Configuring Cisco Unified Communications Gateway Services

Configuring Cisco Unified Communications Gateway Services - Nonsecure Mode

Note If voice gateway is already configured with Cisco Unified Communications Gateway Services in secure mode, remove the secure mode configurations before you proceed with nonsecure mode configuration. Use the **no uc wsapi** command to remove the non-secure mode configuration.

You can configure Cisco Unified Communications Gateway Services in either nonsecure mode or secure mode. When you configure Cisco Unified Communications Gateway Services in nonsecure mode, the command **ip http active-session-modules** *all* is enabled by default, irrespective of whether UC Service APIs provisioned or not. This feature enables all the HTTP applications like UC Gateway Services APIs to register internally for enabling the service. However, if you configure **ip http active-session-modules** *none*, then none of the web applications will register.

To ensure that IOS applications are not enabled by default, configure the following. It explicitly enables web services for specific features of UC Gateway services:

ip http session-module-list [*module_list_name*] [*list_of_modules_to_be_registered*]

ip http active-session-modules [*module_list_name*]

Example

The following is a sample configuration for nonsecure mode. To register only WSAPI services, configure the following:

```
ip http session-module-list wsapi cisco_xmf,cisco_xcc,cisco_xsvc,cisco_xcdr
ip http active-session-modules wsapi
```

To register only XCC services:

```
ip http session-module-list wsapi cisco_xcc
ip http active-session-modules wsapi
```

Prerequisite

- Cisco IOS Release 15.2(2)T or later
- Cisco IOS XE Release 3.10 or later

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip http server**
4. **ip http max-connection** *value*
5. **ip http timeout-policy idle** *seconds* **life** *seconds* **requests** *value*
6. **http client connection persistent**
7. **http client connection idle timeout** *seconds*
8. **uc wsapi**
9. **message-exchange max-failures** *number*
10. **probing max-failures** *number*
11. **probing interval keepalive** *seconds*
12. **probing interval negative** *seconds*
13. **source-address** *ip-address*
14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**`ip http server`** **Example:** `Device(config)# ip http server`	Enables the HTTP server (web server) on the system.
Step 4	**`ip http max-connection`** *`value`* **Example:** `Device(config)# ip http max-connection 100`	Sets the maximum number of concurrent connections to the HTTP sever that will be allowed. The default value is 5.

	Command or Action	Purpose
Step 5	**ip http timeout-policy idle** *seconds* **life** *seconds* **requests** *value* **Example:** `Device(config)# ip http timeout-policy idle 600 life 86400 requests 86400`	Sets the characteristics that determine how long a connection to the HTTP server should remain open. The characteristics are: **idle**—The maximum number of seconds the connection will be kept open if no data is received or response data can not be sent out on the connection. Note that a new value may not take effect on any already existing connections. If the server is too busy or the limit on the life time or the number of requests is reached, the connection may be closed sooner. The default value is 180 seconds (3 minutes). **life**—The maximum number of seconds the connection will be kept open, from the time the connection is established. Note that the new value may not take effect on any already existing connections. If the server is too busy or the limit on the idle time or the number of requests is reached, it may close the connection sooner. Also, since the server will not close the connection while actively processing a request, the connection may remain open longer than the specified life time if processing is occurring when the life maximum is reached. In this case, the connection will be closed when processing finishes. The default value is 180 seconds (3 minutes). The maximum value is 86400 seconds (24 hours). **requests**—The maximum limit on the number of requests processed on a persistent connection before it is closed. Note that the new value may not take effect on any already existing connections. If the server is too busy or the limit on the idle time or the life time is reached, the connection may be closed before the maximum number of requests are processed. The default value is 1. The maximum value is 86400.
Step 6	**http client connection persistent** **Example:** `Device(config)# http client connection persistent`	Enables HTTP persistent connections. **Note** When this command is configured, multiple files are loaded using the same connection. Executing this command determines whether the HTTP client requests a keepalive or closed connection from the server. The HTTP server is responsible for granting or denying the keepalive connection request from the client.
Step 7	**http client connection idle timeout** *seconds* **Example:** `Device(config)# http client connection idle timeout 600`	Sets the number of seconds that the client waits in the idle state until it closes the connection.

