



## Configuring System Components

---

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

This chapter configures the basic Cisco Unity Express components:

- JTAPI parameters that Cisco Unity Express needs to communicate with Cisco CallManager.
- Voice mail, auto attendant, and greeting management system applications that ship with Cisco Unity Express.

Additional procedures for configuring optional or advanced system components, such as servers and custom auto attendant scripts, are described in [“Advanced Configuration” on page 135](#).

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, and troubleshooting systems.

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

- [Configuring JTAPI Parameters, page 72](#)
- [Configuring the Default Voice Mail Application, page 74](#)
- [Configuring the Default Auto Attendant Application, page 77](#)
- [Configuring the Default Greeting Management System Application, page 80](#)
- [Configuring Auto Attendant Prompts, page 80](#)
- [Configuring Auto Attendant Scripts, page 82](#)
- [Configuring JTAPI Triggers for the Applications, page 84](#)
- [Deleting a JTAPI Application Trigger, page 87](#)
- [Deleting an Application, page 89](#)
- [Configuring System-Wide Mailbox Default Values, page 93](#)
- [Checking AIM Flash Memory Wear Activity, page 95](#)

# Configuring JTAPI Parameters

Use this procedure to configure the parameters that Cisco Unity Express needs to communicate with Cisco CallManager.

## Prerequisites

The following information is required to configure the JTAPI parameters:

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

## SUMMARY STEPS

- 
- Step 1** `config t`
- Step 2** `ccn subsystem jtapi`
- Step 3** `ccm-manager address primary-server-ipaddress | primary-server-hostname  
{secondary-server-ipaddress | secondary-server-hostname}  
{tertiary-server-ipaddress | tertiary-server-hostname}`
- Step 4** `ccm-manager username jtapi-user-id password jtapi-user-password`
- Step 5** `ctiport cti-port`
- Step 6** `end`
- Step 7** `exit`
- Step 8** `show ccn subsystem jtapi`




---

**Note** One other command, **ccm-manager credentials**, is for internal Cisco use only.

---

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>config t</code>	Enters configuration mode.
Step 2	<code>ccn subsystem jtapi</code>	Enters JTAPI configuration mode.

	Command or Action	Purpose
Step 3	<pre>ccm-manager address primary-server-ipaddress   primary-server-hostname {secondary-server-ipaddress   secondary-server-hostname} {tertiary-server-ipaddress   tertiary-server-hostname}</pre> <p><b>Example:</b></p> <pre>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</pre>	Specifies up to 3 Cisco CallManager servers. The server IP addresses or hostnames can be entered on one line or on separate command lines. If entered on separate lines, the servers are assigned in order as primary, secondary, and tertiary servers.
Step 4	<pre>ccm-manager username jtapi-user-id password jtapi-user-password</pre> <p><b>Example:</b></p> <pre>se-10-0-0-0(config-jtapi)# ccm-manager username jtapiuser password myjtapi</pre>	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 CallManager.
Step 5	<pre>ctiport cti-port1 cti-port2 cti-port3 cti-port4...</pre> <p><b>Example:</b></p> <pre>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</pre> <pre>se-10-0-0-0(config-jtapi)# ctiport 6001 6002 6003 6004 6005 6006 6007 6008</pre>	Specifies the JTAPI CTI ports that are configured on Cisco CallManager and that are associated with the Cisco CallManager JTAPI user.  Repeat the command to enter more than one port number or enter the ports on one line. For AIM, specify 4 ports. For NM, specify 8 ports.
Step 6	<pre>end</pre>	Exits JTAPI configuration mode.
Step 7	<pre>exit</pre>	Exits configuration mode.
Step 8	<pre>show ccn subsystem jtapi</pre> <p><b>Example:</b></p> <pre>se-10-0-0-0# show ccn subsystem jtapi</pre>	Displays configured JTAPI parameters.
Step 9	<pre>copy running-config startup-config</pre>	Copies the configuration changes to the startup configuration.

