Configuring System Components

Last Updated: June 28, 2007

Command-line interface (CLI) commands are available to configure Cisco Unity Express system components. You enter some commands in EXEC mode and others in configuration mode.

This chapter describes how to configure the following basic Cisco Unity Express components:

- SIP parameters that Cisco Unity Express needs to communicate with Cisco Unified Communications Manager Express (Cisco Unified CME).
- JTAPI parameters that Cisco Unity Express needs to communicate with Cisco Unified Communications Manager.
- Other Cisco Unity Express system components such as Prompts, Scripts, Applications, Triggers, and so on.

All the procedures in this chapter can be implemented using either CLI commands or the graphical user interface (GUI) options. Use the CLI procedures for:

- Bulk provisioning
- Scripting
- Upgrading
- Troubleshooting systems.

This chapter contains the following procedures for configuring Cisco Unity Express system components:

Configuring SIP Call Control Parameters

This section contains the following sections:

Configuring the SIP Proxy Server Location for Cisco Unity Express

The Session Initiation Protocol (SIP) proxy server resides on the router where Cisco Unified CME is installed. Beginning in Cisco Unity Express 2.1, Cisco Unified CME can be installed on a different router from where the Cisco Unity Express hardware and software is installed. The SIP proxy server location information must be configured properly to enable all communications between Cisco Unity Express and Cisco Unified CME. The SIP proxy server also enables the message waiting indicators (MWIs) to work with the Cisco Unity Express voice-mail application.

Required Data for This Procedure

The following information is required to configure the SIP proxy server:

- Hostname or IP address of the router where the SIP proxy server resides
- UDP port of the router where the SIP proxy server resides

SUMMARY STEPS

1. **config t**
2. **ccn subsystem sip**
3. **gateway address** *ip-address*
4. **gateway port** *port-number*
5. **end**
6. **exit**
7. **show ccn subsystem sip**
8. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**config t** **Example:** `se-10-0-0-0# config t`	Enters configuration mode.
Step 2	**ccn subsystem sip** **Example:** `se-10-0-0-0# ccn subsystem sip`	Enters SIP configuration mode.
Step 3	**gateway address** *ip-address* **Example:** `se-10-0-0-0(config-sip)# gateway address 10.100.6.9`	Specifies the hostname or IP address of the router where the SIP proxy server resides.
Step 4	**gateway port** *port-number* **Example:** `se-10-0-0-0(config-sip)# gateway port 5060`	Specifies the UDP port number on which the SIP proxy server listens for incoming SIP messages. The default value is 5060. **Note** We strongly recommend that you do not change this port number.
Step 5	**end** **Example:** `se-10-0-0-0(config-sip)# end`	Exits SIP configuration mode.
Step 6	**exit** **Example:** `se-10-0-0-0(config)# exit`	Exits configuration mode.
Step 7	**show ccn subsystem sip** **Example:** `se-10-0-0-0# show ccn subsystem sip`	Displays the SIP subsystem parameters.
Step 8	**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

The following example illustrates the **show ccn subsystem sip** output, which displays the SIP gateway IP address and SIP port number:

```
se-10-0-0-0# show ccn subsystem sip

SIP Gateway:                            10.100.6.9
SIP Port Number:                        5060
DTMF Relay:                             sip-notify,sub-notify
MWI Notification:                       sub-notify
Transfer Mode:                          refer-consult
SIP RFC Compliance:                     Pre-RFC3261
se-10-0-0-0#
```

Configuring the Call Transfer Mode

Cisco Unity Express permits configuration of attended and semiattended call transfer modes in addition to blind transfers.

SUMMARY STEPS

1. **config t**
2. **ccn subsystem sip**
3. **transfer-mode** {**attended** | **semi-attended** | **blind refer** | **blind bye-also**]}
4. **end**
5. **end**
6. **show ccn subsystem sip**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**config t** **Example:** `se-10-0-0-0# config t`	Enters configuration mode.
Step 2	**ccn subsystem sip** **Example:** se-10-0-0-0(config)# **ccn subsystem sip**	Enters SIP configuration mode.

	Command or Action	Purpose
Step 3	**transfer-mode** {**attended** | **semi-attended** | **blind refer** | **blind bye-also**]} **Example:** se-10-0-0-0(config-sip)# **transfer-mode blind refer**	Specifies the transfer mode. • **attended**—Transfers calls in attended mode using the REFER method. The transfer is completed when the destination extension answers the call. • **semi-attended**—Transfers calls in semi-attended mode using the REFER method. The transfer is completed when the destination extension is ringing. • **blind refer**—Transfers calls without consulting using the REFER method. • **blind bye-also**—Transfers calls without consulting using the BYE/ALSO method. Cisco Unity Express uses this method if the remote end does not support REFER. This is the default value.
Step 4	**end** **Example:** se-10-0-0-0(config-sip)# **end**	Exits SIP configuration mode.
Step 5	**end** **Example:** se-10-0-0-0(config)# **end**	Exits configuration mode.
Step 6	**show ccn subsystem sip** **Example:** se-10-0-0-0# **show ccn subsystem sip**	Displays SIP configuration parameters.

Examples

Following is example output of the **show ccn subsystem sip** command.

```
se-10-0-0-0# show ccn subsystem sip

SIP Gateway:        172.19.167.208
SIP Port Number:    5060
DTMF Relay:         sip-notify rtp-nte
MWI Notification:   outcall
Transfer Mode:      blind (REFER)
SIP RFC Compliance: Pre-RFC3261
```

Configuring DTMF Options

Several options are available for handling incoming and outgoing DTMF signals for SIP calls from Cisco Unified CME and Cisco SRST mode.

Cisco Unity Express provides the following options for transferring DTMF signals for incoming and outgoing SIP calls.

- **rtp-nte**—Uses the media path to relay incoming and outgoing DTMF signals.

  To use the **rtp-nte** option, verify that the Cisco IOS SIP gateway is configured to use RTP-NTE for SIP calls, as shown in the following example:

```
dial-peer voice 1000 voip
   destination-pattern 6700
   session protocol sipv2
   session target ipv4:10.100.9.6
   dtmf-relay rtp-nte
   codec g711ulaw
   no vad
```

- **sub-notify**—Uses Subscribe and Notify messages to relay incoming DTMF signals to Cisco Unity Express. This option is not available for outgoing DTMF signals from Cisco Unity Express.
- **info**—Uses the Info message to relay outgoing DTMF signals from Cisco Unity Express to the Cisco IOS SIP gateway. This option is not available for incoming DTMF signals to Cisco Unity Express.
- **sip-notify**—Uses Unsolicited-Notify messages for incoming and outgoing DTMF signals.

  To use the **sip-notify** option, verify that the Cisco IOS SIP gateway is configured to use Unsolicited NOTIFY for SIP calls, as shown in the following example:

```
dial-peer voice 1 voip
   destination-pattern 6700
   session protocol sipv2
   session target ipv4:10.100.9.6
   dtmf-relay sip-notify
   codec g711ulaw
   no vad
```

You can configure more than one option for transferring DTMF signals. The order in which you configure the options determines their order of preference.

Table 5 shows the various option combinations, the remote end capability, and the signaling option for incoming and outgoing DTMF signals.

Table 5 DTMF Relay Option Combinations

Cisco Unity Express Configuration	Option Supported at Remote End	Option for Incoming DTMF to Cisco Unity Express	Option for Outgoing DTMF from Cisco Unity Express
sub-notify	—	sub-notify	no DTMF
info	—	no DTMF	info
rtp-nte	rtp-nte	rtp-nte	rtp-nte
sip-notify	sip-notify	sip-notify	sip-notify
sip-notify, rtp-nte	rtp-nte, sip-notify	sip-notify[1]	sip-notify[1]
sip-notify, rtp-nte	rtp-nte	rtp-nte	rtp-nte
sip-notify, info	sip-notify	sip-notify	sip-notify
sip-notify, info	no support[2]	no DTMF	info
sip-notify, sub-notify	sip-notify	sip-notify	sip-notify

Table 5 DTMF Relay Option Combinations (continued)

Cisco Unity Express Configuration	Option Supported at Remote End	Option for Incoming DTMF to Cisco Unity Express	Option for Outgoing DTMF from Cisco Unity Express
sip-notify, sub-notify	no support[2]	sub-notify	sub-notify
sip-notify, rtp-nte, info	rtp-nte	rtp-nte	rtp-nte
sip-notify, rtp-nte, info	sip-notify	sip-notify	sip-notify
sip-notify, rtp-nte, info	no support[2]	no DTMF	info
sip-notify, rtp-nte, sub-notify	rtp-nte	rtp-nte	rtp-nte
sip-notify, rtp-nte, sub-notify	sip-notify	sip-notify	sip-notify
sip-notify, rtp-nte, sub-notify	no support[2]	sub-notify	no DTMF
sub-notify, info	—	sub-notify	info
rtp-nte, sub-notify	rtp-nte	rtp-nte	rtp-nte
rtp-nte, sub-notify	no support[2]	sub-notify	no DTMF
rtp-nte, info	rtp-nte	rtp-nte	rtp-nte
rtp-nte, info	no support[2]	no DTMF	info
sip-notify, rtp-nte, sub-notify, info	sip-notify, rtp-nte	sip-notify	sip-notify
sip-notify, rtp-nte, sub-notify, info	rtp-nte	rtp-nte	rtp-nte
sip-notify, rtp-nte, sub-notify, info	no support[2]	sub-notify	info

1. For incoming call. For outgoing call, the remote end decides between rtp-nte and sip-notify.
2. No support for rtp-nte and sip-notify.

SUMMARY STEPS

1. **config t**
2. **ccn subsystem sip**
3. **dtmf-relay** {**rtp-nte** | **sub-notify** | **info** | **sip-notify**}

   To configure more than one signal option, specify them using a single **dtmf-relay** command.
4. **end**
5. **end**
6. **show ccn subsystem sip**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**config t** **Example:** `se-10-0-0-0# config t`	Enters configuration mode.
Step 2	**ccn subsystem sip** **Example:** se-10-0-0-0(config)# **ccn subsystem sip**	Enters SIP configuration mode.
Step 3	**dtmf-relay** {**rtp-nte** | **sub-notify** | **info** | **sip-notify**} **Example:** se-10-0-0-0(config-sip)# **dtmf-relay sip-notify rtp-nte**	Specifies the DTMF signal handling option. Use a single **dtmf-relay** command to specify more than one DTMF option. • **rtp-nte**—Uses the media path to relay incoming and outgoing DTMF signals. **Note** Verify that the Cisco IOS gateway has a dial-peer configured to use **rtp-nte**. • **sub-notify**—Uses Subscribe and Notify messages to relay for incoming DTMF signals to Cisco Unity Express. • **info**—Uses the Info message to relay outgoing DTMF signals from Cisco Unity Express to the Cisco IOS SIP gateway. • **sip-notify**—Uses Unsolicited-Notify messages to relay incoming and outgoing DTMF signals. **Note** Verify that the Cisco IOS gateway has a dial-peer configured to use **sip-notify**.
Step 4	**end** **Example:** se-10-0-0-0(config-sip)# **end**	Exits SIP configuration mode.
Step 5	**end** **Example:** se-10-0-0-0(config)# **end**	Exits configuration mode.
Step 6	**show ccn subsystem sip** **Example:** se-10-0-0-0# **show ccn subsystem sip**	Displays SIP configuration parameters.

Examples

The following example displays the output of the **show ccn subsystem sip** command.

```
se-10-0-0-0# show ccn subsystem sip

SIP Gateway:         172.19.167.208
SIP Port Number:     5060
DTMF Relay:          sip-notify rtp-nte
MWI Notification:    outcall
Transfer Mode:       consult (REFER)
SIP RFC Compliance: Pre-RFC3261
```

Configuring the MWI Notification Option

Cisco Unity Express expands MWI status update capability to include Cisco Unified Communications Manager and Cisco SRST mode. Three notification options are available:

- Outcall Notification (not available in Cisco SRST Mode), page 41
- Sub-Notify Notification, page 42
- Unsolicited Notification, page 42

From the GUI, select **Voice Mail > Message Waiting Indicators > Settings** to configure the MWI notification option.

Outcall Notification (not available in Cisco SRST Mode)

Only Cisco Unified CME can use the SIP **outcall** mechanism for generating MWI notifications. Outcall will not work in Cisco SRST mode.

Note If the MWI notification option is **outcall,** configure the MWI on and off extensions. See "Configuring the MWI On and Off Extensions (not available in Cisco SRST Mode)" on page 45.

The **outcall** option is available for backward compatibility. We recommend that you use either **sub-notify** or **unsolicited** for the MWI notification option.

To use the **outcall** option, Cisco Unified CME must configure two ephone-dns that are registered to receive MWI notifications as follows:

```
ephone-dn 30
  number 8000....
  mwi on
.
.
ephone-dn 31
  number 8001....
  mwi off
```

Note The number of dots in the above example must be equal to the extension length of the phones connected to Cisco Unified CME.

Sub-Notify Notification

Both Cisco Unified CME and Cisco Unified Communications Manager in SRST mode can use the **sub-notify** mechanism for generating MWI notifications. With this mechanism, the MWI notifications will reflect the accurate status of messages in a subscriber's voice mailbox.

After an ephone-dn is configured with the **sub-notify** option, Cisco Unified CME sends a Subscribe message to Cisco Unity Express to register the phone for MWI notifications. When a new voice message arrives in the voice mailbox for the ephone-dn, Cisco Unity Express updates the MWI status. If Cisco Unity Express does not receive the Subscribe message for the ephone-dn, Cisco Unity Express will not update the MWI status when a new message arrives.