	Command or Action	Purpose
Step 8	**uc wsapi** **Example:** `Device(config)# uc wsapi`	Enters Cisco Unified Communications Gateway Services configuration mode.
Step 9	**message-exchange max-failures** *number* **Example:** `Device(config-uc-wsapi)# message-exchange max failures 2`	Configures the maximum number of failed message exchanges between the application and the provider before the provider stops sending messages to the application. Range is 1 to 3. Default is 1.
Step 10	**probing max-failures** *number* **Example:** `Device(config-uc-wsapi)# probing max-failures 5`	Configures the maximum number of failed probing messages before the voice gateway unregisters the application. Range is 1 to 5. Default is 3.
Step 11	**probing interval keepalive** *seconds* **Example:** `Device(config-uc-wsapi)# probing interval keepalive 180`	Configures the interval between probing messages, in seconds. Default is 120 seconds.
Step 12	**probing interval negative** *seconds* **Example:** `Device(config-uc-wsapi)# probing interval negative 10`	Configures the interval between negative probing messages, in seconds.
Step 13	**source-address** *ip-address* **Example:** `Device(config-uc-wsapi)# source-address 10.25.12.13`	Configures the IP address (hostname) as the source IP address for the Cisco Unified Communications Gateway Services. **Note** The source IP address is used by the provider in the NotifyProviderStatus messages.
Step 14	**end** **Example:** `Device(config-uc-wsapi)# end`	Returns to privileged EXEC mode.

Configuring Cisco Unified Communications Gateway Services - Secure Mode

Note If the voice gateway is already configured with Cisco Unified Communications Gateway Services in nonsecure mode, remove the nonsecure mode configurations before you proceed with secure mode configuration.

You can configure Cisco Unified Communications Gateway Services in either nonsecure mode or secure mode. When you configure Cisco Unified Communications Gateway Services in secure mode, the command **ip http active-session-modules** *all* is enabled by default, irrespective of whether UC Service

APIs provisioned or not. Due to this, all the web applications are registered with NGINX proxy. This feature enables all the HTTP applications like UC Gateway Services APIs to register internally for enabling the service. However, if you configure **ip http secure-active-session-modules** *none*, then none of the web applications register with NGINX server.

To ensure that IOS applications are not enabled by default, configure the following. It explicitly enables web services for specific features of UC Gateway services:

ip http session-module-list [*module_list_name*] [*list_of_modules_to_be_registered*]

ip http secure-active-session-modules [*module_list_name*]

Example

The following is a sample configuration for secure mode. To register only WSAPI services, configure the following:

```
ip http session-module-list wsapi cisco_xmf,cisco_xcc,cisco_xsvc,cisco_xcdr
ip http secure-active-session-modules wsapi
```

To register only XCC services:

```
ip http session-module-list wsapi cisco_xcc
ip http secure-active-session-modules wsapi
```

Prerequisites

- Cisco IOS XE Everest Release 16.6.1 or later
- Application certificate ready to import in voice gateway
- Ensure that you have security and uck9 package licenses

Importing Application Certificate

Certificate is a digitally signed statement that is used to authenticate and to secure information on open networks.

When the voice gateway is behaving as a User Agent Server and receives HTTPS connection request from the application, the voice gateway requires the certificate of the application. You have to import the application certificate on to the voice gateway. By importing the application certificate, the voice gateway trusts the application, authenticates the request and establishes HTTPS connection.

Perform this procedure to import the application certificate to voice gateway.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *trustpoint-name*
4. **enrollment terminal**
5. **exit**
6. **crypto pki authenticate** *trust-point name*
7. **Copy the application certificate and paste it on the voice gateway console**
8. **exit**

DETAILED STEP

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**`crypto pki trustpoint`** *`trustpoint-name`* **Example:** `Device(config)#crypto pki trustpoint sampletpname`	Creates a trustpoint.
Step 4	**`enrollment terminal`** **Example:** `Device(ca-trustpoint)# enrollment terminal`	Specifies the manual cut-and-paste certificate enrollment method.
Step 5	**`exit`** **Example:** `Device(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode and returns to global configuration mode.