## Examples

The following example illustrates the `show ccn subsystem jtapi` output:

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

Cisco CallManager:                10.100.10.120
CCM Username:                     jtapiuser
CCM Password:                     *****
Call Control Group 1 CTI ports:   7008,7009,7010,7011
se-10-0-0-0#
```

# Configuring the Default Voice Mail Application

After the Cisco Unity Express software is installed on the system, the voice mail application that ships with Cisco Unity Express must be configured using the procedures described in this section. The application is enabled by default.

To configure the voice mail access and operator telephone numbers, see [“Configuring JTAPI Triggers for the Applications” on page 84](#).

The commands are used in both EXEC and configuration modes.

## Sharing Ports Among Applications and Triggers

One of the parameters that you may configure for the voice mail and auto attendant applications is the maximum number of callers who can access the application concurrently at any given time. This parameter, **maxsessions**, is limited by the number of ports on the Cisco Unity Express module. (See [“Software Licenses and Factory-set Limits” on page 33](#) for the number of ports on your module.) The ports are configured with the **ctiports** command (see [“Configuring JTAPI Parameters” section on page 72](#).)

Consider your expected call traffic when assigning the number of ports to an application. One application may need more available ports than another, but each application should have at least one port available for incoming calls.

Suppose, for example, that your module has 4 ports and you assign 4 to the voice mail application **maxsessions** and 4 to the auto attendant **maxsessions**. If 4 callers access voice mail simultaneously, no ports will be available for auto attendant callers. Only when 0, 1, 2, or 3 callers access voice mail simultaneously will at least one port be available for auto attendant.

Suppose, instead, you assign 3 to the voice mail **maxsessions** and 3 to the auto attendant **maxsessions**. At no time will one application use up all the ports. If voice mail has 3 active calls, then 1 caller can access auto attendant. A second call to auto attendant will not go through at that moment.

Similarly, you must assign the **maxsessions** parameter to each application trigger, which is the telephone number that activates the application’s script. The value of the trigger’s **maxsessions** must not exceed the application’s **maxsessions** value.

## Prerequisites

The following information is required to configure the default voice mail application:

- Application name
- Maximum number of users who can access voice mail simultaneously

## SUMMARY STEPS

1. **config t**
2. **ccn application *fullname***
3. (Optional) **description “*text*”**
4. **maxsessions *number***
5. **end**

6. `exit`
7. `show ccn scripts`
8. `show ccn application`
9. `copy running-config startup-config`

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>config t</code>	Enters configuration mode.
Step 2	<code>ccn application fullname</code>  Example: se-10-0-0-0(config)# <code>ccn application voicemail</code>	Specifies the application to configure and enters application configuration mode. Use the full name of the application for the <i>fullname</i> argument.
Step 3	<code>description "text"</code>  Example: se-10-0-0-0(config-application)# <code>description "Voice Mail"</code>	(Optional) Enter a description of the application. Use double quotes around the text.
Step 4	<code>maxsessions number</code>  Example: se-10-0-0-0(config-application)# <code>maxsessions 8</code>	Specifies the <i>number</i> of users who can access this application simultaneously. See the <a href="#">"Sharing Ports Among Applications and Triggers"</a> section on page 74 for guidelines on assigning this value.
Step 5	<code>end</code>	Exits application configuration mode.
Step 6	<code>exit</code>	Exits configuration mode.
Step 7	<code>show ccn scripts</code>  Example: se-10-0-0-0# <code>show ccn scripts</code>	Displays the configured script names and their descriptions.
Step 8	<code>show ccn application</code>  Example: se-10-0-0-0# <code>show ccn application</code>	Displays details about each configured application.
Step 9	<code>copy running-config startup-config</code>	Copies the configuration change to the startup configuration.

### Example

The following example illustrates the `show ccn scripts` output:

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

Name:                setmwi.aef
Create Date:         Tue Apr 20 20:22:27 PDT 1993
Last Modified Date:  Tue Apr 20 20:22:27 PDT 1993
Length in Bytes:    21990
```

```
Name:                voicebrowser.aef
Create Date:         Tue Apr 20 20:22:39 PDT 1993
Last Modified Date: Tue Apr 20 20:22:39 PDT 1993
Length in Bytes:    13409
```

```
Name:                aa.aef
Create Date:         Tue Apr 20 20:22:47 PDT 1993
Last Modified Date: Tue Apr 20 20:22:47 PDT 1993
Length in Bytes:    56227
```