To use the **sub-notify** option, Cisco Unified CME must configure each ephone-dn that is registered to receive MWI notifications as follows:

For Cisco IOS Releases Prior to 12.3(11)T7

```
sip-ua
.
.
   mwi-server ipv4:10.100.9.6 transport udp port 5060
   number 2010
.
ephone-dn 35
  mwi sip
```

For Cisco IOS Releases 12.3(11)T7 and Later Releases

```
sip-ua
.
.
   mwi-server ipv4:10.100.9.6 transport udp port 5060
   number 2010
.
ephone-dn 35
  mwi sip
```

For Cisco SRST Mode

```
sip-ua
.
.
   mwi-server ipv4:10.100.9.6 transport udp port 5060
   number 2010
.
call-manager-fallback.
  mwi relay
```

Note The SIP server IP address used in these commands must be the IP address of Cisco Unity Express. In the examples shown above, this is 10.100.9.6.

Unsolicited Notification

Both Cisco Unified CME and Cisco Unified Communications Manager in SRST mode can use the **unsolicited** mechanism for generating MWI notifications. With this mechanism, the MWI notifications will reflect the accurate status of messages in a subscriber's voice mailbox.

The **unsolicited** option does not require Cisco Unified CME to send a subscription request for each ephone-dn to Cisco Unity Express for MWI notifications. Cisco Unity Express sends Notify messages to Cisco Unified CME whenever the voice mailbox for any ephone-dn receives a new message. In this way, the MWI status reflects the current voice mailbox message status.

To use the **unsolicited** option, Cisco Unified CME must configure each ephone-dn that is registered to receive MWI notifications as follows:

For Cisco IOS Releases Prior to 12.3(11)T7

```
telephony-service
.
.
   mwi sip-server 10.100.9.6 transport udp port 5060 unsolicited
   number 2010
.
ephone-dn 35
  mwi sip
```

For Cisco IOS Release 12.3(11)T7 and Later Releases

```
sip-ua
.
.
   mwi-server ipv4:10.100.9.6 transport udp port 5060 unsolicited
   number 2010
.
ephone-dn 35
  mwi sip

For Cisco SRST Mode
sip-ua
.
.
   mwi-server ipv4:10.100.9.6 transport udp port 5060 unsolicited
   number 2010
.
call-manager-fallback.
  mwi relay
```

Note The SIP server IP address used in these commands must be the IP address of Cisco Unity Express. In the examples shown above, this is 10.100.9.6.

SUMMARY STEPS

1. **config t**
2. **ccn subsystem sip**
3. **mwi sip** {**outcall** | **sub-notify** | **unsolicited**}
4. **end**
5. **end**
6. **show ccn subsystem sip**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**config t** **Example:** `se-10-0-0-0# config t`	Enters configuration mode.
Step 2	**ccn subsystem sip** **Example:** se-10-0-0-0(config)# **ccn subsystem sip**	Enters SIP configuration mode.
Step 3	**mwi sip** {**outcall** | **sub-notify** | **unsolicited**} **Example:** se-10-0-0-0(config-sip)# **mwi sip sub-notify**	Specifies the MWI notification methods for SIP calls. The default is **outcall**. • **outcall** —Sends MWI notifications using SIP outcall. • **sub-notify**—Sends MWI notifications using SIP Notify. • **unsolicited**—Sends MWI notifications using SIP Unsolicited Notify.
Step 4	**end** **Example:** se-10-0-0-0(config-sip)# **end**	Exits SIP configuration mode.
Step 5	**end** **Example:** se-10-0-0-0(config)# **end**	Exits configuration mode.
Step 6	**show ccn subsystem sip** **Example:** se-10-0-0-0# **show ccn subsystem sip**	Displays SIP configuration parameters.

Examples

The following example displays the output of the **show ccn subsystem sip** command.

```
se-10-0-0-0# show ccn subsystem sip

SIP Gateway:        172.19.167.208
SIP Port Number:    5060
DTMF Relay:         sip-notify, sub-notify
MWI Notification:   sub-notify
Transfer Mode:      consult (REFER)
```

Configuring the MWI On and Off Extensions (not available in Cisco SRST Mode)

Cisco Unity Express uses the MWI on and off extensions with the affected telephone extension to generate a SIP call to Cisco Unified CME, which changes the status of the telephone's MWI light.

This configuration is required only if the MWI notification option is configured as **outcall**. (See the earlier section "Configuring the MWI Notification Option" on page 41.)

Prerequisites

Verify that the MWI on and off extensions are configured on Cisco Unified CME; otherwise, the MWI light will not work.

Required Data for This Procedure

The following information is required to configure the MWI on and off extensions:

- Extension number dedicated to the MWI on extension
- Extension number dedicated to the MWI off extension

SUMMARY STEPS

1. **config t**
2. **ccn application ciscomwiapplication**
3. **parameter strMWI_ON_DN** *on-extension*
4. **parameter strMWI_OFF_DN** *off-extension*
5. **end**
6. exit
7. **copy running-config startup-config**

DETAILED STEPS

	Command of Action	Purpose
Step 1	**config t** **Example:** `se-10-0-0-0# config t`	Enters configuration mode.
Step 2	**ccn application ciscomwiapplication** **Example:** `se-10-0-0-0(config)# ccn application ciscomwiapplication`	Enters configuration mode for the MWI application.
Step 3	**parameter strMWI_ON_DN** *on-extension* **Example:** `se-10-0-0-0(config-application)# parameter strMWI_ON_DN 8000`	Assigns the *on-extension* value as the MWI on extension. Use the same on extension as configured on Cisco Unified CME.

	Command of Action	Purpose
Step 4	**parameter strMWI_OFF_DN** *off-extension* **Example:** `se-10-0-0-0(config-application)# parameter strMWI_OFF_DN 8001`	Assigns the *off-extension* value as the MWI off extension. Use the same off extension as configured on Cisco Unified CME.
Step 5	**end** **Example:** `se-10-0-0-0(config-application)# end`	Exits application configuration mode.

Step 6	`exit`
	
	Example:
	`se-10-0-0-0(config# exit`
		Exits configuration mode.

Step 7	**copy running-config startup-config** **Example:** `se-10-0-0-0# copy running-config startup-config`	Copies the configuration changes to the startup configuration.

Configuring Cisco Unified CME SIP Options for RFC Compliance

Cisco IOS Release 12.4(2)T and earlier releases are not RFC3261 compliant. The lack of compliance causes the Cisco Unity Express software not to interoperate properly with those older Cisco IOS releases when sip-notify or sub-notify are used for DTMF. Cisco Unity Express provides the **protocol** command to ensure compatibility with all Cisco IOS releases.

Required Data for This Procedure

The release number of the Cisco IOS software running on your call platform.

SUMMARY STEPS

1. **config t**
2. **ccn subsystem sip**
3. **protocol** {**pre-rfc3261** | **rfc3261**}
4. **end**
5. **exit**
6. **show ccn subsystem sip**

DETAILED STEPS

	Command of Action	Purpose
Step 1	**config t** **Example:** `se-10-0-0-0# config t`	Enters configuration mode.
Step 2	**ccn subsystem sip** **Example:** `se-10-0-0-0(config-sip)# ccn subsystem sip`	Enters configuration mode for the SIP subsystem.
Step 3	**protocol** {**pre-rfc3261** | **rfc3261**} **Example:** `se-10-0-0-0(config-sip)# protocol rfc3261`	Assigns the protocol type for RFC 3261 compatibility. • **pre-rfc3261**—Use this option if your call platform uses a Cisco IOS release prior to 12.4(2)T. This is the default value. • **rfc3261**—Use this option if your call platform uses Cisco IOS Release 12.4(2)T or a later release.
Step 4	**end** **Example:** `se-10-0-0-0(config-sip)# end`	Exits SIP subsystem configuration mode.

Step 5	`exit`
	
	Example:
	`se-10-0-0-0(config# exit`
		Exits configuration mode.

Step 6	**show ccn subsystem sip** **Example:** `se-10-0-0-0# show ccn subsystem sip`	Displays the configured SIP subsystem parameters.

Example

The following example sets the SIP option to RFC 3261 for call platforms using Cisco IOS Release 12.4(2)T or a later release.

```
se-10-0-0-0# config t
se-10-0-0-0(config)# ccn subsystem sip
se-10-0-0-0(config-sip)# protocol rfc3261
se-10-0-0-0(config-sip)# end
se-10-0-0-0(config)# exit
se-10-0-0-0#
```

Following is example output of the **show ccn subsystem sip** command.

```
se-10-0-0-0# show ccn subsystem sip
SIP Gateway:                            10.10.5.1
SIP Port Number:                        5060
DTMF Relay:                             sip-notify,sub-notify
```

```
MWI Notification:                       sub-notify
Transfer Mode:                          refer-consult
SIP RFC Compliance:                     RFC3261
```

Configuring JTAPI Parameters (Cisco Unified Communications Manager Only)

Use this procedure to configure the parameters that Cisco Unity Express needs to communicate with Cisco Unified Communications Manager.

Cisco Unified Communications Manager and Cisco Unity Express Version Compatibility

Cisco Unity Express can be configured to work with Cisco Unified Communications Manager 4.1, 4.2, 5.0, 5.1, and and 6.0. The following scenarios apply when installing Cisco Unity Express with a different version of Cisco Unified Communications Manager, or upgrading the Cisco Unified Communications Manager version:

- By default, Cisco Unity Express is set up to work with Cisco Unified Communications Manager 4.1. Once you configure the IP Address or Hostname of the Cisco Unified Communications Manager, you must reload Cisco Unity Express module for the configuration to take effect. After this reload, Cisco Unity Express automatically reloads again if the configured Cisco Unified Communications Manager version is different from 4.1.
- If the Cisco Unified Communications Manager server being used by Cisco Unity Express is upgraded, Cisco Unity Express reloads and updates its system files to work with the new version of Cisco Unified Communications Manager. No further action from you is required.

Caution Cisco Unity Express 3.1 does not support versions of Cisco Unified Communications Manager prior to 4.1. If you are using an earlier version of Cisco Unified Communications Manager, you must upgrade to 4.1 or a higher version to interoperate with Cisco Unity Express 3.1.

Required Data for This Procedure

The following information is required to configure the JTAPI parameters:

- IP address or hostname for the primary, secondary, and tertiary Cisco Unified Communications Manager servers
- JTAPI user ID and password from Cisco Unified Communications Manager. The password is case sensitive. These values must match the JTAPI user ID and password that were configured on Cisco Unified Communications Manager.
- List of CTI ports

Note If you are using Cisco Unified Communications Manager 5.0 or a later version, verify that the AXL service is active. To do this, go to the Cisco Unified Communications Manager serviceability website, click on **Tools > Service Activation**. Look for Cisco AXL Web service.

SUMMARY STEPS

1. **config t**
2. **ccn subsystem jtapi**
3. **ccm-manager address** *{primary-server-ip-address | primary-server-hostname}* {*secondary-server-ip-address | secondary-server-hostname*} {*tertiary-server-ip-address | tertiary-server-hostname*}
4. **ccm-manager username** *jtapi-user-id* **password** *jtapi-user-password*
5. **ctiport** *cti-port-number*
6. **end**
7. **exit**
8. **show ccn subsystem jtapi**
9. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**config t**
	
	Example:
	`se-10-0-0-0# config t`
	`se-10-0-0-0(config)#`
		Enters configuration mode.

Step 2	**ccn subsystem jtapi**
	
	Example:
	`se-10-0-0-0(config)# ccn subsystem jtapi`
	`se-10-0-0-0(config-jtapi)#`
		Enters JTAPI configuration mode.

Step 3	**ccm-manager address** *{primary-server-ip-address	primary-server-hostname}* {*secondary-server-ip-address	secondary-server-hostname*} {*tertiary-server-ip-address	tertiary-server-hostname*}
				
				Example:
				`se-10-0-0-0(config-jtapi)# ccm-manager address 10.100.10.120`
				`se-10-0-0-0(config-jtapi)# ccm-manager address 10.100.10.120 10.120.10.120 10.130.10.120`
					Specifies up to three Cisco Unified Communications Manager servers. Enter the server IP addresses or hostnames on one command line or on separate command lines. If entered on separate lines, the servers are assigned in order as primary, secondary, and tertiary servers.
					
					Note Restart the system for these changes to be effective.

Step 4	**ccm-manager username** *jtapi-user-id* **password** *jtapi-user-password* **Example:** `se-10-0-0-0(config-jtapi)# ccm-manager username jtapiuser password myjtapi`	Specifies the JTAPI user ID and password. The password is case sensitive. These values must match the JTAPI user ID and password that were configured on Cisco Unified Communications Manager. **Note** Restart the system for these changes to be effective.

	Command or Action	Purpose

Step 5	**ctiport** *cti-port1 cti-port2 cti-port3 cti-port4...*
	
	Example:
	`se-10-0-0-0(config-jtapi)# ctiport 7008`
	`se-10-0-0-0(config-jtapi)# ctiport 7009`
	`se-10-0-0-0(config-jtapi)# ctiport 7010`
	`se-10-0-0-0(config-jtapi)# ctiport 7011`
	
	`se-10-0-0-0(config-jtapi)# ctiport 6001 6002 6003 6004 6005 6006 6007 6008`
		Specifies the JTAPI CTI ports that are configured on Cisco Unified Communications Manager and that are associated with the Cisco Unified Communications Manager JTAPI user.
		
		Repeat the command to enter more than one port number or enter the ports on one line. For AIM-CUE, specify up to 4 ports. For NM-CUE, specify up to 8 ports. For NM-CUE-EC, specify up to 16 ports.

Step 6	**end**
	
	Example:
	`se-10-0-0-0(config-jtapi)# end`
	`se-10-0-0-0(config)#`
		Exits JTAPI configuration mode.

Step 7	**exit**
	
	Example:
	`se-10-0-0-0(config)# exit`
	`se-10-0-0-0#`
		Exits configuration mode.