Step 6	**`crypto pki authenticate`** *`name`*
	
	`Device(config)# crypto pki authenticate sampletpname`
	
	`Enter the base 64 encoded CA certificate. End with a blank line or the word with "quit" on a line by itself.`
		Requests the application certificate and authenticates it.
		• The certificate request will be displayed on the console terminal so that it may be manually copied (or cut).

Step 7	**Copy the application certificate and paste it on the voice gateway console.**	Manually copy the application certificate text and paste it on the console. • Enter the base 64 encoded CA certificate. End with a blank line or the word "quit" on a line by itself.
Step 8	**`end`** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Exporting Voice Gateway Certificate to the Application

When the voice gateway is behaving as a User Agent Client and requests HTTPS connection to the application, the application requires the voice gateway certificate. You have to export the voice gateway certificate to the application. When application has the voice gateway certificate, it trusts the HTTPS requests coming from the voice gateway and establishes the HTTPS connection.

Note If no trustpoint is configured, voice gateway generates self-signed certificate and uses the same for secure communication. For more information, see the "Configuring a Trustpoint and Specifying Self-Signed Certificate Parameters" section in *Configuring Certificate Enrollment for a PKI*.

Perform this procedure to export voice gateway certificate to the application.

SUMMARY STEPS

Step 1 Execute the following command to see the voice gateway's self-signed certificate:

Example:

```
Device#show crypto pki certificates
Router Self-Signed Certificate
  Status: Available
  Certificate Serial Number (hex): 01
  Certificate Usage: General Purpose
  Issuer:
    cn=IOS-Self-Signed-Certificate-378897163
  Subject:
    Name: IOS-Self-Signed-Certificate-378897163
    cn=IOS-Self-Signed-Certificate-378897163
  Validity Date:
    start date: 12:06:27 IST Jan 18 2017
    end   date: 05:30:00 IST Jan 1 2020
  Associated Trustpoints: TP-self-signed-378897163
Storage: nvram:IOS-Self-Sig#1.cer
```

Voice gateway's self-signed certificate is shown under **`Associated Trustpoints:`**

Step 2 Execute **crypto pki export** *certificate-name* **pem terminal** command to get certificate associated with the trustpoint.