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

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

```
Name:                voicemail
Description:         Voice Mail
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/login.vxml
se-10-0-0-0#
```

# Configuring the Default Auto Attendant Application

After the Cisco Unity Express software is installed on the system, the auto attendant application that ships with Cisco Unity Express must be configured using the procedures described in this section.

To configure the auto attendant access telephone number, see [“Configuring JTAPI Triggers for the Applications” on page 84](#).

The commands are used in both EXEC and configuration modes.

See [“Advanced Configuration” on page 135](#) for procedures to configure user-defined parameters.

## Prerequisites

The following information is required to configure auto attendant:

- To use your own welcome greeting, create a .wav file containing the prerecorded welcome greeting. This file must be uploaded to Cisco Unity Express module so that it can be located and saved in the auto attendant script. Alternatively, you can use the greeting management system (GMS) to record the welcome greeting. See [“Recording an Auto Attendant Greeting or Prompt File” on page 80](#) for guidelines on recording a greeting. See [“Uploading the Auto Attendant Greeting or Prompt File” on page 80](#) for the procedure to upload the prompt to Cisco Unity Express.
- Application name
- Number of times the auto attendant will replay instructions to a caller before the call is disconnected. This count begins when the caller moves past the main menu and starts to hear instructions for a submenu. The main menu will play five times and then, if the caller makes no choice or incorrect choices, will transfer to the operator.
- Extension number of the operator. Auto attendant dials this extension when the caller presses the zero (“0”) button.
- Telephone number that the caller must dial to reach the auto attendant. In many cases, this number is your company telephone number.
- Maximum number of callers that can access auto attendant simultaneously. See the [“Sharing Ports Among Applications and Triggers” section on page 74](#) for guidelines on assigning this value.

## SUMMARY STEPS

1. **config t**
2. **ccn application** *fullname*
3. (Optional) **description** “*text*”
4. **maxsessions** *number*
5. **parameter** *name* “*value*”
6. **enabled**
7. **end**
8. **exit**
9. **show ccn scripts**
10. **show ccn application**

## 11. copy running-config startup-config

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>config t</code>	Enters configuration mode.
Step 2	<code>ccn application fullname</code>  <b>Example:</b> <code>se-10-0-0-0(config)# ccn application AutoAttendant</code>	Specifies the application to configure and enters application configuration mode. Use the full name of the application for the <i>fullname</i> argument.
Step 3	<code>description "text"</code>  <b>Example:</b> <code>se-10-0-0-0(config-application)# description "Auto Attendant"</code>	(Optional) Enter a description of the application. Use double quotes around the text.
Step 4	<code>maxsessions number</code>  <b>Example:</b> <code>se-10-0-0-0(config-application)# maxsessions 8</code>	Specifies the number of users who can access this application simultaneously. See the <a href="#">“Sharing Ports Among Applications and Triggers”</a> section on page 74 for guidelines on assigning this value.
Step 5	<code>parameter name "value"</code>  <b>Example:</b> <code>se-10-0-0-0(config-application)# parameter operExtn "1000"</code> <code>se-10-0-0-0(config-application)# parameter MaxRetry "3"</code>	Specifies parameters for the application. Each parameter must have a name and a value, which is written within double quotes.  For the auto attendant application, the parameters are: <ul style="list-style-type: none"> <li><b>operExtn</b>—Extension that the system dials when a caller presses “0” to reach the auto attendant operator.</li> <li><b>MaxRetry</b>—Maximum number of times a user can incorrectly choose a submenu option before the application disconnects the call.</li> </ul>
Step 6	<code>enabled</code>  <b>Example:</b> <code>se-10-0-0-0(config-application)# enabled</code>	Allows the application to be accessible to the system.
Step 7	<code>end</code>	Exits application configuration mode.
Step 8	<code>exit</code>	Exits configuration mode.
Step 9	<code>show ccn scripts</code>  <b>Example:</b> <code>se-10-0-0-0# show ccn scripts</code>	Displays the configured script names and their descriptions.