Step 8	**show ccn subsystem jtapi** **Example:** `se-10-0-0-0# show ccn subsystem jtapi`	Displays configured JTAPI parameters.
Step 9	**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

Following is example output of the **show ccn subsystem jtapi** command:

```
se-10-0-0-0# show ccn subsystem jtapi

Cisco Call Manager:                    10.100.10.120
CCM JTAPI Username:                    jtapiuser
CCM JTAPI Password:                    *****
Call Control Group 1 CTI ports:        7008,7009,7010,7011
```

Managing Scripts

Cisco Unity Express provides you with building blocks (known as Steps) through its Cisco Unity Express Editor Software, which can be used to create customized call-flows for various applications such as auto-attendant or IVR applications. These call flows can be saved as AEF files (known as scripts).

Cisco Unity Express ships with some internal scripts, which are known as system scripts. These system scripts cannot be downloaded, modified or deleted.

The NME-CUE, NM-CUE-EC, and NM-CUE support up to eight custom scripts; the AIM-CUE supports up to four custom scripts.

Customizing scripts involves the following procedures:

- Creating a Script File, page 51
- Uploading a Script File, page 51
- Displaying the List of Existing Scripts, page 52
- (Optional) Downloading a Script File, page 52
- (Optional) Deleting a Script File, page 52

Creating a Script File

To create a script file, use the Cisco Unity Express Editor software. See to the *Cisco Unity Express 3.1 Guide to Writing Auto-Attendant Scripts* for guidelines and procedures for creating a script file.

The file cannot be larger than 256 KB. Starting with Cisco Unity Express 3.1, script files can also be created using Editor Express. Editor Express can be accessed using the GUI option **System > Scripts > New**.

Note Cisco Unity Express Editor Express provides only a subset of the functionality that is available the Cisco Unity Express Script Editor. Use Cisco Unity Express Editor Express for simple call-flow customizations only.

After creating the script, use the GUI or Cisco Unity Express **ccn copy** command to upload the file to the Cisco Unity Express module. See the next section, "Uploading a Script File, page 51," for the upload procedure.

Note If you create your script using Cisco Unity Express Editor Express, you do not need to upload it as it is directly saved on the Cisco Unity Express module.

Uploading a Script File

After creating the AEF file, upload the file using the **ccn copy url** command in Cisco Unity Express EXEC mode:

ccn copy url ftp://*source-ip-address*/script-filename.aef script script-filename.aef [username username password password]

Example:

```
se-10-0-0-0# ccn copy url ftp://10.100.10.123/AVTscript.aef script AVTscript.aef
se-10-0-0-0# ccn copy url http://www.server.com/AVTscript.aef script AVTscript.aef
```

This command is equivalent to using the GUI option **Voice Mail > Scripts** and selecting **Upload**.

An error message appears if you try to upload more than the maximum number of scripts allowed on your Cisco Unity Express module.

Displaying the List of Existing Scripts

To displays details of the script files existing on the module, use the following command in Cisco Unity Express EXEC mode:

show ccn scripts

Example:

```
se-10-0-0-0# show ccn scripts

Name:                        setmwi.aef
Script type:                 aa
Create Date:                 Wed May 30 19:49:05 PDT 2007
Last Modified Date:          Wed May 30 19:49:05 PDT 2007
Length in Bytes:             27768

Name:                        xfermailbox.aef
Script type:                 aa
Create Date:                 Wed May 30 19:49:14 PDT 2007
Last Modified Date:          Wed May 30 19:49:14 PDT 2007
Length in Bytes:             7579

Name:                        aal.aef
Script type:                 aa
Create Date:                 Thu May 31 22:16:33 PDT 2007
Last Modified Date:          Thu May 31 22:16:33 PDT 2007
Length in Bytes:             10035
```

Downloading a Script File

Scripts can be copied from the auto-attendant and stored on another server or PC.

To download or copy a script file, use the **ccn copy script** command in Cisco Unity Express EXEC mode:

ccn copy script *script-filename* **url ftp://***destination-ip-address***/***script-filename*

Example:

```
se-10-0-0-0# ccn copy script AVTscript.aef url ftp://10.100.10.123/AVTscript.aef
```

Deleting a Script File

To delete an auto-attendant script file from Cisco Unity Express, use the **ccn delete** command in Cisco Unity Express EXEC mode:

ccn delete script *script-filename*

Example:

```
se-10-0-0-0# ccn delete script AVTscript.aef
```

```
Are you sure you want to delete this script? (y/n)
```

Managing Prompts

Cisco Unity Express supports customized greeting and prompt files. The CUE-NM-EC and CUE-NM support up to 120 prompts; the CUE-AIM supports up to 25 prompts.

Customizing prompts requires the following procedures:

- Recording a Greeting or Prompt File, page 53 (required)
- Uploading a Greeting or Prompt File, page 53 (required)
- Downloading a Greeting or Prompt File, page 54 (optional)
- Renaming a Greeting or Prompt File, page 55(optional)
- Deleting a Greeting or Prompt File, page 55 (optional)
- Re-recording a Greeting or Prompt File, page 55 (optional)

Recording a Greeting or Prompt File

Two methods are available to create greeting and prompt files:

- Create a wav file with the following format: G.711 u-law, 8 kHz, 8 bit, Mono. The file cannot be larger than 1 MB (about 2 minutes). After recording the wav file, use the GUI or Cisco Unity Express CLI **ccn copy url** command to copy or upload the file to the Cisco Unity Express module. See the next section, "Uploading a Greeting or Prompt File," for the upload procedure.
- Cisco Unity Express provides an in-built application called Administration via Telephone (AvT), which lets you record customized greeting and prompt files directly on the module using a telephone. For details on how to configure and use AvT, see the chapter Configuring the Administration via Telephone Application, page 133.

  We recommend using the AvT on the TUI to record greetings and prompts because the AvT provides higher sound quality compared to .wav files recorded using other methods.

Uploading a Greeting or Prompt File

After recording the .wav greeting or prompt file, upload the file using the **ccn copy url** command in Cisco Unity Express EXEC mode:

ccn copy url *source-ip-address* **prompt** *prompt-filename* [**language** *xx_YY*] [**username** *name* **password** *password*]

where *prompt-filename* is the file to be uploaded, *xx_YY* is the language of the prompt file, *name* is the FTP server login ID, and *password* is the FTP server password.

The optional language parameter lets you specify the language directory in which you want the prompt to be uploaded. An error message appears if the language specified in the command is not installed on the module. If the language parameter is omitted in this CLI command, the prompt is uploaded to the default system language directory.

Example:

```
se-10-0-0-0# ccn copy url ftp://10.100.10.123/AAprompt1.wav prompt AAprompt1.wav
se-10-0-0-0# ccn copy url http://www.server.com/AAgreeting.wav prompt AAgreeting.wav
```

This command is equivalent to using the GUI option **Voice Mail > Prompts** and selecting **Upload**.

An error message appears if you try to upload more than the maximum number of prompts allowed on your Cisco Unity Express module.

Displaying Existing Greeting or Prompt File lists

To display details of the greeting or prompt files existing on the module, use the following command in Cisco Unity Express EXEC mode:

show ccn prompts [language *xx_YY*]

The optional language parameter lets you specify the language directory from which the prompts will be listed. If the language parameter is omitted in this CLI command, then prompts from all language directories are listed.

Example:

```
se-10-0-0-0# show ccn prompts

Name: AAWelcome.wav
Language: en_US
Last Modified Date: Tue May 29 22:41:44 PDT 2007
Length in Bytes: 15860

Name: AABusinessClosed.wav
Language: en_US
Last Modified Date: Tue May 29 22:41:44 PDT 2007
Length in Bytes: 26038Name: AABusinessOpen.wavLanguage: en_USLast Modified Date: Tue May
29 22:41:44 PDT 2007Length in Bytes: 1638Name: AAHolidayPrompt.wavLanguage: en_USLast
Modified Date: Tue May 29 22:41:44 PDT 2007Length in Bytes: 24982
```

Downloading a Greeting or Prompt File

Greetings and prompts can be copied from the Cisco Unity Express module and stored on another server or PC.

To copy or download a greeting or prompt file, use the **ccn copy prompt** command in Cisco Unity Express EXEC mode:

ccn copy prompt *prompt-filename* **url** ftp:*//destination-ip-address/prompt-filename*
[**language** *xx_YY*] [**username** *name* **password** *password*]

where *prompt-filename* is the file to be downloaded, *destination-ip-address* is the IP address of the FTP server, *xx_YY* is the language directory from which the prompt file is to be downloaded, *name* is the FTP server login ID, and *password* is the FTP server password.

Example:

```
se-10-0-0-0# ccn copy prompt AAprompt2.wav url ftp://10.100.10.123/AAprompt2.wav
```

Renaming a Greeting or Prompt File

To rename a greeting or prompt file already existing on the Cisco Unity Express module, use the **ccn rename prompt** command in Cisco Unity Express EXEC mode:

ccn rename prompt *old-name new-name* [**language** *xx_YY*]

where *old-name* is the existing filename and *new-name* is the revised name, and *xx_YY* is the language directory in which the prompt to be renamed resides. If the language parameter is omitted in this CLI command, the system renames the prompt old-name from the default system language directory.

An error message appears if the prompt old-name does not exist in that language directory.

Example:

```
se-10-0-0-0# ccn rename prompt AAmyprompt.wav AAmyprompt2.wav
```

Deleting a Greeting or Prompt File

To delete a greeting or prompt file from the Cisco Unity Express module, use the **ccn delete** command in Cisco Unity Express EXEC mode:

ccn delete prompt *prompt-filename* [**language** *xx_YY*]

where *prompt-filename* is the file to be deleted, and *xx_YY* is the language directory from which the prompt is to be deleted. If the language parameter is omitted from this CLI command, the system attempts to delete this prompt from the default system language directory.

An error message appears if the prompt prompt-filename does not exist in that language directory.

Example:

```
se-10-0-0-0# ccn delete prompt AAgreeting.wav
```

Re-recording a Greeting or Prompt File

You can rerecord existing greeting and prompt files using the AvT application.

For details on how to rerecord prompts using AvT, see the chapter Configuring the Administration via Telephone Application, page 133.

Managing Applications

After you complete your pre-application tasks by uploading your scripts and prompts, you must create an application on the Cisco Unity Express module.

Cisco Unity Express supports two types of applications:

- Auto-Attendant Applications: This option is available with basic VoiceMail license.
- Interactive Voice Response (IVR) Applications: IVR license needs to be purchased and installed in order to create IVR applications.

Cisco Unity Express ships with some internal applications, which are known as system applications. These system applications cannot be deleted.

The maximum number of Auto-Attendant applications that can be created on Cisco Unity Express if four, regardless of the hardware type. The maximum number of IVR applications that can be created on NME-CUE, NM-CUE-EC and NM-CUE is eight; the maximum number of IVR applications that can be created on AIM-CUE is four.

This section describes the procedure for managing applications and contains the following sections:

Creating and Modifying Applications

Use the following procedure to create or modify an application.

Required Data for This Procedure

- Application name.
- Script name for the application.
- Maxsessions value. See the "Sharing Ports Among Applications and Triggers" section on page 70.
- Name and value for each parameter that the script requires. These may vary, depending on the script that you have created.

Note For more information about creating scripts, see the *Cisco Unity Express 3.1 Guide to Writing Scripts*.

SUMMARY STEPS

1. **config t**
2. **ccn application** *full-name* [**aa** | **ivr**]
3. **default [description | enabled | maxsessions | script | parameter name]**
4. **description "***text***"**
5. **maxsessions** *number*
6. **no [description | enabled | maxsessions | script | parameter name]**
7. **parameter** *name* **"***value***"**
8. **script** *name*
9. **enabled**
10. **end**
11. **exit**
12. **show ccn application** [**aa** | **ivr**]
13. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**config t** **Example:** `se-10-0-0-0# config t`	Enters configuration mode.
Step 2	**ccn application** *full-name* [**aa** | **ivr**] **Example:** `se-10-0-0-0(config)# ccn application myscript aa`	Specifies the application to configure and enters application configuration mode. The *full-name* argument specifies the name of the application to configure. The optional parameter **aa** specifies that the application being configured is an Auto-Attendant application. The optional parameter **ivr** specifies that the application being configured is an IVR application. The default application type (when no optional parameter is specified) is Auto-Attendant.
Step 3	**default** [description | enabled | maxsessions | script | parameter name] **Example:** `se-10-0-0-0(config-application)# default maxsessions`	(Optional) Resets the application configuration as follows: • **description**—Sets the description to the name of the application. • **enabled**—Enables the application. • **maxsessions**—Sets the maxsessions value to the number of licensed ports for that application type. • **script**—No effect. • **parameter name**—Uses the script's default value.
Step 4	**description "***text***"** **Example:** `se-10-0-0-0(config-application)# description "my application"`	(Optional) Enter a description of the application. Use quotes around the text.
Step 5	**maxsessions** *number* **Example:** `se-10-0-0-0(config-application)# maxsessions 5`	Specifies the number of callers who can access this application simultaneously.