Example:

```
Device(config)#crypto pki export TP-self-signed-378897163 pem terminal
% Self-signed CA certificate:
-----BEGIN CERTIFICATE-----
MIIDLjCCAhagAwIBAgIBATANBgkqhkiG9w0BAQUFADAwMS4wLAYDVQQDEyVJT1Mt
U2VsZi1TaWduZWQtQ2VydGlmaWNhdGUtMzc4ODk3MTYzMB4XDTE3MDExODA2MzYy
N1oXDTIwMDEwMTAwMDAwMFowMDEuMCwGA1UEAxMlSU9TLVNlbGYtU2lnbmVkLUNl
cnRpZmljYXRlLTM3ODg5NzE2MzCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoC
ggEBAI/k+Jl/RdXkUu3aBp8qIMVA7ifpRehG9AXJKlqOafc9Ly92hwNxeLGV/U8k
Xlo/fuoyaNyLIu9GwS1BfvM3yHOthhX+T5RHgcj3s1Yctl6HUW93M/EJYluo5RDE
NAXJ2UXa/Utl9ZGjCvat8h3N4QduP2ulIsK1IqyYLDRwD1fiSNFrdZB2zzIE1M7g
eeitn4n1INHiVtH0jOmO4En/FjUa3YPCFEyB1/U17YGWN/GOHguCsZluL8WywAT5
Pq1uaipVxWoCzXCb74BSxTJiHs/tmPGkIHi57RvLKxgqr5vHXCOsWsQ6/C9z6My3
tvE6dtLHuP2RgR6r+3xOhKdqcHECAwEAAaNTMFEwDwYDVR0TAQH/BAUwAwEB/zAf
BgNVHSMEGDAWgBSIzQOOrJrnxzR8LEQ2VIIfVFpO2DAdBgNVHQ4EFgQUiM0Djqya
58c0fCxENlSCH1RaTtgwDQYJKoZIhvcNAQEFBQADggEBABhYrhWv9DZ0sZZt7Smc
o5pgIIFFOtGQYc+ei7H6QNzW5iNSZbSPBAIpmVMQWHVS6cOvJ/N63ayQ+1TN3rZm
wmOU9tFExBzjge0nX+Go+0KdWNNQG4XO8SU7BKwM8iWTsM1jT1j6cb9Bv1kMgXW0
5K5AzVYTbaTP/OMoMCsuOJts+GI/Q82H7tlIbdJFbbu3iVEN+gf3coUrHa4X2jLr
K3EVLniCLedkcXdy5TppTvQM9j1FzkGMIrWAlFlp/Vh2CTigJy8GZ4pWt5QzjO6m
KuP6FZxGPNe8F5BsFCWNM5aHPa8MUqlFKZMuUb50w43SZRT3xfI2WLv1yd49f65T
mBA=
-----END CERTIFICATE-----
```

```
% General Purpose Certificate:
-----BEGIN CERTIFICATE-----
MIIDLjCCAhagAwIBAgIBATANBgkqhkiG9w0BAQUFADAwMS4wLAYDVQQDEyVJT1Mt
U2VsZi1TaWduZWQtQ2VydGlmaWNhdGUtMzc4ODk3MTYzMB4XDTE3MDExODA2MzYy
N1oXDTIwMDEwMTAwMDAwMFowMDEuMCwGA1UEAxMlSU9TLVNlbGYtU2lnbmVkLUNl
cnRpZmljYXRlLTM3ODg5NzE2MzCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoC
ggEBAI/k+Jl/RdXkUu3aBp8qIMVA7ifpRehG9AXJKlqOafc9Ly92hwNxeLGV/U8k
Xlo/fuoyaNyLIu9GwS1BfvM3yHOthhX+T5RHgcj3s1Yctl6HUW93M/EJYluo5RDE
NAXJ2UXa/Utl9ZGjCvat8h3N4QduP2ulIsK1IqyYLDRwD1fiSNFrdZB2zzIE1M7g
eeitn4n1INHiVtH0jOmO4En/FjUa3YPCFEyB1/U17YGWN/GOHguCsZluL8WywAT5
Pq1uaipVxWoCzXCb74BSxTJiHs/tmPGkIHi57RvLKxgqr5vHXCOsWsQ6/C9z6My3
tvE6dtLHuP2RgR6r+3xOhKdqcHECAwEAAaNTMFEwDwYDVR0TAQH/BAUwAwEB/zAf
BgNVHSMEGDAWgBSIzQOOrJrnxzR8LEQ2VIIfVFpO2DAdBgNVHQ4EFgQUiM0Djqya
58c0fCxENlSCH1RaTtgwDQYJKoZIhvcNAQEFBQADggEBABhYrhWv9DZ0sZZt7Smc
o5pgIIFFOtGQYc+ei7H6QNzW5iNSZbSPBAIpmVMQWHVS6cOvJ/N63ayQ+1TN3rZm
wmOU9tFExBzjge0nX+Go+0KdWNNQG4XO8SU7BKwM8iWTsM1jT1j6cb9Bv1kMgXW0
5K5AzVYTbaTP/OMoMCsuOJts+GI/Q82H7tlIbdJFbbu3iVEN+gf3coUrHa4X2jLr
K3EVLniCLedkcXdy5TppTvQM9j1FzkGMIrWAlFlp/Vh2CTigJy8GZ4pWt5QzjO6m
KuP6FZxGPNe8F5BsFCWNM5aHPa8MUqlFKZMuUb50w43SZRT3xfI2WLv1yd49f65T
mBA=
-----END CERTIFICATE-----
```


Step 3 Copy the self-signed CA certificate displayed on the console and upload it onto the application.

