-0# write memory`	Stores the configuration in the startup configuration file.

Example

The following example configures SNMP Traps on the Cisco Unified SIP Proxy:

```
se-10-0-0-0# config terminal
se-10-0-0-0(config)# snmp-server host 10.104.54.108 public
se-10-0-0-0(config)# snmp-server enable traps SG-Element
se-10-0-0-0(config)# snmp-server enable traps
se-10-0-0-0(config)# end
se-10-0-0-0# 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 vCUSP **atrace.log** logs.

Configuring 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

	Command or Action	Purpose
Step 1	**config** **Example:** `se-10-0-0-0# config terminal`	Enables privileged EXEC mode.
Step 2	**process cpu threshold type {total} rising** *percentage* **interval** *seconds* **falling** *percentage* **interval** *seconds* **Example:** `se-10-0-0-0(config)# process cpu threshold type {total} rising 80 interval 300 falling 5 interval 300`	Sets the CPU thresholding 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.

Example

The following example configures CPU thresholding values for SNMP traps on the Cisco Unified SIP Proxy:

```
se-10-0-0-0# config terminal
se-10-0-0-0(config)# process cpu threshold type {total} rising 80 interval 300 falling 5
interval 300
```

Configuring 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*.

Summary Steps

3. **enable**
4. **license smart destinationAddr** *url*
5. **license smart httpProxyAddr** *url*
6. **license smart activate cusp** *count*
7. **license smart register token_id** *token*

Detailed Steps

	Command or Action	Purpose
Step 1	**enable** **Example:** `se-10-0-0-0# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**license smart destinationAddr** *https://tools.cisco.com/its/service/oddce/services/DDCEService* **Example:** `se-10-0-0-0# license smart destinationAddr https://tools.cisco.com/its/service/oddce/services/DDCEService`	Connects to the central licensing server.
Step 3	**license smart httpProxyAddr** *10.1.1.1* **Example:** `se-10-0-0-0# license smart httpProxyAddr 10.1.1.1`	Sets the HTTP(S) proxy server address for smart licensing.
Step 4	**license smart activate cusp** *count* **Example:** `se-10-0-0-0# license smart activate cusp 100`	Activates the request number of licenses. The count must be multiple of 5.
Step 5	**license smart register token_id** *token* **Example:** `se-10-0-0-0# license smart register token_id MjgxZjdkY2RtMWY5Ny00YTk4LOI2N2MtNjcxNmYaMTkzZGFhLHE0MjA3MjY0%0AMjI5N34Z8OVAOdmNzSjdIeG4MMHIzTmZubNFzMHhKOTYyeHlUZWQzQzVIM3Jk%0AHVk3MD0A3D%0N`	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 Cisco Unified SIP Proxy:

```
se-10-0-0-0# enable
se-10-0-0-0# license smart destinationAddr
https://tools.cisco.com/its/service/oddce/services/DDCEService
se-10-0-0-0# license smart httpProxyAddr 10.1.1.1
se-10-0-0-0# license smart activate cusp 100
se-10-0-0-0# license smart register token_id
MjgxZjdkY2RtMWY5Ny00YTk4LOI2N2MtNjcxNmYaMTkzZGFhLHE0MjA3MjY0%0AMjI5NDZ8OVAOdmNzSjdIeG4MMHIzTmZubNFzMHhKOTYyeHl67ZWQzQzVIM3Jk%0AHVk3MD0A3D%0N
```

Setting Backup Parameters

About Backup Parameters

Cisco Unified SIP Proxy backup and restore functions use an FTP server to store and retrieve data. The backup function copies the files from Cisco Unified SIP Proxy to the FTP server and the restore function copies the files from the FTP server to Cisco Unified SIP Proxy. The FTP 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 Cisco Unified SIP Proxy 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 FTP server to use for storing Cisco Unified SIP Proxy backup files and the number of backups that are stored before the system overwrites the oldest one.

Prerequisites

- Verify that an FTP administrator or other user who can log in to the FTP server has full permission on the FTP server, such as read, write, overwrite, create, and delete permissions for files and directories.
- Gather the FTP server URL and the username and password of the FTP server login.
- Determine the number of revisions to save before the oldest backup is overwritten.

Summary Steps

8. **configure terminal**
9. **backup server url** *backup-ftp-url* **username** *backup-ftp-usrname* **password** *backup-ftp-password*
10. **backup revisions number** *number*
11. **end**
12. **show backup**

Detailed Steps

	Command or Action	Purpose
Step 1	**configure terminal** se-10-0-0-0# **config terminal**	Enters configuration mode.
Step 2	**backup server url** *ftp-url* **username** *ftp-username* **password** *ftp-password*} **Example:** se-10-0-0-0(config)> **backup server url ftp://main/backups username "admin" password "wxyz"** se-10-0-0-0(config)> **backup server url ftp://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 *ftp-url* value is the URL to the network FTP server where the backup files will be stored. • The *ftp-username* and *ftp-password* values are the username and password for the network FTP server. In the example, **main** is the hostname of the FTP server and **backups** is the directory where backup files are stored.
Step 3	**backup revisions** *number* **Example:** se-10-0-0-0(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:** se-10-0-0-0(config)> **end**	Exits configuration mode.
Step 5	**show backup** **Example:** se-10-0-0-0> **show backup**	Displays the backup server configuration information, including the FTP server URL and the maximum number of backup files available.

Example

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

```
se-10-0-0-0> enable
se-10-0-0-0# configure terminal
se-10-0-0-0(config)> backup revisions 5
se-10-0-0-0(config)> backup server url ftp://10.12.0.1/ftp username "admin" password
"wxyz"
se-10-0-0-0(config)> end
se-10-0-0-0> show backup
Server URL:                             ftp://10.12.0.1/ftp
User Account on Server:
Number of Backups to Retain:            5
se-10-0-0-0>
```

Related Topics

- For information about the CLI commands, see the *CLI Command Reference for Cisco Unified SIP Proxy Release 10.0*.
- For information about backing up and restoring your configuration, see Backing Up and Restoring Data.

Configuring NTP Servers

When you install the Cisco Unified SIP Proxy software, the system gives you the option of adding up to two Network Time Protocol (NTP) servers. You can add additional NTP servers (the system supports up to three NTP servers), remove one or more NTP servers, or display NTP server information using the CLI.

Adding NTP Servers

About Adding NTP Servers

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

Cisco Unified SIP Proxy 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, Cisco Unified SIP Proxy 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

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `se-10-0-0-0# config terminal`	Enters configuration mode.

Step 2	**ntp server** {*hostname* | *ip-address*} [**prefer**]
	
	Example:
	`se-10-0-0-0(config)> ntp server 192.0.2.14`
	`se-10-0-0-0(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:** `se-10-0-0-0(config)> exit`	Exits configuration mode.
Step 4	**show ntp status** **Example:** `se-10-0-0-0> show ntp status`	Displays statistics about the NTP server.
Step 5	**show ntp configuration** **Example:** `se-10-0-0-0> show ntp configuration`	Displays the configured NTP servers.
Step 6	**copy running-config startup-config** **Example:** `se-10-0-0-0> copy running-config startup-config`	Copies the configuration changes to the startup configuration.

Examples of Adding NTP Servers

The following commands configure the NTP server:

```
se-10-0-0-0# configure terminal
se-10-0-0-0(config)> ntp server 192.0.2.14
se-10-0-0-0(config)> exit
se-10-0-0-0>
```

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

```
se-10-0-0-0> show ntp status

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

Removing an NTP Server

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

- Summary Steps, page 22
- Detailed Steps, page 22

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

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `se-10-0-0-0# configure terminal`	Enters configuration mode.

Step 2	**no ntp server** {*hostname* | *ip-address*}
	
	Example:
	`se-10-0-0-0(config)> no ntp server 192.0.2.14`
	`se-10-0-0-0(config)> no ntp server myhost`
		Specifies the hostname or IP address of the NTP server to remove.

Step 3	**exit** **Example:** `se-10-0-0-0(config)> exit`	Exits configuration mode.
Step 4	**show ntp status** **Example:** `se-10-0-0-0> show ntp status`	Displays statistics about the NTP server.
Step 5	**show ntp configuration** **Example:** `se-10-0-0-0> show ntp status`	Displays the configured NTP servers.
Step 6	**copy running-config startup-config** **Example:** `se-10-0-0-0> copy running-config startup-config`	Copies the configuration changes to the startup configuration.