	Command or Action	Purpose
Step 10	<code>show ccn application</code>  <b>Example:</b> <code>se-10-0-0-0# show ccn application</code>	Displays details about each configured application.
Step 11	<code>copy running-config startup-config</code>	Copies the configuration change to the startup configuration.

**Examples**

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

```

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

Name:                setmwi.aef
Create Date:         Tue Apr 20 20:22:27 PDT 1993
Last Modified Date: Tue Apr 20 20:22:27 PDT 1993
Length in Bytes:    21990

Name:                voicebrowser.aef
Create Date:         Tue Apr 20 20:22:39 PDT 1993
Last Modified Date: Tue Apr 20 20:22:39 PDT 1993
Length in Bytes:    13409

Name:                aa.aef
Create Date:         Tue Apr 20 20:22:47 PDT 1993
Last Modified Date: Tue Apr 20 20:22:47 PDT 1993
Length in Bytes:    56227

Name:                promptgmt.aef
Create Date:         Tue Apr 20 20:22:59 PDT 1993
Last Modified Date: Tue Apr 20 20:22:59 PDT 1993
Length in Bytes:    80781
se-10-0-0-0#
    
```

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

```

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

Name:                autoattendant
Description:         autoattendant
Script:              aa.aef
ID number:           2
Enabled:              yes
Maximum number of sessions: 8
MaxRetry:            3
operExtn:            1000
welcomePrompt:      AAWelcome.wav
se-10-0-0-0#
    
```

# Configuring the Default Greeting Management System Application

The Cisco Unity Express module installation automatically configures the greeting management system (GMS) application. This application permits recording and administration of auto attendant prompts using the telephone user interface (TUI).

Only users with administrative (superuser) privileges or greeting management (ManagePrompt) privileges have access to the GMS. (See [“Adding and Modifying a Group” on page 103](#) for information about assigning privileges.) When a caller dials the GMS number, the GMS authenticates the caller by requesting the caller’s extension and PIN. The GMS disconnects the caller if the caller does not have administrative authority.

To configure the GMS access telephone number, see [“Configuring JTAPI Triggers for the Applications” on page 84](#).

## Configuring Auto Attendant Prompts

Release 1.1 supports customized greeting and prompt files. The NM supports up to 50 prompts; the AIM supports up to 25 prompts.

Customizing prompts requires the following procedures:

- [Recording an Auto Attendant Greeting or Prompt File](#)
- [Uploading the Auto Attendant Greeting or Prompt File](#)
- (Optional) [Downloading an Auto Attendant Greeting or Prompt File](#)
- (Optional) [Deleting an Auto Attendant Greeting or Prompt File](#)

## Recording an Auto Attendant Greeting or Prompt File

Two methods are available to create auto attendant 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. After recording the greeting, use the GUI or Cisco Unity Express CLI **ccn copy** command to copy the file in to the Cisco Unity Express system. See the next section, [“Uploading the Auto Attendant Greeting or Prompt File,”](#) for the upload procedure.
- Use the GMS on the TUI to record the greeting or prompt. Dial the GMS telephone number and select the option to record a greeting. When finished recording, save the file. GMS automatically saves the file in Cisco Unity Express.

The GMS prompt filename has the format UserPrompt\_DateTime.wav, for example: UserPrompt\_11152003144055.wav. You may want to use CLI commands or GUI options to download the file to a PC, rename the file with a meaningful name, then upload the file back to Cisco Unity Express.