Configuring Cisco Unified Communications Gateway Services in Secure Mode

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip http secure-port** *port*
4. **ip http secure-server**
5. **ip http tls-version** *version*
6. **ip http secure-trustpoint** *name*
7. **ip http max-connection** *value*
8. **ip http timeout-policy idle** *seconds* **life** *seconds* **requests** *value*
9. **http client connection persistent**
10. **http client connection idle timeout** *seconds*
11. **uc secure-wsapi**
12. **message-exchange max-failures** *number*
13. **probing max-failures** *number*
14. **probing interval keepalive** *seconds*
15. **probing interval negative** *seconds*
16. **source-address** *ip-address*
17. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip http secure-port** *port* **Example:** `Device(config)# ip http secure-port 1026`	(Optional) HTTPS connection uses inbound port 443 by default. Execute **ip http secure-port** *port* command if you want to change the default port.
Step 4	**ip http secure-server** **Example:** `Device(config)# ip http secure-server`	Enables the HTTPS server (web server) on the system. **Note** If a certificate authority (CA) is used for certification, you should declare the CA trustpoint on the routing device before enabling the HTTPS server. Executing this command checks if there is any existing trustpoint configured. • If no trustpoint is configured, voice gateway generates self-signed certificate and uses the same for secure communication. For more information, see *Configuring Certificate Enrollment for a PKI*. • If you want to use a specific trustpoint that is already configured on the voice gateway, execute **ip http secure-trustpoint** <*name*> command to specify the trustpoint.
Step 5	**ip http tls-version** *version* **Example:** `Device(config)# ip http tls-version 1.2`	(Optional) By default, all TLS versions (1.1, and 1.2) will be enabled. Configure this command if you want to use only one version of TLS.
Step 6	**ip http secure-trustpoint** *name* **Example:** `Device(config)# ip http secure-trustpoint TP-samplename`	(Optional) Specifies the CA trustpoint that should be used to obtain certificate. • Use of this command assumes you have already declared a CA trustpoint using the **crypto ca trustpoint** command and associated submode commands. • Use the same trustpoint name that you used in the associated crypto ca trustpoint command.

	Command or Action	Purpose
Step 7	**ip http max-connection** *value* **Example:** `Device(config)# ip http max-connection 100`	Sets the maximum number of concurrent connections to the HTTP sever that will be allowed. The default value is 5.
Step 8	**ip http timeout-policy idle** *seconds* **life** *seconds* **requests** *value* **Example:** `Device(config)# ip http timeout-policy idle 600 life 86400 requests 86400`	Sets the characteristics that determine how long a connection to the HTTP server should remain open. The characteristics are: **idle**—The maximum number of seconds the connection will be kept open if no data is received or response data can not be sent out on the connection. Note that a new value may not take effect on any already existing connections. If the server is too busy or the limit on the life time or the number of requests is reached, the connection may be closed sooner. The default value is 180 seconds (3 minutes). **life**—The maximum number of seconds the connection will be kept open, from the time the connection is established. Note that the new value may not take effect on any already existing connections. If the server is too busy or the limit on the idle time or the number of requests is reached, it may close the connection sooner. Also, since the server will not close the connection while actively processing a request, the connection may remain open longer than the specified life time if processing is occurring when the life maximum is reached. In this case, the connection will be closed when processing finishes. The default value is 180 seconds (3 minutes). The maximum value is 86400 seconds (24 hours). **requests**—The maximum limit on the number of requests processed on a persistent connection before it is closed. Note that the new value may not take effect on any already existing connections. If the server is too busy or the limit on the idle time or the life time is reached, the connection may be closed before the maximum number of requests are processed. The default value is 1. The maximum value is 86400.
Step 9	**http client connection persistent** **Example:** `Device(config)# http client connection persistent`	Enables HTTP persistent connection. **Note** When this command is configured, multiple files are loaded using the same connection. Executing this command determines whether the HTTPS client requests a keepalive or closed connection from the server. The HTTPS server is responsible for granting or denying the keepalive connection request from the client.