Displaying 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 sample output for the **show ntp associations** command:

```
se-10-0-0-0> show ntp associations

ind assID status  conf reach auth condition  last_event cnt
===========================================================
  1 61253  8000   yes   yes  none    reject
```

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

```
se-10-0-0-0> show ntp servers

     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
 1.100.6.9       0.0.0.0         16 u    - 1024    0    0.000    0.000 4000.00
space reject,       x falsetick,       . excess,          - outlyer
+ candidate,        # selected,        * sys.peer,        o pps.peer
```

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

```
se-10-0-0-0> show ntp source

127.0.0.1: stratum 16, offset 0.000013, synch distance 8.67201
0.0.0.0:        *Not Synchronized*
```

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

```
se-10-0-0-0> show ntp status

NTP reference server :        10.100.6.9
Status:                       reject
Time difference (secs):       0.0
Time jitter (secs):           4.0
```

Related Topics

- For information about the CLI commands, see the *CLI Command Reference for Cisco Unified SIP Proxy Release 10.0*.
- For information about the initial installation of the Cisco Unified SIP Proxy system and the post installation configuration tool, see the *Installation Guide for Cisco Unified SIP Proxy Release 10.0*.
- For information about copying the configuration, see Copying Configurations, page 1.

Setting the Time Zone

When you install the Cisco Unified SIP Proxy software, the system prompts you to set the time zone. If you need to change it, use the **clock timezone** command in Cisco Unified SIP Proxy configuration mode.

To display the time zone, use the **show clock detail** command in module EXEC mode.

Example of Setting the Time Zone

```
se-10-0-0-0# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
se-10-0-0-0(config)> clock timezone
Please identify a location so that time zone rules can be set correctly.
Please select a continent or ocean.
1) Africa            4) Arctic Ocean     7) Australia       10) Pacific Ocean
2) Americas          5) Asia             8) Europe
3) Antarctica        6) Atlantic Ocean   9) Indian Ocean
>? 2
Please select a country.
1) Anguilla                 29) Honduras
2) Antigua & Barbuda        30) Jamaica
3) Argentina                31) Martinique
4) Aruba                    32) Mexico
5) Bahamas                  33) Montserrat
6) Barbados                 32) Netherlands Antilles
7) Belize                   34) Nicaragua
8) Bolivia                  35) Panama
9) Brazil                   36) Paraguay
10) Canada                  37) Peru
11) Caribbean NL            38) Puerto Rico
12) Cayman Islands          39) St Barthelemy
13) Chile                   40) St Kitts & Nevis
14) Colombia                41) St Lucia
15) Costa Rica              42) St Maarten (Dutch)
16) Cuba                    43) St Martin (French)
17) Curacao                 44) St Pierre & Miquelon
18) Dominica                45) St Vincent
19) Dominican Republic      46) Suriname
20) Ecuador                 47) Trinidad & Tobago
21) El Salvador             48) Turks & Caicos Is
22) French Guiana           49) United States
23) Greenland               50) Uruguay
24) Grenada                 51) Venezuela
25) Guadeloupe              52) Virgin Islands (UK)
26) Guatemala               53) Virgin Islands (US)
27) Guyana
28) Haiti
>? 49
Please select one of the following time zone regions.
1) Eastern (most areas)
 2) Eastern - MI (most areas)
 3) Eastern - KY (Louisville area)
 4) Eastern - KY (Wayne)
 5) Eastern - IN (most areas)
 6) Eastern - IN (Da, Du, K, Mn)
 7) Eastern - IN (Pulaski)
 8) Eastern - IN (Crawford)
 9) Eastern - IN (Pike)
10) Eastern - IN (Switzerland)
11) Central (most areas)
```

```
12) Central - IN (Perry)
13) Central - IN (Starke)
14) Central - MI (Wisconsin border)
15) Central - ND (Oliver)
16) Central - ND (Morton rural)
17) Mountain (most areas)
18) Mountain - ID (south); OR (east)
19) Mountain Time - Navajo
20) MMST - Arizona (except navajo)
21) Pacific
22) Alaska (most areas)
23) Alaska - Juneau area
24) Alaska - Sitka area
25) Alaska - Annette Island
26) Alaska - Yakutat
27) Alaska (west)
28) Aleutian Islands
29) Hawaii
>? 21

The following information has been given:

        United States
        Pacific Time

Therefore TZ='America/Los_Angeles' will be used.
Local time is now:      Mon Aug 27 17:23:54 PDT 2018.
Universal Time is now:  Tue Aug 28 00:23:54 UTC 2018.
Is the above information OK?
1) Yes
2) No
>? 1

Save the change to startup configuration and reload the module for the new time zone to
take effect.
se-10-0-0-0(config)>
```

Configuring HTTPS for Administration Web Interface

You can configure the system to allow HTTPS access to Cisco Unified SIP Proxy GUI.

Summary Steps

1. **configure**
2. **crypto key generate rsa label** *labelname* **modulus 1024**
3. **web session security keylabel** *labelname*
4. **end**

Detailed Steps

	Command or Action	Purpose
Step 1	**configure** **Example:** `se-10-0-0-0> configure`	Enters Cisco Unified SIP Proxy configuration mode.

Step 2	**crypto key generate rsa label** *labelname* **modulus 1024**
	
	Example:
	`se-10-0-0-0(cusp-config)> crypto key generate rsa label mykey modulus 1024`
	`Key generation in progress. Please wait...`
	`The label name for key is mykey`
		Generates a self-signed certificate and an RSA private key.

Step 3	**web session security keylabel** *labelname* **Example:** `se-10-0-0-0(cusp-config)> web session security keylabel mykey`	Associates a security key for HTTPS.
Step 4	**end** **Example:** `se-10-0-0-0(cusp-config)> end`	Exits to privileged EXEC mode.

Example of Configuring HTTPS

```
se-10-0-0-0> configure
se-10-0-0-0(cusp)> crypto key generate rsa label mykey modulus 1024
se-10-0-0-0(cusp-config)> web session security keylabel mykey
se-10-0-0-0(cusp-config)> end
```

Configuring Sub-interfaces

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

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 on vCUSP 10.x and later.

Summary Steps

5. **configure**
6. **Interface FastEthernet**
7. **ipaddress**
8. **end**

Detailed Steps

	Command or Action	Purpose
Step 1	**configure** **Example:** `se-10-64-86-166# > conf t`	Enters Cisco Unified SIP Proxy configuration mode.
Step 2	**interface FastEthernet** **Example:** `se-10-64-86-166(config)# interface FastEthernet 0.11`	Enters interface FastEthernet.
Step 3	**ip address** **Example:** `se-10-64-86-166(config-subif)# ip address 10.64.86.159 255.255.255.0`	Configures subinterface for Fastethernet 0.11 under configuration mode.
Step 4	**end** **Example:** `se-10-64-86-166(config-subif)# end`	Exits network command mode.

Example

The following example configures subinterface for Fastethernet:

```
se-10-64-86-166(config)> cusp
se-10-64-86-166(config-subif)# interface FastEthernet 0.11
se-10-64-86-166(config-subif)# ip address 10.64.86.159 255.255.255.0
se-10-64-86-166(config)# end
```

Configuring the Cisco Unified SIP Proxy

Last updated: August 5, 2021

Configuring Logical Networks

Each interface on the Cisco Unified SIP Proxy is associated with a logical network. Logical networks are used to organize server groups, listen points, and other properties. SIP messages are associated with the network on which they arrive.

Summary Steps

1. **cusp**
2. **configure**
3. **sip network** *network*
4. **end network**

Detailed Steps

	Command or Action	Purpose
Step 1	**cusp** **Example:** `se-10-0-0-0> cusp`	Enters Cisco Unified SIP Proxy EXEC mode.
Step 2	**configure** **Example:** `se-10-0-0-0(cusp)> configure`	Enters Cisco Unified SIP Proxy configuration mode.
Step 3	**sip network** *network* **Example:** `se-10-0-0-0(cusp-config)> sip network service-provider`	Creates a network and puts you into network command mode. In this case, the network that is being created is called “service provider”.
Step 4	**end network** **Example:** `se-10-0-0-0(cusp-config-network)> end network`	Exits network command mode.

Example

The following example creates a network called “service-provider”:

```
se-10-0-0-0> cusp
se-10-0-0-0(cusp)> configure
se-10-0-0-0(cusp-config)> sip network service-provider
se-10-0-0-0(cusp-config-network)> end network
```

Configuring Trigger Conditions

You create trigger conditions to allow Cisco Unified SIP Proxy to respond with the appropriate action for various call flows. In general, the more complex the call flow is, the more complex the trigger must be.