	Command or Action	Purpose
Step 6	**no** [description | enabled | maxsessions | script | parameter name] **Example:** se-10-0-0-0(config-application)# **no description**	(Optional) Resets the application configuration as follows: • **description** - Removes the description for this application. • **enabled** - Disables the application. • **maxsessions** - Sets the maxsessions value to zero. • **script** - No effect. • **paramater name** - No effect.
Step 7	**parameter** *name* **"***value***"** **Example:** se-10-0-0-0(config-application)# **parameter MaxRetry "4"** se-10-0-0-0(config-application)# **parameter WelcomePrompt "Welcome.wav"**	Configures script parameters for the application. Each parameter must have a name and a value, which is written within quotes. For more details on Script Parameters, see the "Script Parameters for Applications" section on page 59.
Step 8	**script** *name* **Example:** se-10-0-0-0(config-application)# **script myscript.aef**	Specifies the name of the script that will be used by the application.
Step 9	**enabled** **Example:** se-10-0-0-0(config-application)# enabled	Allows the application to be accessible to the system.
Step 10	**end** **Example:** se-10-0-0-0(config-application)# end	Exits application configuration mode.
Step 11	**exit** **Example:** se-10-0-0-0(config)# exit	Exits configuration mode.
Step 12	**show ccn application** [**aa** | **ivr**] **Example:** se-10-0-0-0# **show ccn application ivr**	Displays details of the specified type of application. If no application type is specified, all applications on the system are displayed.
Step 13	**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

The following example illustrates the **show ccn application** output:

```
se-10-0-0-0# show ccn application
```

```
Name:                                myscript
Description:                         Application Type: aa
Script:                              myscript.aef
ID number:                           2
Enabled:                             yes
Maximum number of sessions:          5
MaxRetry:                             4
WelcomePrompt:                       Welcome.wav
se-10-0-0-0#
```

Script Parameters for Applications

While creating a script with Cisco Unity Express Script Editor, you can specify some script variables to be "parameters." The value of these "parameters" can be easily modified using the Cisco Unity Express configuration commands, without the need to edit the script using the Cisco Unity Express Script Editor. This has two benefits:

- You can deploy the same script at multiple locations and still customize the script flow to some extent for that particular location without needing different scripts for different locations. For example, you can create a simple script which welcomes the caller by playing a prompt such as "Welcome to ABC stores," and then transfers the caller to the operator. You can specify this welcome prompt and the operator extension as script paramaters while creating the script. Then you can deploy the same script at multiple locations and change the welcome prompt and operator extension by using the Cisco Unity Express configuration commands.
- You can create multiple applications using the same script, but with different values for the script parameters, thereby allowing you to provide a different experience to the caller depending on the application being invoked.

To view a list of script parameters, create an application using that script, and then use the **show ccn application** command to display the list of parameters and their default values.

To change the value of these parameters, see Step 7 of the "Creating and Modifying Applications" section on page 56.

Deleting an Application

If you have an application that you do not want to keep, use this procedure to delete the application and any triggers associated with that application.

After you delete the application and triggers, the script associated with the application still remains installed on Cisco Unity Express module.

The following system applications shipp with Cisco Unity Express, and cannot be deleted:

- autoattendant
- ciscomwiapplication
- msgnotification
- promptmgmt (the AvT application)
- voicemail

Required Data for This Procedure

The following information is required to delete an application:

- Application name
- All trigger numbers or URL names associated with the application

SUMMARY STEPS

1. **show ccn application**
2. **show ccn trigger**
3. **config t**
4. **no ccn trigger [sip | jtapi | http] phonenumber** *number*
5. **no ccn application** *name*
6. **exit**
7. **show ccn application**
8. show ccn trigger
9. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show ccn application** **Example:** `se-10-0-0-0# show ccn application`	Displays the currently configured applications. Look for the name of the application you want to delete.
Step 2	**show ccn trigger** **Example:** `se-10-0-0-0# show ccn trigger`	Displays the currently configured triggers. Look for the telephone numbers associated with the application you want to delete.
Step 3	**config t** **Example:** `se-10-0-0-0# config t`	Enters configuration mode.
Step 4	**no ccn trigger** [**sip** | **jtapi** | **http**] **phonenumber** *number* **Example:** `se-10-0-0-0(config)# no ccn trigger sip phonenumber 7200`	Deletes a trigger associated with this application. Repeat this command for each trigger associated with the application.
Step 5	**no ccn application** *name* **Example:** `se-10-0-0-0(config)# no ccn application autoattendant`	Deletes the application called name.

	Command or Action	Purpose
Step 6	**exit** **Example:** `se-10-0-0-0(config)# exit`	Exits configuration mode.
Step 7	**show ccn application** **Example:** `se-10-0-0-0# show ccn application`	Displays the currently configured applications. Confirm that the deleted application is not shown.
Step 8	**show ccn trigger** **Example:** `se-10-0-0-0# show ccn trigger`	Displays the triggers for each configured application. Confirm that the deleted triggers are not displayed.
Step 9	**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

The following is sample output from the **show ccn application** and **show ccn trigger** commands:

```
se-10-0-0-0# show ccn application

Name:                           voicemail
Description:                    voicemail
Script:                         voicebrowser.aef
ID number:                      1
Enabled:                        yes
Maximum number of sessions:     8
logoutUri:                      http://localhost/voicemail/vxmlscripts/mbxLogout.jsp
uri:                            http://localhost/voicemail/vxmlscripts/login.vxml

Name:                                   autoattendant
Description:                            autoattendant
Script:                                 aa.aef
ID number:                              2
Enabled:                                yes
Maximum number of sessions:             8
MaxRetry:                               3
operExtn:                               0
welcomePrompt:                          AAWelcome.wav
se-10-0-0-0#

Name:                                   myapplication
Description:                            My AA application
Script:                                 myscript.aef
ID number:                              3
Enabled:                                yes
Maximum number of sessions:             8
MaxRetry:                               3
operExtn:                               0
welcomePrompt:                          NewAAWelcome.wav
se-10-0-0-0#

se-10-0-0-0# show ccn trigger
```

```
Name:                          6500
Type:                          SIP
Application:                   voicemail
Locale:                        systemDefault
Idle Timeout:                  5000
Enabled:                       yes
Maximum number of sessions:    3

Name:                          6700
Type:                          SIP
Application:                   autoattendant
Locale:                        systemDefault
Idle Timeout:                  5000
Enabled:                       yes
Maximum number of sessions:    8

Name:                          7200
Type:                          SIP
Application:                   myapplication
Locale:                        systemDefault
Idle Timeout:                  5000
Enabled:                       yes
Maximum number of sessions:    8
se-10-0-0-0#
```

The following configuration deletes the auto-attendant application and its trigger:

```
se-10-0-0-0# config t
se-10-0-0-0(config)# no ccn trigger sip phonenumber 50170
se-10-0-0-0(config)# no ccn application myapplication
se-10-0-0-0(config)# exit
```

Now the output of the **show** commands looks similar to the following:

```
se-10-0-0-0# show ccn application

Name:                                voicemail
Description:                         voicemail
Script:                              voicebrowser.aef
ID number:                           1
Enabled:                             yes
Maximum number of sessions:          8
logoutUri:                           http://localhost/voicemail/vxmlscripts/m
bxLogout.jsp
uri:                                 http://localhost/voicemail/vxmlscripts/l
ogin.vxml
se-10-0-0-0#

Name:                                autoattendant
Description:                         autoattendant
Script:                              aa.aef
ID number:                           2
Enabled:                             yes
Maximum number of sessions:          8
MaxRetry:                            3
operExtn:                            0
welcomePrompt:                       AAWelcome.wav
se-10-0-0-0#

se-10-0-0-0# show ccn trigger

Name:                          6500
Type:                          SIP
```

```
Application:                    voicemail
Locale:                         systemDefault
Idle Timeout:                   5000
Enabled:                        yes
Maximum number of sessions:     3

Name:                           6700
Type:                           SIP
Application:                    autoattendant
Locale:                         systemDefault
Idle Timeout:                   5000
Enabled:                        yes
Maximum number of sessions:     8
se-10-0-0-0#
```

Managing Triggers

Triggers are incoming events that invoke application which in turn starts executing the script associated with that application. For example, the incoming event can be an incoming call or an incoming HTTP request.

After you have created and configured your application, you need to create a trigger on the Cisco Unity Express module to point to that application.

Cisco Unity Express supports three types of triggers:

- SIP triggers—Use this type of trigger to invoke applications in Cisco Unified CME and Cisco SRST mode. This type of trigger is identified by the phonenumber which is dialed to invoke the desired application.
- JTAPI triggers—Use this type of trigger to invoke applications in Cisco Unified Communications Manager mode. This type of trigger is identified by the phonenumber which is dialed to invoke the desired application.
- HTTP triggers—Use this type of trigger to invoke applications using an incoming HTTP request. Such a trigger is identified by the URL suffix of the incoming HTTP request. This type of trigger can only be used if an IVR license has been purchased and installed on the system.

Cisco Unity Express ships with some internal triggers, which are known as system triggers. These system triggers cannot be deleted.

This section describes the procedure for managing triggers and contains the following sections:

Configuring SIP Triggers for the Applications

Cisco Unity Express uses SIP to handle incoming calls in Cisco Unified CME and Cisco SRST mode. If you are deploying Cisco Unity Express in either of these modes, you must configure a SIP trigger for your application so that it can be invoked by incoming calls. This type of trigger is identified by the phone number which is dialed to invoke the desired application.

The telephone number that identifies your SIP trigger must match the dial-peer configured on the Cisco IOS SIP gateway. In order for Cisco Unity Express to be able to handle incoming calls on this phone number properly, you must configure the dial-peer on the Cisco IOS SIP gateway as follows:

```
se-10-0-0-0# config t
se-10-0-0-0(config)# dial-peer voice 6000 voip
se-10-0-0-0(config)# destination-pattern 6...
se-10-0-0-0(config)# session protocol sipv2
se-10-0-0-0(config)# session target ipv4:1.100.50.125
se-10-0-0-0(config)# dtmf-relay sip-notify
se-10-0-0-0(config)# codec g711ulaw
se-10-0-0-0(config)# no vad
```

Note Make sure that VAD is turned OFF on the dial-peer, it is configured to use g711ulaw codec and the session target is pointing to Cisco Unity Express module.

Cisco Unity Express supports a maximum of 8 SIP triggers for all applications combined, regardless of the hardware type.

Required Data for This Procedure

The following information is required to configure the SIP triggers for applications:

- Telephone number that invokes the application. The number must be different for different applications. The *number* value should match one of the patterns configured in the *destination-pattern* field of the SIP dial peer pointing to Cisco Unity Express.
- Maximum number of callers that can access the trigger simultaneously. See the section "Sharing Ports Among Applications and Triggers" on page 70 for guidelines on assigning this value.

SUMMARY STEPS

1. **config t**
2. **ccn trigger sip phonenumber** *number*
3. **application** *application-name*
4. **enabled**
5. **maxsessions** *number*
6. **locale** *xx_YY*
7. **end**
8. **exit**
9. **show ccn trigger**
10. **copy running-config startup-config**

DETAILED STEPS

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

Step 2	**ccn trigger sip phonenumber** *number*
	
	Example:
	`se-10-0-0-0(config)# ccn trigger sip phonenumber 50150`
	`se-10-0-0-0(config)# ccn trigger sip phonenumber 50160`
		Specifies the telephone number that acts as the trigger to start the application on the Cisco Unity Express module and enters trigger configuration mode.
		• *number*—The value should match one of the patterns configured in the *destination-pattern* field of the SIP dial peer pointing to Cisco Unity Express.

Step 3	**application** *application-name*
	
	Example:
	`se-10-0-0-0(config-trigger)# application voicemail`
	`se-10-0-0-0(config-trigger)# application autoattendant`
	`se-10-0-0-0(config-trigger)# application promptmgmt`
		Specifies the name of the application to invoke when a call is made to the trigger phone number.

Step 4	**enabled** **Example:** `se-10-0-0-0(config-trigger)# enabled`	Enables the trigger.

Step 5	**maxsessions** *number*
	
	Example:
	`se-10-0-0-0(config-trigger)# maxsessions 3`
	`se-10-0-0-0(config-trigger)# maxsessions 6`
		Specifies the maximum number of callers that this application can handle simultaneously. See the "Sharing Ports Among Applications and Triggers" section on page 70 for guidelines on assigning this value.

Step 6	**locale** *xx_YY* **Example:** `se-10-0-0-0(config-trigger)#  locale en_US`	(Optional) Specifies the trigger language. Any prompts being played out by an application invoked by this trigger will be played out in this language. Use this configuration only if you have more than one language installed on the system. The default for this configuration is to use the system default language as the trigger language.
Step 7	**end** **Example:** `se-10-0-0-0(config-trigger)# end`	Exits trigger configuration mode.
Step 8	**exit** **Example:** `se-10-0-0-0(config)# exit`	Exits configuration mode.

	Command or Action	Purpose
Step 9	**show ccn trigger** **Example:** `se-10-0-0-0# show ccn trigger`	Displays the details of all configured triggers.
Step 10	**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

The following sample configuration sets two triggers on the Cisco Unity Express module:

```
se-10-0-0-0# config t
se-10-0-0-0(config)# ccn trigger sip phonenumber 50150
se-10-0-0-0(config-trigger)# application voicemail
se-10-0-0-0(config-trigger)# maxsessions 4
se-10-0-0-0(config-trigger)# enabled
se-10-0-0-0(config-trigger)# end
se-10-0-0-0(config)#
se-10-0-0-0(config)# ccn trigger sip phonenumber 50160
se-10-0-0-0(config-trigger)# application autoattendant
se-10-0-0-0(config-trigger)# maxsessions 3
se-10-0-0-0(config-trigger)# enabled
se-10-0-0-0(config-trigger)# end
se-10-0-0-0#
```

The output of **show ccn trigger** looks similar to the following:

```
se-10-0-0-0# show ccn trigger

Name:                          50150
Type:                          SIP
Application:                   voicemail
Locale:                        systemDefault
Idle Timeout: 10000
Enabled:                       yes
Maximum number of sessions:    4

Name:                          50160
Type:                          SIP
Application:                   autoattendant
Locale:                        systemDefault
Idle Timeout: 10000
Enabled:                       yes
Maximum number of sessions:    3
se-10-0-0-0#
```

Configuring JTAPI Triggers for the Applications (Cisco Unified Communications Manager Only)

Cisco Unity Express uses JTAPI to handle incoming calls in Cisco Unified Communications Manager mode. If you are deploying Cisco Unity Express in Cisco Unified Communications Manager mode, you must configure a JTAPI trigger for your application so that it can be invoked by incoming calls. This type of trigger is identified by the phone number which is dialed to invoke the desired application.