## Uploading the Auto Attendant Greeting or Prompt File

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

**ccn copy url** *source-ip-address* **prompt** *prompt-filename*

**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.

## Downloading an Auto Attendant Greeting or Prompt File

Greetings and prompts can be copied from the auto attendant and stored to another server or PC.

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

**ccn copy prompt** *prompt-file-name* **url** *destination-ip-address*

**Example:**

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

## Deleting an Auto Attendant Greeting or Prompt File

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

**ccn delete prompt** *prompt-filename*

**Example:**

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

# Configuring Auto Attendant Scripts

Release 1.1 supports customized script files. The NM supports up to 8 scripts; the AIM supports up to 4 scripts.

Customizing scripts requires the following procedures:

- [Creating an Auto Attendant Script, page 82](#)
- [Uploading the Auto Attendant Script File, page 82](#)
- (Optional) [Downloading an Auto Attendant Script File, page 82](#)
- (Optional) [Deleting an Auto Attendant Script File, page 83](#)

## Creating an Auto Attendant Script

The auto attendant script file is created using the script editor program. Refer to *Cisco Unity Express Script Editor Installation and Configuration Guide* for guidelines and procedures for creating a script file.

The file cannot be larger than 256 KB.

After creating the script, use the GUI or Cisco Unity Express CLI **ccn copy** command to copy the file in to the Cisco Unity Express system. See the next section, “[Uploading the Auto Attendant Script File](#),” for the upload procedure.

## Uploading the Auto Attendant Script File

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

```
ccn copy url source-ip-address script script-filename
```

### Example:

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

## Downloading an Auto Attendant Script File

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

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

```
ccn copy script prompt-file-name url destination-ip-address
```

### Example:

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

## Deleting an Auto Attendant 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 GMSscript.aef  
Are you sure you want to delete this script? (y/n)
```

# Configuring JTAPI Triggers for the Applications

After the Cisco CallManager parameters are configured, Cisco Unity Express must be configured to start the voice mail, auto attendant, and GMS applications when a specific signal, or trigger, is invoked. The trigger is a telephone number. When a caller dials a specified telephone number, Cisco Unity Express starts the voice mail, auto attendant, or GMS application.

See [“Advanced Configuration” on page 135](#) for procedures to configure multiple triggers for an application.

## Prerequisites

The following information is required to configure the SIP triggers:

- Telephone number that invokes the application. The number must not be the same for both voice mail and auto attendant.
- 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. For Releases 1.1 and 1.0, only U.S. English is available.
- Maximum number of callers that can access the trigger simultaneously. See the [“Sharing Ports Among Applications and Triggers” section on page 74](#) for guidelines on assigning this value.

## SUMMARY STEPS

Starting from EXEC mode:

1. **config t**
2. **ccn trigger jtapi phonenum** *number*
3. **application** *string*
4. **enabled**
5. **maxsessions** *number*
6. (Optional) **no** *parameter*
7. **end**
8. **exit**
9. **show ccn trigger**
10. **copy running-config startup-config**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>config t</code>	Enters configuration mode.
Step 2	<code>ccn trigger jtapi phonenumber <i>number</i></code>  <b>Example:</b> se-10-0-0-0(config)# <code>ccn trigger jtapi phonenumber 6700</code>	Specifies the telephone number that acts as the trigger to start the application on Cisco Unity Express and enters trigger configuration mode. The <i>number</i> value must match a JTAPI route point configured on Cisco CallManager.
Step 3	<code>application <i>string</i></code>  <b>Example:</b> se-10-0-0-0(config-trigger)# <code>application voicemail</code> se-10-0-0-0(config-trigger)# <code>application autoattendant</code> se-10-0-0-0(config-trigger)# <code>application promptmanagement</code>	Specifies the name of the application to start when the trigger is entered.
Step 4	<code>enabled</code>  <b>Example:</b> se-10-0-0-0(config-trigger)# <code>enabled</code>	Enables the trigger.
Step 5	<code>maxsessions <i>number</i></code>  <b>Example:</b> se-10-0-0-0(config-trigger)# <code>maxsessions 3</code>	Specifies the maximum number of callers the application can handle simultaneously. See the <a href="#">“Sharing Ports Among Applications and Triggers”</a> section on page 74 for guidelines on assigning this value.
Step 6	<code>end</code>	Exits trigger configuration mode.
Step 7	<code>exit</code>	Exits configuration mode.
Step 8	<code>show ccn trigger</code>  <b>Example:</b> se-10-0-0-0# <code>show ccn trigger</code>	Displays the parameter values for all configured triggers.
Step 9	<code>copy running-config startup-config</code>	Copies the configuration change to the startup configuration.

### Example

The following sample configuration sets two triggers, one for the voice mail application and one for the auto attendant application:

```
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#
```

The output for **show ccn trigger** will look similar to the following:

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

Name:                6500
Type:                JTAPI
Application:         voicemail
Locale:              en_US
Idle Timeout:        600
Enabled:             yes
Maximum number of sessions: 4

Name:                6700
Type:                JTAPI
Application:         autoattendant
Locale:              en_US
Idle Timeout:        600
Enabled:             yes
Maximum number of sessions: 8
se-10-0-0-0#
```



# Deleting a JTAPI Application Trigger

Use this procedure to delete a JTAPI application trigger. Deleting the trigger does not delete the application, although the application needs at least one trigger in order to be invoked by the system.

## Prerequisites

The trigger number is required to delete a trigger.

## SUMMARY STEPS

1. `show ccn trigger`
2. `config t`
3. `no ccn trigger jtapi phonenumber number`
4. `exit`
5. `show ccn trigger`
6. `copy running-config startup-config`

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>show ccn trigger</code>	Displays the currently configured triggers. Look for the telephone number that you want to delete. Verify that this telephone number is associated with the correct application.
Step 2	<code>config t</code>	Enters configuration mode.
Step 3	<code>no ccn trigger jtapi phonenumber <i>number</i></code>	Deletes the trigger number.
	<b>Example:</b> <code>se-10-0-0-0(config)# no ccn trigger jtapi phonenumber 5000</code>	
Step 4	<code>exit</code>	Exits configuration mode.
Step 5	<code>show ccn trigger</code>	Displays the configured triggers.
Step 6	<code>copy running-config startup-config</code>	Copies the configuration change to the startup configuration.

## Example

The output for `show ccn trigger` might look similar to the following:

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

Name:                6500
Type:                 JTAPI
Application:          voicemail
Locale:               en_US
```

## ■ Deleting a JTAPI Application Trigger

```

Idle Timeout:          600
Enabled:              yes
Maximum number of sessions: 4

Name:                 6700
Type:                 JTAPI
Application:          autoattendant
Locale:               en_US
Idle Timeout:         600
Enabled:              yes
Maximum number of sessions: 8
se-10-0-0-0#

```

The following configuration removes a trigger from the voice mail application:

```

se-10-0-0-0# config t
se-10-0-0-0(config)# no ccn trigger jtapi phonenumber 6500
se-10-0-0-0(config)# exit

```

Now the output for **show ccn trigger** might look similar to the following:

```

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

Name:                 6700
Type:                 JTAPI
Application:          autoattendant
Locale:               en_US
Idle Timeout:         600
Enabled:              yes
Maximum number of sessions: 8
se-10-0-0-0#

```

# Deleting an Application

If you configure an application that you do not want to keep, use this procedure to delete the application and any triggers associated with that application. If you do not delete the triggers, the application will be invoked when one of the triggers is called.

After you delete the application and triggers, the script associated with the application remains installed on your server but is not used by Cisco Unity Express.

To make this application usable, reconfigure it.

The default voice mail, auto attendant, and greeting management system applications that shipped with Cisco Unity Express cannot be deleted.

## Prerequisites

The following information is required to delete an application:

- Application name
- All trigger numbers associated with the application

## SUMMARY STEPS

1. **show ccn application**
2. **show ccn trigger**
3. **config t**
4. **no ccn trigger jtapi phonenummer *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	<code>show ccn application</code>	Displays the currently configured applications. Look for the name of the application you want to delete.
Step 2	<code>show ccn trigger</code>	Displays the currently configured triggers. Look for the telephone numbers associated with the application you want to delete.
Step 3	<code>config t</code>	Enters configuration mode.

## ■ Deleting an Application

	Command or Action	Purpose
Step 4	<code>no ccn trigger jtapi phonenumber number</code>  <b>Example:</b> <code>se-10-0-0-0(config)# no ccn trigger jtapi phonenumber 6700</code>	Deletes a trigger associated with this application. Repeat this command for each trigger associated with the application.
Step 5	<code>no ccn application name</code>  <b>Example:</b> <code>se-10-0-0-0(config)# no ccn application myapplication</code>	Deletes the application called <i>name</i> .
Step 6	<code>exit</code>	Exits configuration mode.
Step 7	<code>show ccn application</code>	Displays the currently configured applications. Confirm that the deleted application is not displayed.
Step 8	<code>show ccn trigger</code>	Displays the triggers for each configured application.
Step 9	<code>copy running-config startup-config</code>	Copies the configuration change to the startup configuration.

**Example**

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/m
                    bxLogout.jsp
uri:                 http://localhost/voicemail/vxmlscripts/1
                    ogin.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

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:                6700
Type:                JTAPI
Application:         voicemail
Locale:              systemDefault
Idle Timeout:        5000
Enabled:             yes
Maximum number of sessions: 4

Name:                6500
Type:                JTAPI
Application:         autoattendant
Locale:              systemDefault
Idle Timeout:        5000
Enabled:             yes
Maximum number of sessions: 8

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

```

The following configuration deletes the myapplication application and its trigger:

```

se-10-0-0-0# config t
se-10-0-0-0(config)# no ccn trigger jtapi phonenumber 7200
se-10-0-0-0(config)# no ccn application myapplication
se-10-0-0-0(config)# exit

```

Now the output for the **show** commands looks similar to this:

```

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/1
ogin.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#

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

Name:                6700

```

## ■ Deleting an Application

```
Type: JTAPI
Application: voicemail
Locale: systemDefault
Idle Timeout: 5000
Enabled: yes
Maximum number of sessions: 4

Name: 6500
Type: JTAPI
Application: autoattendant
Locale: systemDefault
Idle Timeout: 5000
Enabled: yes
Maximum number of sessions: 8
se-10-0-0-0#
```

# Configuring System-Wide Mailbox Default Values

The following system-wide parameters are configurable for all new voice mailboxes. These values are assigned automatically to a new voice mailbox. Several of these values have factory default values. See “[Feature Overview](#)” on page 31 for the maximum values for your system.

- **Capacity**—The total amount of storage time in hours allowed for all mailboxes in the system. The factory default is the maximum allowed storage for your system.
- **Expiration date**—The number of days a message is saved in the mailbox. When the user logs in to the voice mailbox, the user hears a message listing all the expired messages. The user can save, skip, or delete each message. The factory default value is 30 days.
- **Language**—The language used for voice mail prompts. In Release 1.1, U.S. English is the only language available.
- **Mailbox size**—The maximum number of seconds of storage for voice messages in a mailbox. The factory default value is determined by dividing the maximum storage capacity by the maximum number of mailboxes (personal plus general delivery).
- **Message length**—The maximum number of seconds for any one stored message in a mailbox. The factory default is 60 seconds.
- **Recording time**—The maximum amount of time for a user’s recorded mailbox greeting.
- **Operator extension**—The extension of the voice mail operator.