Summary Steps

1. **cusp**
2. **configure**
3. **trigger condition** *trigger-condition-name*

4. **sequence** *sequence-number*
5. (Optional) **in-network** *network-name*
6. (Optional) **mid-dialog**
7. end sequence
8. end trigger condition

Detailed Steps

	Command or Action	Purpose
Step 1	**cusp** **Example:** `se-10-0-0-0> cusp`	Enters Cisco Unified SIP Proxy EXEC mode.
Step 2	**configure** **Example:** `se-10-0-0-0(cusp)> configure`	Enters Cisco Unified SIP Proxy configuration mode.
Step 3	**trigger condition** *trigger-condition-name* **Example:** `se-10-0-0-0(cusp-config)> trigger condition call-from-service-provider`	Creates a trigger condition and puts you into trigger command mode. In this case, the trigger that is being created is called “call-from-service-provider”.
Step 4	**sequence** *sequence-number* **Example:** `se-10-0-0-0(cusp-config-trigger)> sequence 1`	Creates a sequence with the specified number and puts you into trigger sequence command mode. The number indicates the order in which triggers are evaluated. In this case, the sequence that is being created is sequence number 1.
Step 5	**in-network** *network-name* **Example:** `se-10-0-0-0(cusp-config-trigger-seq)> in-network service-provider`	Optional. Specifies the incoming network name for the trigger condition. In this case, the incoming network is the “service-provider” network.
Step 6	**mid-dialog** **Example:** `se-10-0-0-0(cusp-config-trigger-seq)> mid-dialog`	Optional. A special trigger that bypasses routing policies on mid-dialog messages.

	Command or Action	Purpose
Step 7	**end sequence** **Example:** `se-10-0-0-0(cusp-config-trigger-seq)> end sequence`	Exits the trigger sequence command mode.
Step 8	**end trigger condition** **Example:** `se-10-0-0-0(cusp-config-trigger)> end trigger condition`	Exits the trigger command mode.

Example

In this example, Cisco Unified SIP Proxy only reacts based on the network the call came in on, so the triggers are simple.

```
se-10-0-0-0> cusp
se-10-0-0-0(cusp)> configure
se-10-0-0-0(cusp-config)> trigger condition call-from-service-provider
se-10-0-0-0(cusp-config-trigger)> sequence 1
se-10-0-0-0(cusp-config-trigger-seq)> in-network service-provider
se-10-0-0-0(cusp-config-trigger-seq)> end sequence
se-10-0-0-0(cusp-config-trigger)> end trigger condition

se-10-0-0-0(cusp-config)> trigger condition mid-dialog
se-10-0-0-0(cusp-config-trigger)> sequence 1
se-10-0-0-0(cusp-config-trigger-seq)> mid-dialog
se-10-0-0-0(cusp-config-trigger-seq)> end sequence
se-10-0-0-0(cusp-config-trigger)> end trigger condition
```

Configuring Server Groups

About Server Groups

Server groups define the elements that Cisco Unified SIP Proxy interacts with for each network. The server group name that is used is inserted into the SIP URI of the outgoing request. Some devices, such as Cisco Unified Communications Manager, validate the URI of requests before processing, which means that the end device might need to be configured with a Fully Qualified Domain Name (FQDN) to allow for this.

Two of the fields for each individual element, q-value and weight, are important to use to specify the priorities of elements, and also for load balancing. Calls are routed to specific elements based on q-value. The element with the highest q-value receives all traffic routed to that server group. If multiple elements have the same q-value, traffic is distributed between them based on the load-balancing option used. The

default load-balancing is based on call-id, but weight can also be used. If weight is used, the percentage of traffic that an element receives is equal to its weight divided by the sum of up elements with the same q-value's weights. The sum of their weights does not need to equal 100. You can change the weights and q-values to configure a different priority or load-balancing scheme.

Summary Steps

1. **cusp**
2. **configure**
3. **server-group sip group** *server-group-name network*
4. **element ip-address** *ipaddress port* **{udp | tcp | tls} [q-value** *q-value*] **[weight** *weight*]
5. **lb-type {global | highest-q | request-uri | call-id | to-uri | weight }**
6. **end server-group**

Detailed Steps

	Command or Action	Purpose
Step 1	**cusp** **Example:** `se-10-0-0-0> cusp`	Enters Cisco Unified SIP Proxy EXEC mode.
Step 2	**configure** **Example:** `se-10-0-0-0(cusp)> configure`	Enters Cisco Unified SIP Proxy configuration mode.
Step 3	**server-group sip group** *server-group-name network* **Example:** `se-10-0-0-0(cusp-config)> server-group sip group sp.example.com service-provider`	Creates a SIP server group and enters server group command mode. In this case, the server group being created is called “sp.example.com” and it uses the network called “service-provider”.
Step 4	**element ip-address** *ipaddress port* **{udp | tcp | tls} [q-value** *q-value*] **[weight** *weight*] **Example:** `se-10-0-0-0(cusp-config-sg)> element ip-address 192.168.10.3 5060 tls q-value 1.0 weight 100`	Creates an IP element for a SIP server group and determines the characteristics of the SIP server group. **Note** You can enter this command multiple times.

	Command or Action	Purpose
Step 5	**lb-type {global | highest-q | request-uri | call-id | to-uri | weight }** **Example:** `se-10-0-0-0(cusp-config-sg)> lb-type weight`	Configures the load-balancing algorithm for the SIP server group. In this example, it specifies that the element will be selected proportional to its weight relative to the weights of other elements of the same q-value.
Step 6	**end server-group** **Example:** `se-10-0-0-0(cusp-config-sg)> end server-group`	Exits the server group command mode.

Example

```
se-10-0-0-0> cusp
se-10-0-0-0(cusp)> configure
se-10-0-0-0(cusp-config)> server-group sip group sp.example.com service-provider
se-10-0-0-0(cusp-config-sg)> element ip-address 192.168.10.3 5060 tls q-value 1.0 weight
100
se-10-0-0-0(cusp-config-sg)> element ip-address 192.168.10.4 5060 tls q-value 1.0 weight
50
se-10-0-0-0(cusp-config-sg)> element ip-address 192.168.10.5 5060 tls q-value 1.0 weight
50
se-10-0-0-0(cusp-config-sg)> lb-type weight
se-10-0-0-0(cusp-config-sg)> end server-group
```

Configuring Route Tables

About Route Tables

You must configure route tables to direct SIP requests to their appropriate destinations. Each route table consists of a set of keys that are matched based on the lookup policy. For example, each key might represent the prefix of a phone number dialed.

Summary Steps

1. **cusp**
2. **configure**
3. **route table** *table-name*
4. **key** *key* **response** *response-code*
5. **key** *key* **target-destination** *target-destination network*

6. **end route table**

Detailed Steps

	Command or Action	Purpose
Step 1	**cusp** **Example:** `se-10-0-0-0> cusp`	Enters Cisco Unified SIP Proxy EXEC mode.
Step 2	**configure** **Example:** `se-10-0-0-0(cusp)> configure`	Enters Cisco Unified SIP Proxy configuration mode.
Step 3	**route table** *table-name* **Example:** `se-10-0-0-0(cusp-config)> route table service-provider-table`	Creates a route table and enters route table command mode. In this case, it creates a route table called "service-provider-table".
Step 4	**key** *key* **response** *response-code* **Example:** `se-10-0-0-0(cusp-config-rt)> key * response 404`	Assigns a response code to a lookup key. In this example, it returns a response of "404" to everything.
Step 5	**key** *key* **target-destination** *target-destination network* **Example:** `se-10-0-0-0(cusp-config-rt)> key 510 target-destination cube-sp.example.com cube-sp`	Replaces the key part of the target destination with a specified value. **Note** You can enter this command multiple times.
Step 6	**end route table** **Example:** `se-10-0-0-0(cusp-config-rt)> end route table`	Exits the route table command mode.

Example

```
se-10-0-0-0> cusp
se-10-0-0-0(cusp)> configure
se-10-0-0-0(cusp-config)> route table service-provider-table
se-10-0-0-0(cusp-config-rt)> key * response 404
se-10-0-0-0(cusp-config-rt)> key 510 target-destination cube-sp.example.com cube-sp
se-10-0-0-0(cusp-config-rt)> end route table
```

Configuring Normalization Policies

Normalization policies modify SIP messages to account for incompatibilities between networks. In this case, the service provider cannot handle phone numbers with the escape sequence "91," so the sequence must be removed from the request-uri and TO header.

Summary Steps

1. **cusp**
2. **configure**
3. **policy normalization** *policy_name*
4. **uri-component update request-uri {user | host | host-port | phone | uri} {all |** *match-string*}
   replace-string
5. **uri-component update header {first | last | all} {user | host | host-port | phone | uri} {all |**
   match-string} replace-string
6. **end policy**

Detailed Steps

	Command or Action	Purpose
Step 1	**cusp** **Example:** `se-10-0-0-0> cusp`	Enters Cisco Unified SIP Proxy EXEC mode.
Step 2	**configure** **Example:** `se-10-0-0-0(cusp)> configure`	Enters Cisco Unified SIP Proxy configuration mode.
Step 3	**policy normalization** *policy-name* **Example:** `se-10-0-0-0(cusp-config)> policy normalization outgoing-norm-policy`	Creates a normalization policy and enters policy normalization command mode. In this example, the normalization policy is called "outgoing-norm-policy".
Step 4	**uri-component update request-uri {user | host | host-port | phone | uri} {all |** *match-string*} *replace-string* **Example:** `se-10-0-0-0(cusp-config-norm)> uri-component update request-uri user ^91 ""`	Configures a normalization policy step that updates a URI component field within a request URI.