The telephone number that identifies your JTAPI trigger must match the Route Point configured on the Cisco Unified Communications Manager.

Note This Route Point must be associated with the JTAPI user configured on Cisco Unified Communications Manager. This same JTAPI user must also be configured on Cisco Unity Express module. See the "Configuring Triggers" section on page 111 for details on JTAPI user configuration.

Cisco Unity Express supports a maximum of 8 JTAPI triggers for all applications combined, regardless of the hardware type.

This configuration is required for only for Cisco Unified Communications Manager mode.

Required Data for This Procedure

The following information is required to configure the JTAPI triggers for applications:

- Telephone number that invokes the application. The number must be different for different for applications.
- Number of seconds the system must wait for a caller response before it times out and drops the call.
- Language to use for the prompts. Cisco Unity Express supports several languages. Only one can be installed on the system. See the *Release Notes for Cisco Unity Express 3.1* for a list of available languages.
- Maximum number of callers that can access the trigger simultaneously. See the "Sharing Ports Among Applications and Triggers" section on page 70 for guidelines on assigning this value.

SUMMARY STEPS

1. **config t**
2. **ccn trigger jtapi phonenumber** *number*
3. **application** *application-name*
4. **enabled**
5. **maxsessions** *number*
6. **locale** *xx_YY*
7. **end**
8. **exit**
9. **show ccn trigger**
10. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**config t** **Example:** `se-10-0-0-0# config t`	Enters configuration mode.
Step 2	**ccn trigger jtapi phonenumber** *number* **Example:** `se-10-0-0-0(config)# ccn trigger jtapi phonenumber 6700`	Specifies the telephone number that acts as the trigger to start the application on Cisco Unity Express and enters trigger configuration mode. The *number* value must match a JTAPI route point configured on Cisco Unified Communications Manager.
Step 3	**application** *application-name* **Example:** `se-10-0-0-0(config-trigger)# application promptmgmt`	Specifies the name of the application to invoke when a call is made to the trigger phone number.
Step 4	**enabled** **Example:** `se-10-0-0-0(config-trigger)# enabled`	Enables the trigger.
Step 5	**maxsessions** *number* **Example:** `se-10-0-0-0(config-trigger)# maxsessions 3`	Specifies the maximum number of callers that this trigger can handle simultaneously. See the "Sharing Ports Among Applications and Triggers" section on page 70 for guidelines on assigning this value.
Step 6	**locale** *xx_YY* **Example:** `se-10-0-0-0(config-trigger)#  locale en_US`	(Optional) Specifies the trigger language. Any prompts being played out by an application invoked by this trigger will be played out in this language. Use this configuration only if you have more than one language installed on the system. The default for this configuration is to use the system default language as the trigger language.
Step 7	**end** **Example:** `se-10-0-0-0(config-trigger)# end`	Exits trigger configuration mode.
Step 8	**exit** **Example:** `se-10-0-0-0(config)# exit`	Exits configuration mode.

	Command or Action	Purpose
Step 9	**show ccn trigger** **Example:** `se-10-0-0-0# show ccn trigger`	Displays the details of all configured triggers.
Step 10	**copy running-config startup-config** **Example:** `se-10-0-0-0# copy running-config startup-config`	Copies the configuration change to the startup configuration.

Examples

The following sample configuration sets two triggers on the Cisco Unity Express module:

```
se-10-0-0-0# config t
se-10-0-0-0(config)# ccn trigger jtapi phonenumber 6500
se-10-0-0-0(config-trigger)# application voicemail
se-10-0-0-0(config-trigger)# maxsessions 4
se-10-0-0-0(config-trigger)# enabled
se-10-0-0-0(config-trigger)# end
se-10-0-0-0(config)#
se-10-0-0-0(config)# ccn trigger jtapi phonenumber 6700
se-10-0-0-0(config-trigger)# application autoattendant
se-10-0-0-0(config-trigger)# maxsessions 8
se-10-0-0-0(config-trigger)# enabled
se-10-0-0-0(config-trigger)# end
se-10-0-0-0(config)# exit
se-10-0-0-0#
```

Output of the **show ccn trigger** command looks similar to the following:

```
se-10-0-0-0# show ccn trigger

Name:                          6500
Type:                          JTAPI
Application:                   voicemail
Locale:                        systemDefault
Idle Timeout:                  10000
Enabled:                       yes
Maximum number of sessions:    4

Name:                          6700
Type:                          JTAPI
Application:                   autoattendant
Locale:                        systemDefault
Idle Timeout:                  10000
Enabled:                       yes
Maximum number of sessions:    8
se-10-0-0-0#
```

Configuring HTTP Triggers for the Applications

Cisco Unity Express can accept incoming HTTP requests to invoke an application using an HTTP trigger. For example, you can use it to initiate an IVR application notifying customers that their order has been filled and shipped. This type of trigger is identified by the URL suffix of the incoming HTTP request.

This type of trigger can only be used if an IVR license has been purchased and installed on the system.

For details on how to configure and use HTTP triggers, see the *Cisco Unity Express Interactive Voice Response CLI Administrator Guide*.

Configuring Multiple Triggers for an Application

Your network may require multiple triggers for one or more Cisco Unity Express applications. For example, the following are some scenarios where multiple triggers for the same application are useful:

- Multiple language support—You have an auto-attendant application which you want to deploy in two different languages. One way to achieve this would be to have two different triggers (call-in numbers) pointing to the same application, but with different values for the **locale** parameter.

  For example, assume that you have call-in numbers 6700 and 6900 (both pointing to the same auto-attendant application), the locale for the trigger 6700 is configured to be *xx_XX,* and the locale for the trigger 6900 is configured to be *yy_YY.* If the callers dial 6700, they will hear the auto-attendant greetings in the language *xx_XX*. If the callers dial 6900, they will hear the auto-attendant greetings in the language *yy_YY.*

- Different call treatment for internal and external callers—You have an auto-attendant application, and you want to provide slightly different Menu options for internal and external callers. In other words, you want to provide an option to the internal callers to transfer to the inventory department, but you do not want to present this option to the external callers. One way to achieve this would be to have two different triggers (call-in numbers) pointing to the same application, and by making a branching decision in your script by checking the called number using the "Get Call Contact Info" step.

Repeat the procedure described in the "Configuring SIP Triggers for the Applications" section on page 64 and the "Configuring JTAPI Triggers for the Applications (Cisco Unified Communications Manager Only)" section on page 67 (depending on your deployment mode) to create multiple triggers for an application.

Sharing Ports Among Applications and Triggers

Accessing an Application

The maximum number of callers that can access an application concurrently is determined by two parameters:

- The maxsessions value configured for the triggers invoking the application.
- The maxsessions value configured for the application itself.

If more calls than the trigger's configured maxsession value are received, callers hear a busy tone.

If more calls than the application's configured maxsession value are received, Cisco Unity Express plays an error prompt to the callers.

The following example shows how the maxsessions values for applications and triggers play a role in how many active calls can be made to an application. In this example:

- Your module has 8 ports.
- You assigned the auto-attendant application a maxsessions value of 5.
- You configured 2 triggers both invoking the same auto-attendant application.

- You configured one trigger with a maxsessions value of 2 and the other trigger with a maxsessions value of 4.

The maximum number of callers that can access the auto-attendant application simultaneously is five, not six. This is because although your system has a total of six sessions available for the two triggers, they both are accessing the same application, which allows only five concurrent sessions. The maxsessions value of the application acts as the gating factor in this case.

On the other hand, suppose you configure both triggers with a maxsessions value of **2**. Now, the maximum number of concurrent calls to the application is four, not five. This is because the system has a total of only four ports assigned to the two triggers. The maxsessions value assigned to the triggers acts as the factor in this case.

Sharing Ports Among Different Applications

Cisco Unity Express supports multiple voice applications, and each of these applications need voice ports in order to execute. Consider the expected call traffic for each application when assigning the maxsessions for them. One application may have a higher call volume and therefore need more sessions than another, and at the same time you may want each application to have at least one session available for incoming calls. You should distribute the ports to your applications keeping in mind the usage of each application.

For example, your module has four ports and you configure the voice-mail application to have four maxsessions, and the auto-attendant application also to have four maxsessions. If four callers access voice-mail simultaneously, no ports will be available for auto-attendant callers. Only when zero, one, two, or three callers access voice-mail simultaneously will at least one port be available for auto-attendant.

As another examle, you configure the voice-mail auto-attendant applications to have three maxsessions. At no time will one application use up all the ports. If voice-mail has three active calls, one caller can access auto-attendant. A second call to either voice-mail or auto-attendant will not be successful.

Configuring Holiday Lists

Cisco Unity Express permits configuration of holiday lists that can be used by an application to play a customizable greeting to callers when the company is closed for a holiday. The following sections describe how to configure and use Cisco Unity Express holiday lists:

Overview of Holidays

You can configure:

- Year-specific holidays
- Fixed holidays

Year-Specific Holidays

- Cisco Unity Express supports up to three year-specific holiday lists for: the previous year, the current year, and the next year. If a year has no configured entries, the system handles that year as having no year-specific holidays.

  For example, if the current year is 2005 and you have not configured entries for 2006 (the next year), the system handles 2006 as having zero (0) holidays. You may configure holidays for 2005 and 2006 (the next year) but not for 2007.
- Each year-specific list can contain a maximum of 26 holidays.
- By default, all three year-specific holiday lists are empty.
- The administrator can delete entries from a previous year list but cannot add or modify that list in any other way.
- The system automatically deletes the previous year list at the beginning of the new calendar year.
- For example, the system will delete the 2004 holiday list on January 1, 2006.
- To copy holidays from one year to the next, use the GUI option Copy all to next year under **System > Holiday Settings**.

Fixed Holidays

- Fixed holidays are permanent holidays which apply to all years and do not need to be re-configured year after year (unlike year-specific holidays). If a holiday falls on the same date every year, those may be configured as fixed holidays.

  For example, if your business is always closed on January 1st for New Year celebrations, then you may configure January 1st as a fixed holiday.
- A maximum of 10 fixed holidays can be configured on the system.
- By default, there are no fixed holidays configured on the system.
- Fixed holidays may overlap with year-specific holidays. If you create a year-specific holiday
- that overlaps with a fixed holiday, a warning is issued. However, no warning is issued if you try to create a fixed holiday that overlaps with a year-specific holiday.

To configure holiday lists, use the graphical user interface (GUI) System > Holiday Settings option or the command-line interface (CLI) commands described in this section.

Using the Holiday Lists

The Cisco Unity Express Editor provides a step “Is Holiday” that checks the holidays configured on the system to determine whether the specified date is a holiday or not. The step takes as input the date to check against the holiday list. See the *Cisco Unity Express 3.1 Guide to Writing Scripts* for more information on steps.

For example, you can use the “Is Holiday” step in your script to check if the current day is a holiday. If it is a holiday, you can play a customized greeting to the caller, such as “We are closed today. If this is an emergency, please call 1-222-555-0150 for assistance. Otherwise, please call back later.”

Configuring Holiday Lists

Configuring Year-Specific Holiday Lists

Use the following command in Cisco Unity Express configuration mode to configure a year-specific holiday list:

calendar holiday date *yyyy mm dd* [**description** *holiday-description*]

where *yyyy* is the 4-digit year, *mm* is the 2-digit month, *dd* is the 2-digit day, and *holiday-description* is an optional description of the holiday. If the description is more than one word, enclose the text in quotes (“ ”).

The valid values for *yyyy* are the current year or the next year. An error message appears if the year or date is out of range.

Example:

```
se-10-0-0-0# config t
se-10-0-0-0(config)# calendar holiday date 2005 05 30 description “Memorial Day”
se-10-0-0-0(config)# exit
se-10-0-0-0#
```

Configuring the Fixed Holiday List

Use the following command in Cisco Unity Express configuration mode to configure a fixed holiday:

calendar holiday fixed *month day* **[description** *holiday-description***]**

where *month* is the 2-digit month, *day* is the 2-digit day, and *holiday-description* is an optional description of the holiday. If the description is more than one word, enclose the text in quotes (“ ”).

Example:

```
se-10-0-0-0# config t
se-10-0-0-0(config)# calendar holiday fixed 07 04 description “Independence Day”
se-10-0-0-0(config)# exit
se-10-0-0-0#
```

Displaying the Holiday List

Several CLI commands are available in Cisco Unity Express EXEC mode for displaying the holiday lists.

Displaying All Holiday Lists

The following command displays all the holiday lists configured on the system:

show calendar holiday [**all**]

This command displays the date and description for all holidays for all years. This display includes both year-specific holidays and fixed holidays. The output of this command appears similar to the following:

```
se-10-0-0-0# show calendar holiday

*******************************
           Year: 2004
*******************************
September 04    Labor Day
November  25    Thanksgiving

*******************************
           Year: 2005
*******************************
July      04    July 4th
September 05    Labor Day
November  24    Thanksgiving
December  25    Christmas
```

Displaying Holiday Lists for a Specific Year

The following command displays the holidays configured for a specific year:

show calendar holiday year *yyyy*

where *yyyy* is the 4-digit year. This command displays the date and description for all holidays configured for the specified year. This display includes both year-specific holidays and fixed holidays. If no holidays are configured for that year and the fixed holiday list is empty, the message "No holidays found for the specified year" appears. The output of this command appears similar to the following:

```
se-10-0-0-0-0# show calendar holiday year 2005

*******************************
           Year: 2005
*******************************
July      04    July 4th
September 05    Labor Day
November  24    Thanksgiving
December  25    Christmas
```

Displaying Holiday Lists for a Specific Month

The following command displays the holidays configured for a specific month in a specified year:

show calendar holiday year *yyyy* **month** *mm*

where *yyyy* is the 4-digit year and *mm* is the 2-digit month. This command displays the date and description for all holidays configured for the specified month in the specified year.This display includes both year-specific holidays and fixed holidays. If no holidays are configured for that month and there are no holidays in that month, the message "No holidays found for the specified month" appears.