	Command or Action	Purpose
Step 10	**http client connection idle timeout** *seconds* **Example:** `Device(config)# http client idle timeout 600`	Sets the number of seconds that the client waits in the idle state until it closes the connection.
Step 11	**uc secure-wsapi** **Example:** `Device(config)# uc secure-wsapi`	Enters secure Cisco Unified Communications Gateway Services configuration mode.
Step 12	**message-exchange max-failures** *number* **Example:** `Device(config-uc-wsapi)# message-exchange max failures 2`	Configures the maximum number of failed message exchanges between the application and the provider before the provider stops sending messages to the application. Range is 1 to 3. Default is 1.
Step 13	**probing max-failures** *number* **Example:** `Device(config-uc-wsapi)# probing max-failures 5`	Configures the maximum number of failed probing messages before the voice gateway unregisters the application. Range is 1 to 5. Default is 3.
Step 14	**probing interval keepalive** *seconds* **Example:** `Device(config-uc-wsapi)# probing interval keepalive 180`	Configures the interval between probing messages, in seconds. Default is 120 seconds.
Step 15	**probing interval negative** *seconds* **Example:** `Device(config-uc-wsapi)# probing interval negative 10`	Configures the interval between negative probing messages, in seconds.
Step 16	**source-address** *ip-address* **Example:** `Device(config-uc-wsapi)# source-address 10.25.12.13`	Configures the IP address (hostname) as the source IP address for the Cisco Unified Communications Gateway Services. **Note** The source IP address is used by the provider in the NotifyProviderStatus messages.
Step 17	**end** **Example:** `Device(config-uc-wsapi)# end`	Returns to privileged EXEC mode.

Configuring the XCC Provider on the Voice Gateway

Perform this procedure to configure the XCC provider on the voice gateway.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. Enter Cisco Unified Communications Gateway Services configuration mode:
   a. **uc wsapi**

      or

   b. **uc secure-wsapi**
4. **provider xcc**
5. **no shutdown**
6. **remote-url** *url*
7. If you enable DTMF detection for XCC application and the DTMF method used for the call is **rtp-nte**, then configure the following on outbound dial-peer of CUBE:
   a. **dtmf-relay rtp-nte digit-drop**
   b. **dtmf-interworking standard**
8. **exit**
9. **end**

DETAILED STEPS

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

Step 3	**uc wsapi**
	or
	uc secure-wsapi
	
	Example:
	`Device(config)# uc wsapi`
	
	or
	
	`Device(config)# uc secure-wsapi`
		Enters Cisco Unified Communications Gateway Services configuration mode.

Step 4	**provider xcc** **Example:** `Device(config-uc-wsapi)# provider xcc`	Enters XCC provider configuration mode.
Step 5	**no shutdown** **Example:** `Device(config-uc-wsapi-xcc)# no shutdown`	Activates XCC provider.

	Command or Action	Purpose

Step 6	**`remote-url url`**
	
	Example:
	`Device(config-uc-wsapi-xcc)# remote-url http://192.0.2.0:24/my_callcontrol`
	or
	`Device(config-uc-wsapi-xcc)# remote-url https://192.0.2.0:24/my_callcontrol`
		Specifies the URL (IP address and port number) that the application uses to communicate with XCC provider. The XCC provider uses the IP address and port to authenticate incoming requests.
		
		Note Only IPv4 address is allowed in secure mode (under **uc secure-wsapi** configuration).

Step 7	**`dtmf-relay rtp-nte digit-drop`**
	
	Example:
	`Device(config-uc-wsapi-xcc)# dtmf-relay rtp-nte digit-drop`
	
	`dtmf-interworking standard`
	
	Example:
	`Device(config-uc-wsapi-xcc)# dtmf-interworking standard`
		(Optional) If you enable DTMF detection for XCC application and the DTMF method used for the call is **rtp-nte**, then configure the **`dtmf-relay rtp-nte digit-drop`** and **`dtmf-interworking standard`** commands on the outbound dial-peer of CUBE to avoid any DTMF issues in the RTP data path of the call.
		
		Note The **digit-drop** command is available only when the **rtp-nte** keyword is configured.

Step 8	**`exit`** **Example:** `Device(config-uc-wsapi-xcc)# exit`	Exits XCC configuration mode.
Step 9	**`end`** **Example:** `Device(config-uc-wsapi)# end`	Returns to privileged EXEC mode.