	Command or Action	Purpose
Step 5	**uri-component update header {first | last | all} {user | host | host-port | phone | uri} {all |** *match-string}* *replace-string* **Example:** `se-10-0-0-0(cusp-config-norm)> uri-component update TO all user ^91 ""`	Configures a normalization policy step that updates a URI component field within a header of the source message.
Step 6	**end policy** **Example:** `se-10-0-0-0(cusp-config-norm)> end policy`	Exits policy normalization command mode.

Example

```
se-10-0-0-0> cusp
se-10-0-0-0(cusp)> configure
se-10-0-0-0(cusp-config)> policy normalization outgoing-norm-policy
se-10-0-0-0(cusp-config-norm)> uri-component update request-uri user ^91 ""
se-10-0-0-0(cusp-config-norm)> uri-component update TO all user ^91 ""
se-10-0-0-0(cusp-config-norm)> end policy
```

Configuring Lookup Policies

Lookup policies decide how the keys in the route tables are used. Each key represents the beginning of the phone number dialed because each policy states to match the user component of the request-uri against the keys in its route table. The user component of the request-uri is the phone number called. The rule used to match is prefix, which means that the longest prefix match in the route table is used. So if the dialed number is 510-1XX-XXXX, the call is sent to the cme.example.com server group. If the dialed number is 510-XXX-XXXX, the call is sent to the cucm.example.com server group. The four policies in the following example are identical, except that they each refer to their specific table.

Summary Steps

1. **cusp**
2. **configure**
3. **policy lookup** *policy-name*
4. **sequence** *sequence-number table-name* **field** {**in-network | local-ip-address | local-ip-port | remote-ip-address | remote-ip-port**} | **header** {**p-asserted identity| from | to | diversion| remote-party-id**} | **request uri [uri component** {**param| user | phone | host| host-port| uri**}]
5. **rule {exact | prefix | subdomain | subnet | fixed** *length*} **[case-insensitive]**
6. **end sequence**

7. **end policy**

Detailed Steps

	Command or Action	Purpose
Step 1	**cusp** **Example:** `se-10-0-0-0> cusp`	Enters Cisco Unified SIP Proxy EXEC mode.
Step 2	**configure** **Example:** `se-10-0-0-0(cusp)> configure`	Enters Cisco Unified SIP Proxy configuration mode.
Step 3	**policy lookup** *policy-name* **Example:** `se-10-0-0-0(cusp-config)> policy lookup service-provider-policy`	Creates a policy with the specified name and enters policy lookup command mode. In this case, creates a policy called “service-provider-policy”.
Step 4	**sequence** *sequence-number table-name* **field {in-network | local-ip-address | local-ip-port | remote-ip-address | remote-ip-port} | header {p-asserted identity| from | to | diversion| remote-party-id} | request uri [uri component {param| user | phone | host| host-port| uri}]** **Example:** `se-10-0-0-0(cusp-config-lookup)> sequence 1`	Creates a sequence with the specified number and enters policy lookup sequence command mode. Sequences are performed according to the order of their number.
Step 5	**rule {exact | prefix | subdomain | subnet | fixed** *length***} [case-insensitive]** **Example:** `se-10-0-0-0(cusp-config-lookup-seq)> rule prefix`	Creates a rule that determines the routing algorithm for the lookup policy. In this case, it creates a rule that specifies that the lookup policy searches for the longest prefix match.
Step 6	**end sequence** **Example:** `se-10-0-0-0(cusp-config-lookup-seq)> end sequence`	Exits policy lookup sequence command mode.
Step 7	**end policy** **Example:** `se-10-0-0-0(cusp-config-lookup)> end policy`	Exits policy lookup command mode.

Example

```
se-10-0-0-0> cusp
se-10-0-0-0(cusp)> configure
se-10-0-0-0(cusp-config)> policy lookup service-provider-policy
```

```
se-10-0-0-0(cusp-config-lookup)> sequence 1 service-provider-table request-uri
uri-component user
se-10-0-0-0(cusp-config-lookup-seq)> rule prefix
se-10-0-0-0(cusp-config-lookup-seq)> end sequence
se-10-0-0-0(cusp-config-lookup)> end policy
```

Configuring Routing Triggers

Routing triggers correlate trigger conditions with lookup policies. A single policy is chosen based on which corresponding condition is matched. The conditions are evaluated in ascending order based on sequence number. The mid-dialog condition is the first one so that the policy step is skipped for mid-dialog messages. Based on the following configuration, after the INVITE message is successfully routed, all subsequent messages (which are mid-dialog) bypass routing policies.

Summary Steps

1. **cusp**
2. **configure**
3. **trigger routing sequence** *sequence-number* **{by-pass | policy** *policy*} **[condition** *trigger-condition*]

Detailed Steps

	Command or Action	Purpose
Step 1	**cusp** **Example:** `se-10-0-0-0> cusp`	Enters Cisco Unified SIP Proxy EXEC mode.
Step 2	**configure** **Example:** `se-10-0-0-0(cusp)> configure`	Enters Cisco Unified SIP Proxy configuration mode.
Step 3	**trigger routing sequence** *sequence-number* **{by-pass | policy** *policy*} **[condition** *trigger-condition*] **Example:** `se-10-0-0-0(cusp-config)> trigger routing sequence 2 policy service-provider-policy condition call-from-service-provider`	Associates a routing policy with a trigger condition. In this example, the second sequence follows the previously-created policy called "service-provider-policy" and the previously-created trigger called "call-from-service-provider".

Example

```
se-10-0-0-0> cusp
se-10-0-0-0(cusp)> configure
se-10-0-0-0(cusp-config)> trigger routing sequence 1 by-pass condition mid-dialog
se-10-0-0-0(cusp-config)> trigger routing sequence 2 policy service-provider-policy
condition call-from-service-provider
se-10-0-0-0(cusp-config)> trigger routing sequence 3 policy cube-sp-policy condition
call-from-cube-sp
se-10-0-0-0(cusp-config)> trigger routing sequence 4 policy cube-es-policy condition
call-from-cube-es
se-10-0-0-0(cusp-config)> trigger routing sequence 5 policy enterprise-policy condition
call-from-enterprise
```

Configuring Normalization Triggers

Normalization triggers correlate trigger conditions with normalization policies. There are two types of triggers: pre-normalization, which occurs before routing, and post-normalization, which occurs after routing. Similar to routing policies, a special policy bypasses normalization on mid-dialog messages.

Summary Steps

1. **cusp**
2. **configure**
3. **trigger pre-normalization sequence** *sequence-number* {**by-pass** | **policy** *policy*} [**condition** *trigger-condition*]

Detailed Steps

	Command or Action	Purpose
Step 1	**cusp** **Example:** `se-10-0-0-0> cusp`	Enters Cisco Unified SIP Proxy EXEC mode.

	Command or Action	Purpose
Step 2	**configure** **Example:** `se-10-0-0-0(cusp)> configure`	Enters Cisco Unified SIP Proxy configuration mode.
Step 3	**trigger pre-normalization sequence** *sequence-number* **{by-pass | policy** *policy*} [**condition** *trigger-condition*] **Example:** `se-10-0-0-0(cusp-config)> trigger pre-normalization sequence 2 policy outgoing-norm-policy condition call-from-cube-sp`	Configures a pre-normalization algorithm for incoming SIP messages to a normalization policy. In this example, the second sequence follows the previously-created policy called "outgoing-norm-policy" and the previously-created trigger called "call-from-cube-sp".

Example

```
se-10-0-0-0> cusp
se-10-0-0-0(cusp)> configure
se-10-0-0-0(cusp-config)> trigger pre-normalization sequence 1 by-pass condition
mid-dialog
se-10-0-0-0(cusp-config)> trigger pre-normalization sequence 2 policy outgoing-norm-policy
condition call-from-cube-sp
```

Configuring Listen and Record-Route Ports

You must configure listen and record-route ports for each network. For the listen and record-route ports, the actual addresses of the Cisco Unified SIP Proxy module are used. The **sip record-route** command inserts the record-route header into outgoing requests. The **sip listen** command allows for Cisco Unified SIP Proxy to accept incoming requests on that port.