**Caution**

The voice mail telephone number and the voice mail operator’s telephone number should not be the same. If they are, then a user trying to call the operator while in the voice mail system will be directed back to the voice mail system. Also, an outside caller who presses the button for the operator will be connected to the voice mail system.

See “[Creating and Modifying a Voice Mailbox](#)” on page 107 for the procedure to configure different values for mailbox size, message length, and expiration date for a specific mailbox.

## SUMMARY STEPS

1. **config t**
2. **voicemail capacity time** *minutes*
3. **voicemail defaults** {**expiration** *days* | **language** | **mailboxsize** *mailboxsize\_seconds* | **messagesize** *messagesize\_seconds*}
4. **voicemail operator telephone** *tel\_number*
5. **voicemail recording time** *minutes*
6. **exit**
7. **copy running-config startup-config**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>config t</code>	Enters configuration mode.
Step 2	<code>voicemail capacity time minutes</code>  <b>Example:</b> <code>se-10-0-0-0(config)# voicemail capacity time 3000</code>	Sets the <i>time</i> value as the system-wide maximum storage space in minutes allowed for all configured mailboxes.
Step 3	<code>voicemail default {expiration days   language   mailboxsize mailboxsize_seconds   messagesize messagesize_seconds}</code>  <b>Example:</b> <code>se-10-0-0-0(config)# voicemail default expiration 30</code> <code>se-10-0-0-0(config)# voicemail default mailboxsize 300</code> <code>se-10-0-0-0(config)# voicemail default messagesize 120</code>	Assigns default values for new individual or general delivery mailboxes. These values can be configured later to other values for specific mailboxes. <ul style="list-style-type: none"> <li>• <b>expiration days</b>—Sets the number of days a message can be stored in a mailbox before the voice mail system deletes it.</li> <li>• <b>language</b>—Specifies the language used for voice mail prompts. In Release 1.0, only U.S. English is available.</li> <li>• <b>mailboxsize mailboxsize_seconds</b>—Sets the maximum number of seconds for storing messages in a mailbox.</li> <li>• <b>messagesize messagesize_seconds</b>—Sets the maximum number of seconds for a caller's message stored in a mailbox.</li> </ul>
Step 4	<code>voicemail operator telephone tel_number</code>  <b>Example:</b> <code>se-10-0-0-0(config)# voicemail operator telephone 9000</code>	Assigns the <i>tel_number</i> value as the voice mail operator's extension. A mailbox owner dials this extension while in the voice mail system to reach the voice mail operator. Do not assign this extension to a group. This extension need not be the same as the auto attendant operator extension.
Step 5	<code>voicemail recording time minutes</code>  <b>Example:</b> <code>se-10-0-0-0(config)# voicemail recording time 10</code>	Assigns the <i>time</i> value in minutes as the maximum recording time for any greeting or message in the voice mail system.
Step 6	<code>exit</code>	Exits configuration mode.
Step 7	<code>copy running-config startup-config</code>	Copies the configuration change to the startup configuration.



# Checking AIM Flash Memory Wear Activity

Cisco Unity Express tracks the use and wear of the AIM flash memory as log and trace data are saved to the module. To display this data, use the **show interface ide 0** command in Cisco Unity Express EXEC mode.

**show interface ide 0**

The following is sample output:

```
se-10-0-0-0# show interface ide 0
IDE hd0 is up, line protocol is up
  3496 reads, 46828544 bytes
  0 read errors
  9409 write, 137857024 bytes
  0 write errors
  0.09933333333333333% worn
```

To check the log and trace files on the flash memory, use the **log trace** command in Cisco Unity Express EXEC mode.

**log trace**

Logging and tracing to the flash memory is turned off by default. Executing the **log trace** command starts the log and trace functions immediately.

The command displays the `atrace.log` and `messages.log` files. Each file has a fixed length of 10 MB, and tracing or logging stops automatically when the file reaches this length. New files overwrite the old files.