Configuring the XSVC Provider on the Voice Gateway

Perform this procedure to configure the XSVC providers on the voice gateway.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Enter Cisco Unified Communications Gateway Services configuration mode:
   a. **uc wsapi**
   or
   b. **uc secure-wsapi**
4. **provider xsvc**
5. **no shutdown**
6. **remote-url** [*url-number*] *url*
7. **exit**

8. **trunk group** *name*
9. **description**
10. **xsvc**
11. **exit**
12. **voip trunk group** *name*
13. **description**
14. **xsvc**
15. **session target ipv4:***destination-address*
16. **exit**
17. **end**

DETAILED STEPS

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

Step 3	**`uc wsapi`**
	or
	`uc secure-wsapi`
	
	Example:
	`Device(config)# uc wsapi`
	
	or
	
	`Device(config)# uc secure-wsapi`
		Enters Cisco Unified Communications Gateway Services configuration mode.

Step 4	**`provider xsvc`** **Example:** `Device(config-uc-wsapi)# provider xsvc`	Enters XSVC provider configuration mode.
Step 5	**`no shutdown`** **Example:** `Device(config-uc-wsapi-xsvc)# no shutdown`	Activates XSVC provider.

	Command or Action	Purpose

Step 6	**remote-url** [*url-number*] *url*
	
	Example:
	`Device(config-uc-wsapi-xsvc)# remote-url 1 http://192.0.2.0:24/my_route_control`
	or
	`Device(config-uc-wsapi-xsvc)# remote-url 1 https://192.0.2.0:24/my_route_control`
		Specifies up to 8 different URLs (IP address and port number) that applications can use to communicate with the XSVC provider. The XSVC provider uses the IP address and port to authenticate incoming requests.
		
		The *url-number* identifies the unique url. Range is 1 to 8.
		
		Note In secure mode, only one remote URL is allowed. Only IPv4 address can be configured.

Step 7	**exit** **Example:** `Device(config-uc-wsapi-xsvc)# exit`	Exits XSVC configuration mode.
Step 8	**trunk group** *name* **Example:** `Device(config)# trunk group SJ_PRI`	Enters trunk-group configuration mode to define a trunk group.
Step 9	**description** **Example:** `Device(config)# description IN`	Enter a description for the trunk group. The name is passed to external application as part of XSVC status and XCC connection messages.
Step 10	**xsvc** **Example:** `Device(config-trunk-group)# xsvc`	Enables xsvc monitoring on the trunk group.
Step 11	**exit** **Example:** `Device(config-trunk-group)# exit`	Exits trunk group configuration mode.
Step 12	**voip trunk group** *name* **Example:** `Device(config)# trunk group SJ_SIP`	Enters VOIP trunk-group configuration mode to define a trunk group.
Step 13	**description** **Example:** `Device(config-voip-trk-gp)# description IN`	Enter a description for the VOIP trunk group. The name is passed to external application as part of XSVC status and XCC connection messages.
Step 14	**xsvc** **Example:** `Device(config-voip-trk-gp)# xsvc`	Enables xsvc monitoring on the VOIP trunk group.
Step 15	**session target ipv4:***destination address* **Example:** `Device(config-voip-trk-gp)# session target ipv4:9.10.31.254`	Configures the IP address of the remote voice gateway.

	Command or Action	Purpose
Step 16	**exit** **Example:** `Device(config-voip-trk-gp)# exit`	Exits VOIP trunk group configuration mode.
Step 17	**end** **Example:** `Device(config-uc-wsapi)# end`	Returns to privileged EXEC mode.

Configuring the XCDR Provider on the Voice Gateway

Perform this procedure to configure the XCDR provider on the voice gateway.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **uc wsapi**
4. **provider xcdr**
5. **no shutdown**
6. **remote-url** [*url-number*] *url*
7. **exit**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**uc wsapi** **Example:** `Device(config)# uc wsapi`	Enters Cisco Unified Communications Gateway Services configuration mode.
Step 4	**provider xcdr** **Example:** `Device(config-uc-wsapi)# provider xcdr`	Enters XCDR provider configuration mode.