Summary Steps

1. **cusp**
2. **configure**
3. **sip record-route** *network_name* **{tcp | tls | udp}** *ip_address [port]*
4. **sip listen** *network_name* **{tcp | tls | udp}** *ip_address port*

Detailed Steps

	Command or Action	Purpose
Step 1	**cusp** **Example:** `se-10-0-0-0> cusp`	Enters Cisco Unified SIP Proxy EXEC mode.
Step 2	**configure** **Example:** `se-10-0-0-0(cusp)> configure`	Enters Cisco Unified SIP Proxy configuration mode.
Step 3	**sip record-route** *network_name* **{tcp | tls | udp}** *ip_address [port]* **Example:** `se-10-0-0-0(cusp-config)> sip record-route service-provider udp 10.10.10.99 5060`	Enables record-routing for a SIP network. In this example, the "service-provider" network is associated with a record-route configuration and the IP address that populates the record-route header field is "10.10.10.99" and the port that populates the record-route header is 5060.
Step 4	**sip listen** *network_name* **{tcp | tls | udp}** *ip_address port* **Example:** `se-10-0-0-0(cusp-config)> sip listen service-provider udp 10.10.10.99 5060`	Creates a listener that listens for SIP traffic on a specific SIP network, host, and port.

Example

```
se-10-0-0-0> cusp
se-10-0-0-0(cusp)> configure
se-10-0-0-0(cusp-config)> sip record-route service-provider udp 10.10.10.99 5060
se-10-0-0-0(cusp-config)> sip listen service-provider udp 10.10.10.99 5060
```

Configuring a Hostname

If the upstream element is using DNS SRV for routing to the two Cisco Unified SIP Proxies in a network, you must configure the two Cisco Unified SIP Proxies to have the same FQDN by entering the **sip alias** command in Cisco Unified SIP Proxy configuration mode on both Cisco Unified SIP Proxies.

Summary Steps

1. **cusp**

2. **configure**
3. **sip alias** *hostname*

Detailed Steps

	Command or Action	Purpose
Step 1	**cusp** **Example:** `se-10-0-0-0> cusp`	Enters Cisco Unified SIP Proxy EXEC mode.
Step 2	**configure** **Example:** `se-10-0-0-0(cusp)> configure`	Enters Cisco Unified SIP Proxy configuration mode.
Step 3	**sip alias** *hostname* **Example:** `se-10-0-0-0(cusp-config)> sip alias myhost`	Configures the hostname of this instance.

Example

```
se-10-0-0-0> cusp
se-10-0-0-0(cusp)> configure
se-10-0-0-0(cusp-config)> sip alias myhost
```

Configuring Transport Layer Security (TLS)

Creating and Importing a Signed Certificate

Cisco Unified SIP Proxy supports TLS, Transmission Control Protocol (TCP), and User Datagram Protocol (UDP). Establishing TLS connections requires some extra steps because the connections require authentication using signed certificates.

Prerequisites

You need an FTP server or HTTP to import certificate requests.

Summary Steps

1. **configure terminal**
2. **crypto key generate [rsa {label** *label-name* **| modulus** *modulus-size***} | default]**
3. **crypto key certreq label** *label-name* **url {ftp: | http:}**
4. **crypto key import rsa label** *label-name* **{der url {ftp: | http: } | pem { terminal | url {ftp: | http: }} [default]**
5. **crypto key import cer label mykey url ftp:**
6. **offline**
7. **reload**

Detailed Steps

	Command or Action	Purpose
Step 1	**configure terminal** **Example:** `se-10-0-0-0# configure terminal`	Enters configuration mode.
Step 2	**crypto key generate [rsa {label** *label-name* **| modulus** *modulus-size***} | default]** **Example:** `se-10-0-0-0(config)> crypto key generate rsa label mykey modulus 512 default`	Creates an RSA private key.
Step 3	**crypto key certreq label** *label-name* **url {ftp: | http:}** **Example:** `se-10-0-0-0(config)> crypto key certreq label mykey url ftp:`	Creates a certificate request to be signed.
Step 4	**crypto key import rsa label** *label-name* **{der url {ftp: | http: } | pem { terminal | url {ftp: | http: }} [default]** **Example:** `se-10-0-0-0(config)> crypto key import trustcacert label rootCA url ftp:`	After the certificate request is signed, imports the trusted certificate authority (CA) certificate that you used to sign the request.

	Command or Action	Purpose
Step 5	**crypto key import rsa label** *label-name* **{der url {ftp: | http: } | pem { terminal | url {ftp: | http: }} [default]** **Example:** `se-10-0-0-0(config)> crypto key import cer label mykey url ftp:`	After the root CA is imported, imports the signed certificate.

Step 6	**offline**
	
	Example:
	`se-10-0-0-0> offline`
	`!!!WARNING!!!: Putting the system offline will terminate all active calls.`
	`Do you wish to continue[n]?: y`
		Initiates Cisco Unified SIP Proxy offline mode.

Step 7	**reload** **Example:** `se-10-0-0-0(offline)> reload`	Restarts the Cisco Unified SIP Proxy system and enables Cisco Unified SIP Proxy to verify the imported trusted certificate.

Example of Creating a Signed Certificate

```
se-10-0-0-0# configure terminal
se-10-0-0-0(config)> crypto key generate rsa label mykey modulus 512 default
Key generation in progress. Please wait...
The label name for the key is mykey

se-10-0-0-0(config)> crypto key certreq label mykey url ftp:
Address or name of remote host? 192.168.202.216
Username (ENTER if none)? anonymous
Password (not shown)?
Destination path? netmod/mykey.csr
Uploading CSR file succeed

se-10-0-0-0(config)> crypto key import trustcacert label rootCA url ftp:
Import certificate file...
Address or name of remote host? 192.168.202.216
Source filename? netmod/rootCA/cacert.pem
1212 bytes received.

se-10-0-0-0(config)> crypto key import cer label mykey url ftp:
Import certificate file...
Address or name of remote host? 192.168.202.216
Source filename? netmod/mycert.cer
952 bytes received.
Import succeeded
```

What To Do Next

- Import the trusted CA certificates for any of the TLS peer elements.

Configuring TLS on Cisco Unified SIP Proxy

After you import the certificates, you must enable TLS connections. If you want more security, you can create a list of trusted peers. If you create such a list, only connections from those peers are accepted. The peer's hostname entry must be the peer's subjectAltName in its certificate. If subjectAltName is not used in the certificate, the peer's hostname entry must be CN.

Summary Steps

1. **cusp**
2. **configure**
3. **sip tls**
4. **sip tls trusted-peer** {*peer's-hostname*}
5. **sip tls connection-setup-timeout** *{value in seconds}*
6. **sip tls [v1.0 | v1.1 | v1.2]**

Detailed Steps

	Command or Action	Purpose
Step 1	**cusp** **Example:** `se-10-0-0-0> cusp`	Enters Cisco Unified SIP Proxy EXEC mode.
Step 2	**configure** **Example:** `se-10-0-0-0(cusp)> configure`	Enters Cisco Unified SIP Proxy configuration mode.
Step 3	**sip tls** **Example:** `se-10-0-0-0(cusp-config)> sip tls`	Enables the use of SIP TLS connections with other SIP entities, providing secure communication over the Internet.
Step 4	**sip tls trusted-peer** {*peer's-hostname*} **Example:** `se-10-0-0-0(cusp-config)> sip tls trusted-peer example.com`	Creates a list of trusted peers.

	Command or Action	Purpose
Step 5	**sip tls connection-setup-timeout** {*value in seconds*} **Example:** se-10-0-0-0(cusp-config)> **sip tls connection-setup-timeout <1-60>**	It is the time specified in Cisco Unified SIP Proxy by the user to establish connection with the trusted peer. The default value is 1 second. The range of values is 1 to 60 seconds.
Step 6	**sip tls [v1.0 | v1.1 | v1.2]** **Example:** se-10-0-0-0(cusp-config)> **sip tls v1.0**	Enables SIP TLS versions. The default value is all TLS versions with fall-back.The connection between the user and the trusted peer fails to establish when the user tries to connect using the TLS version that the trusted peer does not support. In the case where the trusted peer does not support a specific TLS version, the user retries the connection with the trusted peer using the downgraded version of TLS. For example, if the trusted peer does not support TLS v1.2, then the user retries the connection using TLS v1.1.

Example of Configuring TLS

```
se-10-0-0-0> cusp
se-10-0-0-0(cusp)> configure
se-10-0-0-0(cusp-config)> sip tls
se-10-0-0-0(cusp-config)> sip tls trusted-peer example.com
se-10-0-0-0(cusp-config)> sip tls connection-setup-timeout <1-60>
se-10-0-0-0(cusp-config)> sip tls v1.2
```

Configuring Lite Mode

One of the ways you can configure the performance of the Cisco Unified SIP Proxy is to switch the module to Lite Mode. In Lite Mode, which requires you to disable record-route, the module's performance is boosted. In standard mode, the module processes calls up to the licensed limit.