The output of this command appears similar to the following:

```
se-10-0-0-0# show calendar holiday year 2005 month 12

*******************************
           Year: 2005
*******************************
December  25    Christmas
```

Deleting Holidays from the List

Several CLI commands are available in Cisco Unity Express configuration mode for deleting holidays from the list.

Deleting a Year-Specific Holiday from the Holiday List

The following command deletes a year-specific holiday:

Caution Use this command with caution because this operation is irreversible. Do not press the "Enter" key after the year; doing so deletes the holiday list for the entire year.

no calendar holiday date *yyyy mm dd*

where *yyyy* is the 4-digit year, *mm* is the 2-digit month, and *dd* is the 2-digit day.

Example:

```
se-10-0-0-0# config t
se-10-0-0-0(config)# no calendar holiday date 2004 11 25
se-10-0-0-0(config)# end
```

Deleting Year-Specific Holidays from a Specific Month

Caution Use this command with caution because this operation is irreversible and may cause the loss of the temporary holiday configuration for the entire month.

The following command deletes the year-specific holidays configured for a specific month in the specified year:

no calendar holiday year *yyyy* **month** *mm*

where *yyyy* is the 4-digit year and *mm* is the 2-digit month.

Example:

```
se-10-0-0-0# config t
se-10-0-0-0(config)# no calendar holiday year 2004 month 09
se-10-0-0-0(config)# end
```

Deleting Year-Specific Holidays for a Specific Year

Caution Use this command with caution because this operation is irreversible and may cause the loss of the holiday configuration for the entire year.

The following command deletes all the year-specific holidays configured for the specified year:

no calendar holiday year *yyyy*

where *yyyy* is the 4-digit year.

Example:

```
se-10-0-0-0# config t
se-10-0-0-0(config)# no calendar holiday year 2004
se-10-0-0-0(config)# end
```

Deleting a Fixed Holiday from the Holiday List

The following command deletes a fixed holiday:

no calendar holiday fixed *month day*

where *month* is the 2-digit month and *day* is the 2-digit day.

Example:

```
se-10-0-0-0# config t
se-10-0-0-0(config)# no calendar holiday fixed 07 04
se-10-0-0-0(config)# exit
```

Configuring Business Hours

Cisco Unity Express provides support for business hour schedules that specify the hours when the business is open or closed during the week.

The following sections describe this feature, its configuration, and the procedures for using it:

Overview of Business-Hours Schedules

You can configure up to 4 weekly business-hours schedules. Each day is divided into 48 half-hour time slots. Each of these time slots can be configured to specify whether the business is open or closed during that time. Use the graphical user interface (GUI) **System > Business Hours Settings** option or the command-line interface (CLI) commands described in this section to configure these slots.

The Cisco Unity Express system ships with one default schedule called “systemschedule.” This schedule indicates the business is open 24 hours per day, 7 days per week. Use the GUI **System > Business Hours Settings** option or CLI commands to modify or delete this default schedule. This schedule counts towards the maximum limit of 4 business-hours schedules.

Using the Business-Hours Schedule

The Cisco Unity Express Editor provides a step "Business Hours" that checks whether the business is open or closed during a specified time slot. The step takes three parameters as input: a date, time and the name of a schedule configured on the system. See the *Cisco Unity Express 3.1 Guide to Writing Scripts* for more information about steps.

For example, you can use the "Business Hours" step in your script to check whether the business is currently open or not. If it is closed, you can play a customized greeting to the caller, such as "You have reached us during our off-hours. If this is an emergency, please call 1-222-555-0150 for assistance. Otherwise, please call back later."

Creating a Business-Hours Schedule

Follow this procedure to create a business-hours schedule.

Data Required for This Procedure

The following information is required to configure a business-hours schedule:

- Schedule name

  The maximum length of the name is 31 alphanumeric characters, including uppercase letters A through Z, lowercase letters a through z, digits 0 through 9, underscore (_), and dash (-). The first character of the name must be a letter.

  If a schedule with this name does not exist, the system will create it. By default, a newly created schedule is open, 24 hours per day, 7 days per week.

  If the schedule already exists, any changes will modify the schedule.
- Day of the week
- Starting and ending clock times when the business is open and closed

SUMMARY STEPS

1. **config t**
2. **calendar biz-schedule** *schedule-name*
3. **closed day** *day-of-week* **from** *hh:mm* **to** *hh:mm*
4. **open day** *day-of-week* **from** *hh:mm* **to** *hh:mm*
5. **end**
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**config t** **Example:** `se-10-0-0-0# config t`	Enters configuration mode.
Step 2	**calendar biz-schedule** *schedule-name* **Example:** `se-10-0-0-0(config)# calendar biz-schedule normal_hours`	Specifies the name for the business-hours schedule and enters business configuration mode. The name must be one word. If a schedule with this name does not exist, the system creates it. If the schedule already exists, any changes modify the schedule. If the maximum number of schedules exists, the system displays an error message.

Step 3	**closed day** *day-of-week* **from** *hh*:*mm* **to** *hh*:*mm*
	
	Example:
	`se-10-0-0-0(config-business)# closed day 2 from 00:00 to 08:30`
	`se-10-0-0-0(config-business)# closed day 2 from 17:30 to 24:00`
		Enter the day of the week and the times when the business is closed for that day. Valid values for *day-of-week* are 1 to 7, where 1 represents Sunday, 2 is Monday, 3 is Tuesday, 4 is Wednesday, 5 is Thursday, 6 is Friday, and 7 is Saturday. Use the 24-hour clock format for *hh*. Valid *mm* values are 00 and 30 only.

Step 4	**open day** *day-of-week* **from** *hh*:*mm* **to** *hh*:*mm* **Example:** `se-10-0-0-0(config-business)# open day 2 from 08:30 to 17:30`	Enter the day of the week and the times when the business is open for that day. Valid values for *day-of-week* are 1 to 7, where 1 represents Sunday, and so on. Use the 24-hour clock format for *hh*. Valid *mm* values are 00 and 30 only.
Step 5	Repeat Steps 3 and 4 for each day of the week that needs business hours scheduled.	—

Step 6	**end**
	
	Example:
	`se-10-0-0-0(config-business)# end`
	`se-10-0-0-0(config)#`
		Exits business configuration mode.

Step 7	**exit**
	
	Example:
	`se-10-0-0-0(config)# exit`
	`se-10-0-0-0#`
		Exits configuration mode.

Examples

The following example configures a new business-hours schedule:

```
se-10-0-0-0# config t
se-10-0-0-0(config)# calendar biz-schedule normal
Adding new schedule
se-10-0-0-0(config-business)# closed day 1 from 00:00 to 24:00
se-10-0-0-0(config-business)# closed day 2 from 00:00 to 08:30
se-10-0-0-0(config-business)# closed day 2 from 17:30 to 24:00
se-10-0-0-0(config-business)# closed day 3 from 00:00 to 08:30
se-10-0-0-0(config-business)# closed day 3 from 17:30 to 24:00
se-10-0-0-0(config-business)# closed day 4 from 00:00 to 08:30
se-10-0-0-0(config-business)# closed day 4 from 17:30 to 24:00
se-10-0-0-0(config-business)# closed day 5 from 00:00 to 08:30
se-10-0-0-0(config-business)# closed day 5 from 20:00 to 24:00
se-10-0-0-0(config-business)# closed day 6 from 00:00 to 08:30
se-10-0-0-0(config-business)# closed day 6 from 18:00 to 24:00
se-10-0-0-0(config-business)# closed day 7 from 00:00 to 09:00
se-10-0-0-0(config-business)# closed day 7 from 13:00 to 24:00
se-10-0-0-0(config-business)# end
se-10-0-0-0(config)# exit
```

Modifying Business-Hours Schedules

In Cisco Unity Express configuration mode, use the following command to access a business-hours schedule for modification:

calendar biz-schedule *schedule-name*

where *schedule-name* is the name of the business-hours schedule to modify. If a schedule with the specified business name does not exist, the system creates it.

The following example accesses the existing "normal" business-hours schedule:

```
se-10-0-0-0# config t
se-10-0-0-0(config)# calendar biz-schedule normal
Modifying existing schedule
se-10-0-0-0(config-business)# open day 1 from 09:00 to 12:00
se-10-0-0-0(config-business)# end
se-10-0-0-0(config)# exit
se-10-0-0-0#
```

Only the hours specified using these commands are affected. The other time slots in the business-hours schedule are not modified.

Changing the Status of Open or Closed Hours

To modify an existing schedule, specify the open and closed hours for each day as needed.

Changing an Open Slot to a Closed Slot

Use either of the following configuration mode commands to change an open slot to a closed slot:

no open day *day-of-week* **from** *hh*:*mm* **to** *hh*:*mm*

closed day *day-of-week* **from** *hh*:*mm* **to** *hh*:*mm*

where *day-of-week* is the numeric day of the week (1 equals Sunday), *hh* are hours in the 24-hour clock format, and *mm* are minutes, either 00 or 30.

For example, if Monday is open from 09:00 to 17:00, then **no open day 2 from 09:00 to 10:00** or **closed day 2 from 09:00 to10:00** closes Monday 9:00 a.m. to 10:00 a.m.

Changing a Closed Slot to an Open Slot

Use either of the following commands to change a closed slot to an open slot:

no closed day *day-of-week* **from** *hh*:*mm* **to** *hh*:*mm*

open day *day-of-week* **from** *hh*:*mm* **to** *hh*:*mm*

where *day-of-week* is the numeric day of the week (1 equals Sunday), *hh* are hours in the 24-hour clock format, and *mm* are minutes, either 00 or 30.

For example, if Monday is closed from 00:00 to 10:00, then **no closed day 2 from 09:00 to 10:00** or **open day 2 from 09:00 to 10:00** opens the Monday time slot 9:00 a.m. to 10:00 a.m.

Examples

The following output shows the "normal" business-hours schedule:

```
se-10-0-0-0# show calendar biz-schedule normal

******************************
Schedule: normal
Day                 Open Hours
------------------------------
Sunday              None
Monday              08:30 to 17:30
Tuesday             08:30 to 17:30
Wednesday           08:30 to 17:30
Thursday            08:30 to 20:00
Friday              08:30 to 18:00
Saturday            09:00 to 13:00
```

The following commands modify the "normal" business hours by closing Monday hours from 8:30 to 9:30 and opening Saturday hours from 1:00 p.m. to 2:00 p.m.:

```
se-10-0-0-0# config t
se-10-0-0-0(config)# calendar biz-schedule normal
se-10-0-0-0(config-business)# no open day 2 from 08:30 to 09:30
se-10-0-0-0(config-business)# no closed day 7 from 13:00 to 14:00
se-10-0-0-0(config-business)# end
se-10-0-0-0(config)# exit
```

The following output shows the changed schedule:

```
se-10-0-0-0# show calendar biz-schedule normal

******************************
Schedule: normal
Day                 Open Hours
------------------------------
Sunday              None
Monday              09:30 to 17:30
Tuesday             08:30 to 17:30
Wednesday           08:30 to 17:30
Thursday            08:30 to 20:00
Friday              08:30 to 18:00
```

```
Saturday            09:00 to 14:00
```

Displaying Business-Hours Schedules

Several CLI commands are available in Cisco Unity Express EXEC mode for displaying the business-hours schedules.

Displaying a Specific Schedule

The following command displays a specific business-hours schedule:

show calendar biz-schedule *schedule-name*

where *schedule-name* is the name of the schedule. This command displays each day of the week and the open hours. The output of this command appears similar to the following.

```
se-10-0-0-0# show calendar biz-schedule normal

*****************************
Schedule: normal
Day                 Open Hours
-------------------------------
Sunday              None
Monday              08:30 to 17:30
Tuesday             08:30 to 17:30
Wednesday           08:30 to 17:30
Thursday            08:30 to 20:00
Friday              08:30 to 18:00
Saturday            09:00 to 13:00
```

Displaying All Businesses Schedules

The following command displays all the configured business-hours schedules in the system:

show calendar biz-schedule [**all**]

This command displays the open hours for each day of the week for each schedule. The output of this command appears similar to the following:

```
sse-10-0-0-0# show calendar biz-schedule

*****************************
Schedule: systemschedule
Day                 Open Hours
-------------------------------
Sunday              Open all day
Monday              Open all day
Tuesday             Open all day
Wednesday           Open all day
Thursday            Open all day
Friday              Open all day
Saturday            Open all day

*****************************
Schedule: normal
Day                 Open Hours
-------------------------------
Sunday              None
```

```
Monday              08:30 to 17:30
Tuesday             08:30 to 17:30
Wednesday           08:30 to 17:30
Thursday            08:30 to 20:00
Friday              08:30 to 18:00
Saturday            09:00 to 13:00

*****************************
Schedule: holiday-season
Day                 Open Hours
-----------------------------
Sunday              09:00 to 15:00
Monday              08:30 to 17:30
Tuesday             08:30 to 17:30
Wednesday           08:30 to 17:30
Thursday            08:00 to 21:00
Friday              08:00 to 21:00
Saturday            08:00 to 21:30
```

Deleting a Business-Hours Schedule

The following configuration mode command deletes a specified business-hours schedule:

no calendar biz-schedule *schedule-name*

where *schedule-name* is the name of the business-hours schedule to delete.

If you delete a business-hours schedule which is being used in the "B6usiness Hours" step in an application, the step assumes that the business is open 24 hours a day, 7 days a week.

The following example deletes the "normal" business-hours schedule:

```
se-10-0-0-0# config t
se-10-0-0-0(config)# no calendar biz-schedule normal
se-10-0-0-0(config)# exit
se-10-0-0-0#
```

Configuring System-Wide Fax Parameters

Version 3.1 extends its convergence feature set to include fax support. It allows both inbound and outbound faxing. Outbound faxing enables faxes to be printed to the fax machine.