	Command or Action	Purpose
Step 5	**no shutdown** **Example:** `Device(config-uc-wsapi-xcdr)# no shutdown`	Activates XCDR provider.
Step 6	**remote-url** [*url-number*] *url* **Example:** `Device(config-uc-wsapi-xcdr)# remote-url 1 http://209.133.85.47:8090/my_route_control`	Specifies up to eight different URLs (IP address and port number) that applications can use to communicate with the XCDR provider. The XCDR provider uses the IP address and port to authenticate incoming requests. The *url-number* identifies the unique url. Range is 1 to 8.
Step 7	**exit** **Example:** `Device(config-uc-wsapi-xcdr)# exit`	Exits XCDR configuration mode.
Step 8	**end** **Example:** `Device(config-uc-wsapi)# end`	Returns to privileged EXEC mode.

Configuring the XMF Provider on the Voice Gateway

Perform this procedure to configure the XMF provider on the voice gateway.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **uc wsapi**
4. **provider xmf**
5. **no shutdown**
6. **remote-url** *url*
7. **exit**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**uc wsapi** **Example:** `Device(config)# uc wsapi`	Enters Cisco Unified Communications Gateway Services configuration mode.
Step 4	**provider xmf** **Example:** `Device(config-uc-wsapi)# provider xcc`	Enters XMF provider configuration mode.
Step 5	**no shutdown** **Example:** `Device(config-uc-wsapi-xcc)# no shutdown`	Activates XMF provider.
Step 6	**remote-url** *url* **Example:** `Device(config-uc-wsapi-xcc)# remote-url http://209.133.85.47:8090/my_callcontrol`	Specifies the URL (IP address and port number) that the application uses to communicate with XMF provider. The XMF provider uses the IP address and port to authenticate incoming requests.
Step 7	**exit** **Example:** `Device(config-uc-wsapi-xcc)# exit`	Exits XMF configuration mode.
Step 8	**end** **Example:** `Device(config-uc-wsapi)# end`	Returns to privileged EXEC mode.

Configuration Example

The following example sets up the voice gateway for Cisco Unified Communications Gateway Services. It enables the HTTP server and the XCC, XSVC, and XCDR providers. The configuration specifies the address and port that the application uses to communicate with the XCC, XSVC, and XCDR provider. It also identifies the trunk group that XSVC will be monitoring.

Note XSVC and XCDR can support up to eight different remote URLs.

```
ip http server
!
call fallback monitor
call fallback icmp-ping count 1 interval 2 timeout 100
!
uc wsapi
 source-address 10.1.1.1
 provider xcc
  remote-url http://test.com:8090/xcc
 !
 provider xsvc
  remote-url 1 http://test.com:8090/xsvc
!
 provider xcdr
  remote-url 1 http://test.com:8090/xcdr
!
trunk group pri
 xsvc

voip trunk group 1
 xsvc
 session target ipv4: 11.1.1.1
!
interface Serial0/1/0:23
isdn switch-type primary-ni
isdn incoming-voice voice
 trunk-group pri
```

Verifying and Troubleshooting Cisco Unified Communications Gateway Services

Use the following show commands to gather information on the performance of the Cisco Unified Communications Gateway Services:

- **show wsapi registration**
- **show wsapi http client**
- **show wsapi http server**
- **show wsapi xsvc routes**

Use the following debug commands to gather troubleshooting information on the service provider:

- **debug wsapi xcc [CR | all | function | default | detail | error | inout | event]**
- **debug wsapi xsvc [CR | all | function | default | detail | error | inout | event]**
- **debug wsapi xcdr [CR | all | function | default | detail | error | inout | event**]
- **debug wsapi xmf [CR | all | function | default | detail | error | inout | event**]
- **debug wsapi infrastructure [CR | all | function | default | detail | error | inout | event]**

Command Reference

This section documents the CLI commands that are used on the voice gateway.

- message-exchange max-failures
- probing interval
- probing max-failures
- provider
- remote-url
- show call media forking
- show voip trunk group
- show wsapi
- source-address (uc-wsapi)
- uc wsapi
- uc secure-wsapi
- voip trunk group
- xsvc