By default, the module is in standard mode.

For information on the performance difference when using Lite Mode versus standard mode, see the Release Notes for Cisco Unified SIP Proxy Release 10.0.

- Summary Steps, page 19
- Detailed Steps, page 20
- Example, page 20

Summary Steps

1. **cusp**
2. **configure**
3. **lite-mode**

Detailed Steps

	Command or Action	Purpose
Step 1	**cusp** **Example:** `se-10-0-0-0> cusp`	Enters Cisco Unified SIP Proxy EXEC mode.
Step 2	**configure** **Example:** `se-10-0-0-0(cusp)> configure`	Enters Cisco Unified SIP Proxy configuration mode.
Step 3	**lite-mode** **Example:** `se-10-0-0-0(cusp-config)> lite-mode`	Puts the Cisco Unified SIP Proxy module into Lite Mode.

Example

The following example puts the module into Lite Mode:

```
se-10-0-0-0> cusp
se-10-0-0-0(cusp)> configure
se-10-0-0-0(cusp-config)> lite-mode
```

Configuring Performance Control

About Performance Control

One of the ways you can configure the performance of the Cisco Unified SIP Proxy is to restrict the number of calls that the Cisco Unified SIP Proxy can handle.

Summary Steps

1. **cusp**
2. **configure**
3. **call-rate-limit** *limit*

Detailed Steps

	Command or Action	Purpose
Step 1	**cusp** **Example:** `se-10-0-0-0> cusp`	Enters Cisco Unified SIP Proxy EXEC mode.
Step 2	**configure** **Example:** `se-10-0-0-0(cusp)> configure`	Enters Cisco Unified SIP Proxy configuration mode.
Step 3	**call-rate-limit** *limit* **Example:** `se-10-0-0-0(cusp-config)> call-rate-limit 50`	Sets the maximum call rate that the Cisco Unified SIP Proxy can handle.

Example

The following example limits the number of calls that the system can process to 50:

```
se-10-0-0-0> cusp
se-10-0-0-0(cusp)> configure
se-10-0-0-0(cusp-config)> call-rate-limit 50
```

Committing the Configuration

Now you must commit the configuration. Committing the configuration serves two purposes: the configuration becomes active, and is persisted.

- To see the current active configuration, enter the **show configuration active** command.
- To see what the active configuration will be after you commit your changes, enter the **show configuration candidate** command.
- To commit the configuration for this example, enter the following command:

  ```
  se-10-0-0-0(cusp-config)> commit
  ```

Backing Up and Restoring Data

Last updated: August 5, 2021

Note Setting up a backup server is part of the initial configuration process. If you have not already done this, see "Setting Backup Parameters" on page 18.

About Backing Up and Restoring Data

Cisco Unified SIP Proxy backup and restore functions use an FTP server to store and retrieve data. The backup function copies the files from the Cisco Unified SIP Proxy module to the FTP server and the restore function copies the files from the FTP server to the Cisco Unified SIP Proxy application. The FTP server can reside anywhere in the network as long as the backup and restore functions can access it with an IP address or hostname.

We recommend that you back up your configuration files whenever you make changes to the system or application files. Do backups regularly to preserve configuration data.

The system supports the following types of backup:

- All—Backs up all files and data.
- Configuration—Backs up only system and application settings.
- Data—Backs up only routes and application data.

Restrictions for Backing Up and Restoring Data

- You must be in offline mode when you back up or restore the system, so we recommend performing these tasks when call traffic is least impacted. Offline mode terminates all calls.
- Cisco Unified SIP Proxy does not support the following backup and restore capabilities:
  - Scheduled backup and restore operations. The backup and restore procedures begin when the appropriate command is entered.
  - Centralized message storage arrangement. Cisco Unified SIP Proxy backup files cannot be used or integrated with other message stores.
  - Selective backup and restore. Only full backup and restore functions are available. Individual messages or other specific data can be neither stored nor retrieved.

Backing Up Files

About Backing Up Files

Cisco Unified SIP Proxy automatically assigns a backup ID to each backup. Although there are the three different types of backups, the system does not take into account the type of backup when generating the backup ID. Therefore, you will never have two backups with the same backup ID, even if one is a configuration file and the other a data file.

To determine the backup ID of the file you want to restore, use the **show backup server** or **show backup history** commands in either EXEC or offline mode. Those commands list all available backup copies on the remote backup server and their respective backup IDs.

Summary Steps

1. **offline**
2. **backup category** {**all** | **configuration** | **data**}
3. **continue**
4. **show backup history**
5. **show backup server**

Detailed Steps

	Command or Action	Purpose

Step 1	**offline**
	
	Example:
	`se-10-0-0-0# offline`
	`!!!WARNING!!!: Putting the system offline will terminate all active calls.`
	`Do you wish to continue[n]? : y`
		Enters offline mode. All calls are terminated.
		
		Note Cisco Unified SIP Proxy still routes calls in offline mode.

Step 2	**backup category** {**all**	**configuration**	**data**}
			
			Example:
			`se-10-0-0-0(offline)# backup category all`
			`se-10-0-0-0(offline)# backup category configuration`
			`se-10-0-0-0(offline)# backup category data`
				Specifies the type of data to be backed up and stored.

	Command or Action	Purpose
Step 3	**continue** **Example:** `se-10-0-0-0(offline)# continue`	Exits offline mode and returns the system to the previous online mode. The system begins processing new calls and voice messages.
Step 4	**show backup history** **Example:** `se-10-0-0-0> show backup history`	Displays each backup file, its backup ID, the type of data stored in the file, and the success or failure of the backup procedure.
Step 5	**show backup server** **Example:** `se-10-0-0-0> show backup server`	Displays a list of the backup files available on the backup server. The files are grouped by category, with the date of each backup and the backup file ID.

Examples

The following examples display the output from the **show backup history** and **show backup server** commands:

```
se-10-0-0-0> show backup history

blade522> show backup history
#Start Operation
Category: Configuration
Backup Server: ftp://192.168.1.35/pub/cusp_backup
Operation: Backup
Backupid: 1
Date: Tue Oct 21 06:14:30 EDT 2008
Result: Success
Reason:
#End Operation

#Start Operation
Category: Configuration
Backup Server: ftp://192.168.1.35/pub/cusp_backup
Operation: Restore
Backupid: 1
Restoreid: 1
Date: Tue Oct 21 06:17:21 EDT 2008
Result: Success
Reason:
#End Operation

se-10-0-0-0> show backup server

Category: Data
Details of last 5 backups
Backupid: 1
Date: Tue Jul 22 10:55:52 PDT 2008
Description:
Backupid: 2
Date: Tue Jul 29 18:06:33 PDT 2008
Description:
Backupid: 3
Date: Tue Jul 29 19:10:32 PDT 2008
Description:
```

```
Category: Configuration
Details of last 5 backups
Backupid: 1
Date: Tue Jul 22 10:55:48 PDT 2008
Description:
Backupid: 2
Date: Tue Jul 29 18:06:27 PDT 2008
Description:
Backupid: 3
Date: Tue Jul 29 19:10:29 PDT 2008
Description:

se-10-0-0-0>
```

Restoring Files

About Restoring Files

After you create the backup files, you can restore them when needed. Restoring is done in offline mode, which terminates all calls. You should therefore consider restoring files when call traffic is least impacted.

To determine the backup ID of the file you want to restore, use the **show backup server** or **show backup history** commands in either EXEC or offline mode.

Summary Steps

1. **show backup server**
2. **offline**
3. **restore id** *backup_ID* **category** {**all** | **configuration** | **data**}
4. **show backup history**
5. **reload**

Detailed Steps

	Command or Action	Purpose
Step 1	**show backup server** **Example:** `se-10-0-0-0> show backup server`	Lists the data and configuration backup files. Look in the backup ID field for the revision number of the file that you want to restore.

	Command or Action	Purpose

Step 2	**offline**
	
	Example:
	`se-10-0-0-0# offline`
	`!!!WARNING!!!: Putting the system offline will terminate all active calls.`
	`Do you wish to continue[n]? : y`
		Enters offline mode. All calls are terminated.
		
		Note Cisco Unified SIP Proxy still routes calls in offline mode.

Step 3	**restore id** *backup_ID* **category** {**all** | **configuration** | **data**} **Example:** `se-10-0-0-0(offline)# restore id 22 category all`	Specifies the backup ID *v*alue and the file type to be restored.
Step 4	**show backup history** **Example:** `se-10-0-0-0> show backup history`	Displays the success or failure of backup and restore procedures, and also the backup IDs.
Step 5	**reload** **Example:** `se-10-0-0-0(offline)# reload`	Activates the uploaded file information and restarts the Cisco Unified SIP Proxy system.