This functionality requires T.37 fax support from the Cisco IOS gateways. Third-party fax servers are not supported.

After you complete the appropriate prerequisites (see below), you can configure the system level fax parameters as described below. This procedure also includes enabling a mailbox to receive faxes from a fax gateway.

In order to send and receive a fax on Cisco Unity Express, you need to configure inbound and outbound fax gateways. Inbound gateway is used for receiving fax, and outbound gateway is used for sending or printing fax. You can use the same Cisco IOS gateway for both inbound and outbound faxing. Also, in order to print a fax received by Cisco Unity Express, the phone number of a fax machine must be configured.

Prerequisites

You must configure Cisco IOS gateway for T.37 on-ramp and off-ramp fax support. See the chapter Configuring Your Cisco IOS Gateway for T.37 On-Ramp and Off-Ramp Fax Support, page 359 for more details.

If you want to restrict specified extensions from using this feature, you must configure a restriction table as described in the "Configuring Restriction Tables" section on page 279.

Required Data for This Procedure

This procedure requires:

- IP address or hostname of the outbound fax gateway
- IP address or hostname for the inbound fax gateway
- Fax number used to print faxes

SUMMARY STEPS

1. **config t**
2. **fax gateway outbound address** {*hostname* | *ip-address*}
3. **fax gateway inbound address** {*hostname* | *ip-address*}
4. **fax print** *E.164*
5. **exit**
6. **voice mailbox owner** *name*
7. **enable fax**
8. **end**
9. **show fax configuration**

DETAILED STEPS

	Command or Action	Purpose

Step 1	`config t`
	
	Example:
	`se-10-0-0-0# config t`
		Enters configuration mode.

Step 2	**fax gateway outbound address** {*hostname* | *ip-address*} **Example:** se-10-0-0-0(config)# fax gateway outbound address 172.21.21.40	Configures an outbound fax gateway (also known as Off-ramp). The fax subsystem uses this outbound gateway to send faxes.

	Command or Action	Purpose
Step 3	**fax gateway inbound address {***hostname* | *ip-address***}** **Example:** se-10-0-0-0(config)# fax gateway inbound address 172.21.21.40	Configures an inbound fax gateway (also known as On-ramp). The fax subsystem uses this inbound gateway to receive faxes. The system will reject any incoming faxes from any other IP Address or hostname.
Step 4	**fax print** *E.164-number* **Example:** se-10-0-0-0(config)# fax print 5550112	Configures the system level fax number for printing the faxes.
Step 5	**exit** **Example:** se-10-0-0-0(config)# exit	Returns to privileged EXEC mode.
Step 6	**voice mailbox owner** *name* **Example:** se-10-0-0-0(config)# voice mailbox owner owner22	Creates a mailbox for the specified user and enters mailbox configuration mode.
Step 7	**enable fax** **Example:** se-10-0-0-0(config)# end	Enables the specified mailbox to receive faxes from a fax gateway.
Step 8	**end** **Example:** se-10-0-0-0(config)# end	Returns to privileged EXEC mode.
Step 9	**show fax configuration** **Example:** se-10-0-0-0# show fax configuration	(Optional) Displays the configuration for the inbound fax gateway, outbound fax gateway, and the default fax number which is used for printing faxes.

Example

The following sample configuration configures the fax parameters on Cisco Unity Express module:

```
se-10-0-0-0# config t
se-10-0-0-0(config)# fax gateway inbound address 172.21.21.40
se-10-0-0-0(config)# fax gateway outbound address 172.21.21.40
se-10-0-0-0(config)# fax print 5550112
se-10-0-0-0(config)# exit
se-10-0-0-0(config)# voice mailbox owner owner22
se-10-0-0-0(config)# enable fax
se-10-0-0-0(config)# end
```

The output for show fax configuration is similar to the following:

```
se-10-0-0-0> show fax configuration

Inbound Fax Gateway: 172.21.21.40
```

```
Outbound Fax Gateway: 172.21.21.40
Fax Printing Number: 5550112
```

Configuring SMTP Parameters

Cisco Unity Express supports various features which need to send outgoing e-mail messages. In order to send these e-mails, an external SMTP server is required.

This section describes how to configure an external SMTP server and its parameters on the Cisco Unity Express module. The SMTP server address can either be a hostname or IP address. To use a hostname, verify that the DNS server is configured.

If the SMTP server requires authentication, you must also provide the user ID and password of a valid user account on the SMTP server.

Configuring an SMTP Server

Use the following procedure to configure an SMTP server and its parameters in Cisco Unity Express configuration mode.

Required Data for This Procedure

- SMTP server hostname or IP address
- SMTP authentication parameters (user ID and password, or credential string)

SUMMARY STEPS

1. **config t**
2. **smtp server address** {*hostname* | *ip-address*} **authentication** {**none** | **username** *userid* **password** *password* | **credentials** *credential-string*}
3. **end**
4. show smtp server

DETAILED STEPS

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

Step 2	**smtp server address** {*hostname* | *ip-address*} **authentication** {**none** | **username** *userid* **password** *password* | **credentials** *credential-string*}
	
	Example:
	`se-10-0-0-0(config)# smtp server address 10.10.5.5 authentication none`
	`se-10-0-0-0(config)# smtp server address mainsmtp authentication username smtp123 password pwd123`
	`se-10-0-0-0(config)# smtp server address 172.16.1.1 authentication credentials 3CmyKjEFhzkjd8QxCVjv552jZsjj zh3bSd8ZZNgd+Y9J3xlk2B35j0nfGWTYHfmPSd8ZZNgd+Y9J3xlk2B35j0 nfGWTYHfmPSd8ZZNgd`
		Configures an SMTP server, which is required for sending outbound e-mails.
		• *hostname*—Hostname of the SMTP server.
		• *ip-address*—IP address of the SMTP server.
		• **none**—Indicates that the SMTP server does not require authentication.
		• *userid*—User ID of a valid user account on the SMTP server.
		• *password*— Password of a valid user account on the SMTP server.
		• *credential-string*—Authentication credential string for the SMTP server. Copy and paste this string from the running or startup configuration.

Step 3	**end**	Exits configuration mode.
Step 4	**show smtp server** **Example:** `se-10-0-0-0# show smtp server`	Displays the SMTP server settings.

Example

The following is sample output of the **show smtp server** command.

```
se-10-0-0-0# show smtp server

SMTP Server: 172.16.1.1
Authentication: Required
Username: smtp123
```

Configuring Historical Reporting

Starting with Cisco Unity Express 3.0, information and statistics related to call and application events can be saved in a historical reporting database on the module. This historical data can later be used to generate various types of usage reports using the Cisco Unified Communications Express Historical Reporting Client.

Collection of historical data is disabled by default. You must enable it before the system starts saving these statistics in the reporting database. However, if an IVR license is purchased and installed on the module, the collection of historical data gets automatically enabled.

Cisco Unity Express can store up to 365 days worth of historical data on NME-CUE, NM-CUE-EC, and NM-CUE; it can store up to 90 days of historical data on AIM-CUE. The historical reporting maintenance components consist of a database purging service that periodically removes any data older than this.

A special privilege is required for a user to be able to log in to the Cisco Unified Communications Express Historical Reporting Client software and view historical reports.

The following sections describe the procedures for configuring historical reporting parameters:

Configuring the Local Historical Reporting Database

Historical reporting data is stored in a local (internal) database. Use the **database local** command to configure storage of historical statistics on the local or internal database.

The **no** and **default** forms of this command have no effect.

SUMMARY STEPS

1. **config t**
2. **ccn reporting historical**
3. **database local**
4. **description** *text*
5. **enabled**
6. **end**
7. **exit**
8. **show ccn reporting historical**

DETAILED STEPS

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

Step 2	**ccn reporting historical**
	
	Example:
	`se-10-0-0-0(config)# ccn reporting historical`
	`se-10-0-0-0(config-hrdm)#`
		Enters historical reporting database configuration mode.

Step 3	**database local** **Example:** `se-10-0-0-0(config-hrdm)# database local`	Configures local database to log historical statistics for reporting. This command is for future use.
Step 4	**description** *word* **Example:** `se-10-0-0-0(config-hrdm)# description "Chicago office database"`	(Optional) Sets the description for the historical reporting database. Use quotes around the text. The default value of the description is the hostname of the Cisco Unity Express system. The **no** and **default** forms of this command set the description value to the configured hostname of the system.
Step 5	**enabled** **Example:** `se-10-0-0-0(config-hrdm)# enabled`	Enables historical reporting. The collection of historical data is disabled by default. You must enable it before the system starts saving these statistics in the reporting database. However, if an IVR license is purchased and installed on the module, the collection of historical data is automatically enabled Use the **no** form of this command to disable the historical reporting database. If the historical reporting database is disabled, call-related events are not stored in the database. Use the **default** form of this command to enable the database.
Step 6	**end** **Example:** `se-10-0-0-0(config-hrdm)# end`	Saves and exits historical reporting database configuration mode.
Step 7	**exit** **Example:** `se-10-0-0-0(config)# exit`	Exits global configuration mode.
Step 8	**show ccn reporting historical** **Example:** `se-10-0-0-0# show ccn reporting historical`	Displays the historical reporting database parameters.

Examples

Following is example output of the **show ccn reporting historical** command:

```
se-10-0-0-0# show ccn reporting historical

Database Information
--------------------
Enabled    : Yes
Description: Chicago office database
DB Usage: 50%
Current Maintenance Status: idle

Purge Schedule
--------------
Daily Time: 4:00 AM
Data older than 365 days will be purged
Date of last completed purge:

Purge Capacity Configuration
----------------------------
Email Address:  abcd@domain.com
Warning Capacity: 65%
Purge Capacity: 75%
Oldest Days to purge: 7
```

Configuring the Database Purge Schedule

Use the **purge schedule** command in historical reporting database configuration mode to update the daily schedule for automatic purging of historical data.

A daily purge starts at the time of day specified (in hours:minutes 24-hour format). Stored data that is older than that specified in the *days-to-keep* value (in days) is purged from the database starting daily at the time specified.

The default purge schedule is set at 04:00.

Note Because the purging of historical data on the module is resource-intensive, we recommend that the purge be scheduled to run during off-peak hours.

The default number of days is 90 for AIM-CUE and 365 for the NM-CUE-EC,NM-CUE, and NME-CUE. The maximum value you can specify for *days-to-keep* is summarized in Table 6. The **no** and **default** form of this command sets the purge scheduled time to 04:00, and the number of days to the default value for that particular system hardware module.

Table 6 Maximum Days-to-Keep Value

Database	Storage Limits
AIM-CUE	90 days or database 90% full
NM-CUE-EC, NM-CUE, and NME-CUE	365 days or database 90% full

SUMMARY STEPS

1. **config t**
2. **ccn reporting historical**

3. **purge schedule time** *hh:mm* **days-to-keep** *days*
4. **end**
5. **exit**
6. **show ccn reporting historical**

DETAILED STEPS

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

Step 2	**ccn reporting historical**
	
	Example:
	`se-10-0-0-0(config)# ccn reporting historical`
	`se-10-0-0-0(config-hrdm)#`
		Enters historical reporting database configuration mode.

Step 3	**purge schedule time** *hh:mm* **days-to-keep** *days* **Example:** `se-10-0-0-0(config-hrdm)# purge schedule time 04:00 days-to-keep 30`	Configures the daily purge schedule and the number of days of this historical data to retain data older than the specified days-to-keep value will get purged at the scheduled time.
Step 4	**end** **Example:** `se-10-0-0-0(config-hrdm)# end`	Saves and exits historical reporting database configuration mode.
Step 5	**exit** **Example:** `se-10-0-0-0(config)# exit`	Exits global configuration mode.
Step 6	**show ccn reporting historical** **Example:** `se-10-0-0-0# show ccn reporting historical`	Displays the historical reporting database parameters.

Examples

Following is example output of the **show ccn reporting historical** command:

```
se-10-0-0-0# show ccn reporting historical

Database Information
--------------------
Enabled    : Yes
Description: Chicago office database
DB Usage: 50%
Current Maintenance Status: idle
```

```
Purge Schedule
--------------
Daily Time: 5:00 AM
Data older than 30 days will be purged
Date of last completed purge:

Purge Capacity Configuration
----------------------------
Email Address:  abcd@domain.com
Warning Capacity: 65%
Purge Capacity: 75%
```

Configuring the Database Capacity Threshold for a Purge

Use the **purge purge-capacity** command in historical reporting database configuration mode to set the purge threshold as a percentage of the total database capacity and the number of days of historical data that is to be purged from the database.

When the database capacity reaches the configured threshold, historical data older than the configured *days-to-purge* value is removed from the database. The default purge capacity percentage is 90, and the *days-to-purge* default value is 7. The maximum purge capacity percentage value allowed is 90. The **no** and **default** form of this command sets the purge capacity percentage value to 90, and the number of *days-to-purge* to 7.

SUMMARY STEPS

1. **config t**
2. **ccn reporting historical**
3. **purge purge-capacity percentage** *percent* **days-to-purge** *days*
4. **end**
5. **exit**
6. **show ccn reporting historical**

DETAILED STEPS

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

Step 2	**ccn reporting historical**
	
	Example:
	`se-10-0-0-0(config)# ccn reporting historical`
	`se-10-0-0-0(config-hrdm)#`
		Enters historical reporting database configuration mode.