Related Topics

- For information about setting up the backup server as part of the initial configuration process, see "Setting Backup Parameters" on page 18.
- For information on the CLI commands used to back up and restore the configuration, see the *CLI Command Reference for Cisco Unified SIP Proxy Release 10.0*.

Maintaining the Cisco Unified SIP Proxy System

Last updated: August 5, 2021

Copying Configurations

Use module EXEC commands to copy the startup configuration and running configuration to and from the hard disk on the Cisco Unified SIP Proxy module, the network FTP server, and the network TFTP server.

Note Depending on the specific TFTP server you are using, you might need to create a file with the same name on the TFTP server and verify that the file has the correct permissions before transferring the running configuration to the TFTP server.

Copying the Startup Configuration from the Hard Disk to Another Location

Starting in module EXEC mode, use the following command to copy the startup configuration on the hard disk to another location:

copy startup-config {**ftp:** *user-id:password@ftp-server-url* | **tftp:***tftp-server-url*}

Syntax Description

ftp: *user-id:password@*	Username and password for the FTP server. Include the colon (:) and the at sign (@) in your entry.
ftp-server-url	URL of the FTP server including directory and filename. An example is ftps://server/dir/filename.
tftp:*tftp-server-url*	URL of the TFTP server including directory and filename. An example is tftps://server/dir/filename.

This command is interactive and prompts you for the information. You cannot enter the parameters in one line. In this example, the startup configuration is copied to the FTP server, which requires a username and password to transfer files. The startup configuration file is saved on the FTP server with the filename “start”.

```
se-10-0-0-0> copy startup-config ftp
Address or name of remote host? admin:messaging@ftps://server/dir/start
```

```
Source filename? temp_start
```

The following example shows the startup configuration copied to the TFTP server, which does not require a username and password. The command saves the startup configuration in the TFTP directory called “configs” as a file called “temp_start”.

```
se-10-0-0-0> copy startup-config tftp
Address or name of remote host? tftps://server/dir/temp_start
Source filename? temp_start
```

Copying the Startup Configuration from the Network FTP Server to Another Location

Starting in module EXEC mode, use the following command to copy the startup configuration on the network FTP server to another location:

copy ftp: {nvram:startup-config | running-config | startup-config | system:running-config}

For a description of this command, see the *CLI Command Reference for Cisco Unified SIP Proxy Release 10.0*.

This command is interactive and prompts you for the information. You cannot enter the parameters in one line. The following example illustrates this process. In this example, the FTP server requires a username and password. This command copies the file called “start” that resides in the FTP server directory called “configs” to the startup configuration.

```
se-10-0-0-0> copy ftp: startup-config
!!!WARNING!!! This operation will overwrite your startup configuration.
Do you wish to continue[y]? y
Address or name or remote host? admin:messaging@tftps://server/configs
Source filename? start
```

Copying the Running Configuration from the Hard Disk to Another Location

Starting in module EXEC mode, use the following command to copy the running configuration on the hard disk to another location:

copy running-config {ftp: *user-id:password*@**ftps://***server/dir/filename* **|startup-config | tftp:tftps://***server/dir/filename* **}**

For a description of this command, see the *CLI Command Reference for Cisco Unified SIP Proxy Release 10.0*.

The command works in two ways, depending on where you are copying the command:

- If you copy the running configuration to the startup configuration, enter the command on one line, like in the following example:

  ```
  se-10-0-0-0> copy running-config startup-config
  ```

- If you copy the running configuration to the FTP or TFTP server, this command becomes interactive and prompts you for the information. You cannot enter the parameters in one line. In the following example, the running configuration is copied to the FTP server, which requires a username and password. The running configuration is copied to the directory called “configs” as a file called “saved_start”.

  ```
  se-10-0-0-0> copy running-config ftp:
  Address or name of remote host? admin:messaging@ftps://server/configs
  Source filename? saved_start
  ```

Copying the Running Configuration from the Network TFTP Server to Another Location

Starting in module EXEC mode, use the following command to copy the running configuration from the network TFTP server to another location:

copy tftp: {**running-config** | **startup-config**} **tftps:**//*server/dir/filename*

Syntax Description

running-config	Active configuration on hard disk.
startup-config	Startup configuration on hard disk.
tftp-server-url	URL of the TFTP server.

This command is interactive and prompts you for the information. You cannot enter the parameters in one line. The following example illustrates this process. In this example, the file called "start" that resides in the directory called "configs" on the TFTP server is copied to the startup configuration.

```
se-10-0-0-0> copy tftp: startup-config
!!!WARNING!!! This operation will overwrite your startup configuration.
Do you wish to continue[y]? y
Address or name of remote host? tftps://server/configs
Source filename? start
```

Checking Hard Disk Memory Wear Activity

Cisco Unified SIP Proxy tracks the use and wear of the hard disk memory as log and trace data are saved to the module. To display this data, use the **show interfaces** command in module EXEC mode.

The following is sample output:

```
se-10-0-0-0> show interfaces
GigabitEthernet 0 is up, line protocol is up
    Internet address is 10.10.1.20 mask 255.255.255.0 (configured on router)
        25629 packets input, 1688582 bytes
        0 input errors, 0 dropped, 0 overrun, 0 frame errors
        25634 packets output, 1785015 bytes
        0 output errors, 0 dropped, 0 overrun, 0 collision errors
        0 output carrier detect errors
IDE hd0 is up, line protocol is up
    2060 reads, 32704512 bytes
    0 read errors
    489797 write, 2520530944 bytes
    0 write errors
```

Troubleshooting

Last updated: August 5, 2021

Note Use the information in this chapter in conjunction with the *CLI Command Reference for Cisco Unified SIP Proxy Release 10.0.* That document contains detailed information about each CLI command listed here, including when to use it, how to use it, and any cautionary information.

This chapter contains a brief overview of troubleshooting using the CLI and contains the following sections:

Using CLI Commands to Troubleshoot the System

Cisco technical support personnel may request that you run one or more of these commands when troubleshooting a problem. Cisco technical support personnel provides additional information about the commands at that time.

Caution Some of these commands may impact performance of your system. We strongly recommend that you do not use these commands unless directed to do so by Cisco Technical Support.

About Logging

You can use log messages to help you debug system problems. Log messages are saved to the messages.log file.

Logging and tracing to the hard disk is turned off by default. Executing the **log trace boot** command starts the log and trace functions immediately.

To check the log and trace files on the hard disk, use the **show logs** command in Cisco Unified SIP Proxy EXEC mode. It displays the list of logs available, their size and their dates of most recent modification.

Each file has a fixed length of 10 MB, and tracing or logging stops automatically when the file reaches this length. New files overwrite the old files.

Tip If you cannot view the contents of the log files, copy the log files from Cisco Unified SIP Proxy to an external server and use a text editor, such as **vi**, to display the content.

Log Commands

Cisco Unified SIP Proxy has the following log commands:

- **log console** command
- **log console monitor** command
- **log server** command
- **log trace boot** command
- **log trace buffer save** command
- **show logs** command
- **show trace log** command

Example of Log Output

The following is an example of the log output:

```
se-Module(exec-mping)> show logs

SIZELAST_MODIFIED_TIMENAME
28719Mon Dec 22 14:15:06 EST 2008linux_session.log
2573Fri Dec 19 08:28:13 EST 2008install.log
8117Fri Dec 19 08:27:51 EST 2008dmesg
2274Fri Dec 19 08:27:55 EST 2008syslog.log
10455Thu Dec 18 16:38:13 EST 2008sshd.log.prev
1268Fri Dec 19 08:28:09 EST 2008atrace.log
384 Fri Dec 19 08:27:55 EST 2008debug_server.log
10380Thu Dec 18 16:06:58 EST 2008postgres.log.prev
1361Fri Dec 19 08:28:14 EST 2008sshd.log
5598Fri Dec 19 08:30:13 EST 2008postgres.log
1014Fri Dec 19 08:27:57 EST 2008klog.log
2298494Sun Dec 21 23:30:00 EST 2008messages.log
85292Fri Dec 19 08:25:33 EST 2008shutdown_installer.log
```

Using Trace Commands

To troubleshoot network configuration in Cisco Unified SIP Proxy, use the **trace enable** command in Cisco Unified SIP Proxy EXEC mode.

Cisco Unified SIP Proxy has the following trace commands:

- **log trace boot** command
- **log trace buffer save** command
- **show trace log** command
- **show trace options** command
- **trace disable** command

- **trace enable** command
- **trace level** command

Using Show Commands

In addition to the standard show commands, use the following commands to troubleshoot your Cisco Unified SIP Proxy configuration:

- **show status queue**
- **show status server-group radius** [*server-group-name*]
- **show status server-group sip** [*server-group-name*]
- **show status sip**

Troubleshooting Configuration Changes

Problem You lost some configuration data.

Recommended Action Copy your changes to the running configuration at frequent intervals. See "Copying Configurations" on page 1.

Problem You lost configuration data when you rebooted the system.

Explanation You did not save the data before the reboot.

Recommended Action Use the **copy running-config startup-config** command to copy your changes from the running configuration to the startup configuration. When Cisco Unified SIP Proxy reboots, it reloads the startup configuration. See "Copying Configurations" on page 1.

Note Messages are considered application data and are saved directly to the disk in the startup configuration. (They should be backed up on another server in case of a power outage or a new installation.) All other configuration changes require an explicit "save configuration" operation to preserve them in the startup configuration.

Related Topics

- For information about the CLI commands, see the *CLI Command Reference for Cisco Unified SIP Proxy Release 10.0.*
- For information about copying configurations, see "Copying Configurations" on page 1.

Configuration Example

Last updated: August 5, 2021

The following is an example of what you will see after you have configured your Cisco Unified SIP Proxy system and then enter the **show configuration active verbose** command.

```
se-10-0-0-0(cusp-config)> show configuration active verbose
Building CUSP configuration...
!
server-group sip global-load-balance call-id
server-group sip retry-after 0
server-group sip element-retries udp 3
server-group sip element-retries tls 1
server-group sip element-retries tcp 1
sip alias myhostname
sip dns-srv
 enable
 no naptr
 end dns
!
no sip header-compaction
no sip logging
!
sip max-forwards 70
sip network cube-es standard
 no non-invite-provisional
 allow-connections
 retransmit-count invite-server-transaction 9
 retransmit-count non-invite-client-transaction 9
 retransmit-count invite-client-transaction 5
 retransmit-timer clientTn 64000
 retransmit-timer serverTn 64000
 retransmit-timer T4 5000
 retransmit-timer T2 4000
 retransmit-timer T1 500
 retransmit-timer TU2 32000
 retransmit-timer TU1 5000
 end network
!
sip network cube-sp standard
 no non-invite-provisional
 allow-connections
 retransmit-count invite-server-transaction 9
 retransmit-count invite-client-transaction 5
 retransmit-count non-invite-client-transaction 9
 retransmit-timer T4 5000
 retransmit-timer T2 4000
 retransmit-timer T1 500
 retransmit-timer TU2 32000
 retransmit-timer TU1 5000
 retransmit-timer clientTn 64000
 retransmit-timer serverTn 64000
 end network
!
sip network enterprise standard
 no non-invite-provisional
 allow-connections
 retransmit-count invite-client-transaction 5
```

```
 retransmit-count invite-server-transaction 9
 retransmit-count non-invite-client-transaction 9
 retransmit-timer serverTn 64000
 retransmit-timer T4 5000
 retransmit-timer T2 4000
 retransmit-timer T1 500
 retransmit-timer TU2 32000
 retransmit-timer TU1 5000
 retransmit-timer clientTn 64000
 end network
!
sip network service-provider standard
 no non-invite-provisional
 allow-connections
 retransmit-count invite-server-transaction 9
 retransmit-count non-invite-client-transaction 9
 retransmit-count invite-client-transaction 5
 retransmit-timer serverTn 64000
 retransmit-timer TU1 5000
 retransmit-timer TU2 32000
 retransmit-timer T1 500
 retransmit-timer T2 4000
 retransmit-timer T4 5000
 retransmit-timer clientTn 64000
 end network
!
sip overload reject retry-after 0
!
no sip peg-counting
!
sip privacy service
sip queue message
 drop-policy head
 low-threshold 80
 size 2000
 thread-count 20
 end queue
!
sip queue radius
 drop-policy head
 low-threshold 80
 size 2000
 thread-count 20
 end queue
!
sip queue request
 drop-policy head
 low-threshold 80
 size 2000
 thread-count 20
 end queue
!
sip queue response
 drop-policy head
 low-threshold 80
 size 2000
 thread-count 20
 end queue
!
sip queue st-callback
 drop-policy head
 low-threshold 80
 size 2000
 thread-count 10
```

```
 end queue
!
sip queue timer
 drop-policy none
 low-threshold 80
 size 2500
 thread-count 8
 end queue
!
sip queue xcl
 drop-policy head
 low-threshold 80
 size 2000
 thread-count 2
 end queue
!
route recursion
!
sip tcp connection-timeout 240
sip tcp max-connections 256
sip tls
!
trigger condition call-from-cube-es
 sequence 1
  in-network cube-es
  end sequence
 end trigger condition
!
trigger condition call-from-cube-sp
 sequence 1
  in-network cube-sp
  end sequence
 end trigger condition
!
trigger condition call-from-enterprise
 sequence 1
  in-network enterprise
  end sequence
 end trigger condition
!
trigger condition call-from-service-provider
 sequence 1
  in-network service-provider
  end sequence
 end trigger condition
!
trigger condition mid-dialog
 sequence 1
  mid-dialog
  end sequence
 end trigger condition
!
accounting
 no enable
 no client-side
 no server-side
 end accounting
!
server-group sip group cme.example.com enterprise
 element ip-address 192.168.10.6 5060 tls q-value 1.0 weight 0
 failover-resp-codes 503
 lbtype global
 ping
 end server-group
```

```
!
server-group sip group cube-es.example.com cube-es
 element ip-address 192.168.20.4 5060 tls q-value 1.0 weight 0
 element ip-address 192.168.20.3 5060 tls q-value 1.0 weight 0
 failover-resp-codes 503
 lbtype global
 ping
 end server-group
!
server-group sip group cube-sp.example.com cube-sp
 element ip-address 10.10.20.3 5060 tls q-value 1.0 weight 0
 element ip-address 10.10.20.4 5060 tls q-value 1.0 weight 0
 failover-resp-codes 503
 lbtype global
 ping
 end server-group
!
server-group sip group cucm.example.com enterprise
 element ip-address 192.168.10.4 5060 tls q-value 1.0 weight 50
 element ip-address 192.168.10.5 5060 tls q-value 1.0 weight 50
 element ip-address 192.168.10.3 5060 tls q-value 1.0 weight 100
 failover-resp-codes 503
 lbtype weight
 ping
 end server-group
!
server-group sip group sp.example.com service-provider
 element ip-address 10.10.10.3 5060 udp q-value 1.0 weight 0
 failover-resp-codes 503
 lbtype global
 ping
 end server-group
!
route table cube-es-table
 key * response 404
 key 5101 target-destination cme.example.com enterprise
 key 510 target-destination cucm.example.com enterprise
 end route table
!
route table cube-sp-table
 key * target-destination sp.example.com service-provider
 end route table
!
route table enterprise-table
 key * response 404
 key 5101 target-destination cme.example.com enterprise
 key 91 target-destination cube-es.example.com cube-es
 key 510 target-destination cucm.example.com enterprise
 end route table
!
route table service-provider-table
 key * response 404
 key 510 target-destination cube-sp.example.com cube-sp
 end route table
!
policy normalization outgoing-norm-policy
 uri-component update TO all user ^91 ""
 uri-component update request-uri user ^91 ""
 end policy
!
policy lookup cube-es-policy
 sequence 1 cube-es-table request-uri uri-component user
  rule prefix
  end sequence
```

```
 end policy
!
policy lookup cube-sp-policy
 sequence 1 cube-sp-table request-uri uri-component user
  rule prefix
  end sequence
 end policy
!
policy lookup enterprise-policy
 sequence 1 enterprise-table request-uri uri-component user
  rule prefix
  end sequence
 end policy
!
policy lookup service-provider-policy
 sequence 1 service-provider-table request-uri uri-component user
  rule prefix
  end sequence
 end policy
!
trigger routing sequence 5 policy enterprise-policy condition call-from-enterpri
se
trigger routing sequence 4 policy cube-es-policy condition call-from-cube-es
trigger routing sequence 3 policy cube-sp-policy condition call-from-cube-sp
trigger routing sequence 2 policy service-provider-policy condition
call-from-service-provider
trigger routing sequence 1 by-pass condition mid-dialog
trigger pre-normalization sequence 2 policy outgoing-norm-policy condition
call-from-cube-sp
trigger pre-normalization sequence 1 by-pass condition mid-dialog
!
no server-group sip global-ping
!
sip listen service-provider udp 10.10.10.99 5060
sip listen cube-sp tls 10.10.20.99 5060
sip listen cube-es tls 192.168.20.99 5060
sip listen enterprise tls 192.168.10.99 5060
!
sip record-route cube-es tls 192.168.20.99 5060
sip record-route service-provider udp 10.10.10.99 5060
sip record-route cube-sp tls 10.10.20.99 5060
sip record-route enterprise tls 192.168.10.99 5060
!
end
se-10-0-0-0(cusp-config)>
```

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U