	Command or Action	Purpose
Step 3	**purge purge-capacity percentage** *percent* **days-to-purge** *days* **Example:** `se-10-0-0-0(config-hrdm)# purge purge-capacity percentage 95 days-to-purge 7`	Configures the purge capacity threshold and the number of days of historical data to be purged from the database.
Step 4	**end** **Example:** `se-10-0-0-0(config-hrdm)# end`	Saves and exits historical reporting database configuration mode.
Step 5	**exit** **Example:** `se-10-0-0-0(config)# exit`	Exits global configuration mode.
Step 6	**show ccn reporting historical** **Example:** `se-10-0-0-0# show ccn reporting historical`	Displays the historical reporting database parameters.

Examples

Following is example output of the **show ccn reporting historical** command:

```
se-10-0-0-0# show ccn reporting historical

Database Information
--------------------
Enabled    : Yes
Description: Chicago office database
DB Usage: 50%
Current Maintenance Status: idle

Purge Schedule
--------------
Daily Time: 5:00 AM
Data older than 30 days will be purged
Date of last completed purge:

Purge Capacity Configuration
----------------------------
Email Address:  abcd@domain.com
Warning Capacity: 65%
Purge Capacity: 75%
```

Configuring the Database the Threshold Capacity for Warning Notification

Use the **purge warning-capacity** command to configure a percentage value of the total database capacity that, when reached, causes the system to send an e-mail message warning that the database capacity is approaching its limit. To configure the e-mail address to which this warning message gets sent, see the "Configuring the Purge Notification E-mail Addresses" section on page 94.

The default warning capacity percentage is 85. The maximum warning capacity percentage value allowed is 90. The **no** and **default** forms of this command set the warning capacity to 85%.

SUMMARY STEPS

1. **config t**
2. **ccn reporting historical**
3. **purge warning-capacity percentage** *percent*
4. **end**
5. **exit**
6. **show ccn reporting historical**

DETAILED STEPS

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

Step 2	**ccn reporting historical**
	
	Example:
	`se-10-0-0-0(config)# ccn reporting historical`
	`se-10-0-0-0(config-hrdm)#`
		Enters historical reporting database configuration mode.

Step 3	**purge warning-capacity percentage** *percent* **Example:** `se-10-0-0-0(config-hrdm)# purge warning-capacity percentage 65`	Configures the percentage value of the total database capacity that, when reached, causes the system to send an e-mail message warning that the database capacity is approaching its limit.
Step 4	**end** **Example:** `se-10-0-0-0(config-hrdm)# end`	Saves and exits historical reporting database configuration mode.
Step 5	**exit** **Example:** `se-10-0-0-0(config)# exit`	Exits global configuration mode.
Step 6	**show ccn reporting historical** **Example:** `se-10-0-0-0# show ccn reporting historical`	Displays the historical reporting database parameters.

Examples

Following is example output of the **show ccn reporting historical** command:

```
se-10-0-0-0# show ccn reporting historical

Database Information
--------------------
Enabled    : Yes
Description: Chicago office database
DB Usage: 50%
Current Maintenance Status: idle

Purge Schedule
--------------
Daily Time: 5:00 AM
Data older than 30 days will be purged
Date of last completed purge: Fri Feb 10 22:00:00 EST

Purge Capacity Configuration
----------------------------
Email Address:  abcd@domain.com
Warning Capacity: 65%
Purge Capacity: 75%
```

Configuring the Purge Notification E-mail Addresses

Use the **purge notification** command to configure e-mail addresses of up to 255 characters in length, to which purge notification and warning messages are sent..

There is no default e-mail address. If an e-mail address is not configured, e-mail notifications are not sent.

If more than one e-mail address needs to be configured, enter the e-mail addresses separated by commas without spaces.

Use the **no** and **default** forms of this command to remove this configuration.

SUMMARY STEPS

1. **config t**
2. **ccn reporting historical**
3. **purge notification email address** *email-address*
4. **end**
5. **exit**
6. **show ccn reporting historical**

DETAILED STEPS

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

Step 2	**ccn reporting historical**
	
	Example:
	`se-10-0-0-0(config)# ccn reporting historical`
	`se-10-0-0-0(config-hrdm)#`
		Enters historical reporting database configuration mode.

Step 3	**purge notification email address** *email-address* **Example:** `se-10-0-0-0(config-hrdm)# purge notification email address abcd@efghij.com`	Configures an e-mail address or e-mail addresses, to which purge notification and warning messages are sent.
Step 4	**end** **Example:** `se-10-0-0-0(config-hrdm)# end`	Saves and exits historical reporting database configuration mode.
Step 5	**exit** **Example:** `se-10-0-0-0(config)# exit`	Exits global configuration mode.
Step 6	**show ccn reporting historical** **Example:** `se-10-0-0-0# show ccn reporting historical`	Displays the historical reporting database parameters.

Examples

Following is example output of the **show ccn reporting historical** command:

```
se-10-0-0-0# show ccn reporting historical

Database Information
--------------------
Enabled    : Yes
Description: Chicago office database
DB Usage: 50%
Current Maintenance Status: idle

Purge Schedule
--------------
Daily Time: 5:00 AM
Data older than 30 days will be purged
Date of last completed purge: Fri Feb 10 22:00:00 EST

Purge Capacity Configuration
----------------------------
```

```
Email Address:  abcd@domain.com
Warning Capacity: 65%
Purge Capacity: 75%
```

Manually Purging the Historical Reporting Database

Use the **purge now** command to initiate a manual purge of the historical reporting database and remove historical data older than the specified *days-to-keep* number of days.

When the database is purged, historical data older than the specified *days-to-keep* value (in the range of 1–1000 days) is removed from the database. The *days-to-keep* value is required to initiate a manual purge.

Note Because the purging of historical data on the module is resource-intensive, we recommend that the manual purge be done during off-peak hours.

SUMMARY STEPS

1. **ccn reporting historical purge now days-to-keep** *days*
2. **show ccn reporting historical**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**ccn reporting historical purge now days-to-keep** *days* **Example:** `se-10-0-0-0(config)# ccn reporting historical purge now days-to-keep 30`	Manually purges the historical reporting database and removes historical data older than the *days-to-keep* number of days.
Step 2	**show ccn reporting historical** **Example:** `se-10-0-0-0# show ccn reporting historical`	Displays the historical reporting database parameters.

Examples

The following example illustrates the output when the database is manually purged:

```
se-10-0-0-0# ccn reporting historical purge now days-to-keep 7
Historical Database Purge Initiated
-----------------------------------
Time:  Fri Feb 10 04:00:00 EST
Data older than [ 7 ] days will be purged
```

The following example illustrates the **show ccn reporting historical** output:

```
se-10-0-0-0# show ccn reporting historical

Database Information
--------------------
```

```
Enabled    : Yes
Description: Chicago office database
DB Usage: 50%
Current Maintenance Status: idle

Purge Schedule
--------------
Daily Time: 5:00 AM
Data older than 30 days will be purged
Date of last completed purge: Fri Feb 10 22:00:00 EST

Purge Capacity Configuration
----------------------------
Email Address:  abcd@domain.com
Warning Capacity: 65%
Purge Capacity: 75%
```

Exporting Historical Report Data to an External Server

You can export historical reporting call contact detailed records (CCDRs) to an external server from the Cisco Unity Express module for post processing. Use the **copy hrdb url** command to export ASCII comma separated values of the historical data to an external server as aflat file.

Note We recommend that this command be executed during off peak hours or when the system is in a quiescent state.

SUMMARY STEPS

1. **copy hrdb url** *url*

DETAILED STEPS

	Command or Action	Purpose

Step 1	**copy hrdb url** *url*
	
	Example:
	`se-10-0-0-0# copy hrdb url ftp://1.2.3.4/hr.txt`
	`% Total % Received % Xferd Average Speed Time Time Time Current`
	
	`Dload Upload Total Spent Left Speed`
	
	`100 3584k 0 0 0 3584k 0 1259k --:--:-- 0:00:02 --:--:-- 1794k`
	`se-10-0-0-0#`
		Copies and uploads the historical reporting data in ASCII comma separated value format from the module to the specified URL.

Examples

The following are output examples of ASCII files formatted as comma separated values (CSVs) that are uploaded to the external server:

```
1,0,0,1,2,3,-1,1001,2,-1,16904,2007-05-30 13:19:34.032,2007-05-30
13:19:41.357,-240,6666,6666,15000000001,2,voicemail,7,C3E380E8-E0811DC-8295BE88-935E7691@1
92.1.1.110,,,,,,,,,
2,0,0,1,2,3,-1,1001,2,-1,16912,2007-05-30 13:19:44.197,2007-05-30
13:19:47.194,-240,6666,6666,15000000002,2,voicemail,2,CAEC0AEE-E0811DC-8299BE88-935E7691@1
92.1.1.110,,,,,,,,,

3,0,0,1,2,3,-1,1001,2,-1,16902,2007-05-30 13:19:55.992,2007-05-30
13:19:59.575,-240,6666,6666,15000000003,2,voicemail,3,D1F49256-E0811DC-829DBE88-935E7691@1
92.1.1.110,,,,,,,,,
```

Call contact detailed records (CCDRs) column fields described in Table 7 are listed sequentially in the ASCII CSV files :

You can define the custom variables 1 through 10 to suit your needs.

Table 7 Call Contact Detailed Records (CCDRs) Descriptions

Field Name	Data Type	Required Field	Possible Values	Description
sessionID	decimal(28)	NOT NULL		When a caller calls into the system, a unique session ID is established. This session ID is used for entire call, through all conferences and transfers.
sessionSeqNum	smallint	NOT NULL	[0, 1, 2, 3, …]	Each transfer of a call creates a new sequence number, but the session ID remains the same.
profileID	int	NOT NULL		Always set to 0 (reserved for future use).
contactType	tinyint	NOT NULL	1 = incoming 2 = outgoing 3 = internal	Incoming calls are those calls coming into the system. Outgoing call are calls originated by the Cisco Unity Express system. Internal calls are transfers.
contactDisposition	tinyint	NOT NULL	1 = abandoned 2 = handled	The call was either processed or abandoned during this part of the call.
originatorType	tinyint	NOT NULL	2= device 3= unknown	Device indicates call was originated by the CTI port. Unknown device includes gateway.
originatorID	int	NULL	CTI port, NULL	For gateway or unknown originator type, the value is NULL.
originatorDN	nvarchar(30)	NULL		Call ANI, the telephone number of the originator of the caller. For gateway or unknown originator type, the value is NULL.
destinationType	smallint	NULL	2 = device 3= unknown	Device indicates call was presented to a CTI port. Unknown device includes gateway.
destinationID	int	NULL	CTI port, NULL	For gateway or unknown destination type, the value is NULL.
destinationDN	nvarchar(30)	NULL		For gateway or unknown destination type, the value is NULL.
startDateTime	datetime	NOT NULL		Start date and time when this call leg was connected.

Table 7 Call Contact Detailed Records (CCDRs) Descriptions (continued)

Field Name	Data Type	Required Field	Possible Values	Description
endDateTime	datetime	NOT NULL		End date and time when this call leg was transferred or disconnected.
gmtOffset	smallint	NOT NULL		DST adjusted offset.
calledNumber	nvarchar(30)	NOT NULL		If the call was a transfer, this is the number to which the call was transferred. In other cases, this information is the same as the Original Called Number.
origCalledNumber	nvarchar(30)	NOT NULL		Telephone number the caller originally dialed.
applicationTaskID	decimal(28)	NULL		Task ID of currently executing application.
applicationID	int	NULL		Unique identifier of the application that processed this call.
applicationName	nvarchar(30)	NULL		Application name that processed this call.
connectTime	smallint	NULL		Number of seconds for which this call leg was in answered or connected state.
callID	varchar(64)			Globally unique Call ID
customVariable1	varchar (40)	NULL		Contents of the first custom variable of the currently executing application.
customVariable2	varchar (40)	NULL		Contents of the second custom variable of the currently executing application.
customVariable3	varchar (40)	NULL		Contents of the third custom variable of the currently executing application.
customVariable4	varchar (40)	NULL		Contents of the fourth custom variable of the currently executing application.
customVariable5	varchar (40)	NULL		Contents of the fifth custom variable of the currently executing application.
customVariable6	varchar (40)	NULL		Contents of the sixth custom variable of the currently executing application.
customVariable7	varchar (40)	NULL		Contents of the seventh custom variable of the currently executing application.
customVariable8	varchar (40)	NULL		Contents of the eighth custom variable of the currently executing application.
customVariable9	varchar (40)	NULL		Contents of the ninth custom variable of the currently executing application.
customVariable10	varchar (256)	NULL		Contents of the tenth custom variable of the currently executing application.

Assigning Historical Report Viewing Privileges to a Group

A special privilege is required for a user to be able to log in to the Cisco Unified Communications Express Historical Reporting Client software and view historical reports.

The name of the privilege required for this purpose is ViewHistoricalReports. All members of the group, which has this privilege, are able to view historical reports.

See the "Configuring Privileges" section on page 110 for details on assigning privileges.

SUMMARY STEPS

1. **config t**
2. **groupname** *name* **privilege ViewHIstoricalReports**
3. **exit**
4. **show** *groupname* privileges

DETAILED STEPS

	Command or Action	Purpose
Step 1	**config t** **Example:** `se-10-0-0-0# config t`	Enters global configuration mode.
Step 2	**groupname** *name* **privilege ViewHIstoricalReports** **Example:** `se-10-0-0-0(config)# groupname myGroup privilege ViewHIstoricalReports`	Allows the specified group name to view historical statistics reports.
Step 3	**end** **Example:** `se-10-0-0-0(config)# end`	Saves and exits global configuration mode.
Step 4	**show** *groupname* **privileges** **Example:** `se-10-0-0-0# show ccn groupname`	Displays the which privileges are set for the specified group names.

Examples

An example of the sequence of commands for assigning historical report viewing privilege is as follows:

```
se-10-0-0-0# config t
se-10-0-0-0(config)# groupname my_group privilege ViewHistoricalReports
se-10-0-0-0(config)# exit
se-10-0-0-0# show groups privileges
```