



CHAPTER 2

Configuring Cisco Unified Communication IOS Services

This chapter contains the following sections:

- [Configuring the Router for Cisco Unified Communication IOS Services, page 2-1](#)
- [Verifying and Troubleshooting Cisco Unified Communication IOS Services, page 2-10](#)
- [Command Reference, page 2-10](#)

Configuring the Router for Cisco Unified Communication IOS Services

This section describes how to configure the router to support the providers on the gateway.

Prerequisite

Cisco IOS Release 15.2(2)T

Configuring Cisco Unified Communication IOS Services on the Router

Perform this procedure to configure Cisco Unified Communication IOS services on the router.

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 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 STEPSⁱ

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ip http server Example: Router(conf)# ip http server	Enables the HTTP server (web server) on the system.
Step 4	ip http max-connection <i>value</i> Example: Router(conf)# 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	<pre>ip http timeout-policy idle seconds life seconds requests value</pre> <p>Example: Router(conf)# ip http timeout-policy idle 600 life 86400 requests 86400</p>	<p>Sets the characteristics that determine how long a connection to the HTTP server should remain open. The characteristics are:</p> <p>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).</p> <p>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).</p> <p>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.</p>
Step 6	<pre>http client persistent</pre> <p>Example: Router(conf)# http client persistent</p>	<p>Enables HTTP persistent connections.</p>
Step 7	<pre>http client connection idle timeout seconds</pre> <p>Example: Router(conf)# http client idle timeout 600</p>	<p>Sets the number of seconds that the client waits in the idle state until it closes the connection.</p>
Step 8	<pre>uc wsapi</pre> <p>Example: Router(conf)# uc wsapi</p>	<p>Enters Cisco Unified Communication IOS Service configuration mode.</p>

	Command or Action	Purpose
Step 9	message-exchange max-failures <i>number</i> Example: Router(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 <i>number</i> Example: Router(config-uc-wsapi)# probing max-failures 5	Configures the maximum number of failed probing messages before the router unregisters the application. Range is 1 to 5. Default is 3.
Step 11	probing interval keepalive <i>seconds</i> Example: Router(config-uc-wsapi)# probing interval 180	Configures the interval between probing messages, in seconds. Default is 120 seconds.
Step 12	probing interval negative <i>seconds</i> Example: Router(config-uc-wsapi)# probing interval negative 10	Configures the interval between negative probing messages, in seconds.
Step 13	source-address <i>ip-address</i> Example: Router(config-uc-wsapi)# source-address 172.1.12.13	Configures the IP address (hostname) as the source IP address for the UC IOS service. Note The source IP address is used by the provider in the NotifyProviderStatus messages.
Step 14	end Example: Router(config-uc-wsapi)# end	Returns to privileged EXEC mode.

Configuring the XCC Provider on the Router

Perform this procedure to configure the XCC provider on the router.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	uc wsapi Example: Router(config)# uc wsapi	Enters Cisco Unified Communication IOS Service configuration mode.
Step 4	provider xcc Example: Router(config-uc-wsapi)# provider xcc	Enters XCC provider configuration mode.
Step 5	no shutdown Example: Router(config-uc-wsapi-xcc)# no shutdown	Activates XCC provider.
Step 6	remote-url url Example: Router(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 XCC provider. The XCC provider uses the IP address and port to authenticate incoming requests.
Step 7	exit Example: Router(config-uc-wsapi-xcc)# exit	Exits XCC configuration mode.
Step 8	end Example: Router(config-uc-wsapi)# end	Returns to privileged EXEC mode.

Configuring the XSVC Provider on the Router

Perform this procedure to configure the XSVC providers on the router.

SUMMARY STEPS

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

3. **uc 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: Router> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	uc wsapi Example: Router(config)# uc wsapi	Enters Cisco Unified Communication IOS Service configuration mode.
Step 4	provider xsvc Example: Router(config-uc-wsapi)# provider xsvc	Enters XSVC provider configuration mode.
Step 5	no shutdown Example: Router(config-uc-wsapi-xsvc)# no shutdown	Activates XSVC provider.

	Command or Action	Purpose
Step 6	remote-url <i>[url-number] url</i> Example: Router(config-uc-wsapi-xsvc)# remote-url 1 http://209.133.85.47:8090/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 <i>url-number</i> identifies the unique url. Range is 1 to 8.
Step 7	exit Example: Router(config-uc-wsapi-xsvc)# exit	Exits Xsvc configuration mode.
Step 8	trunk group <i>name</i> Example: Router(config)# trunk group SJ_PRI	Enters trunk-group configuration mode to define a trunk group.
Step 9	description Example: Router(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: Router(config-trunk-group)# xsvc	Enables xsvc monitoring on the trunk group.
Step 11	exit Example: Router(config-trunk-group)# exit	Exits trunk group configuration mode.
Step 12	voip trunk group <i>name</i> Example: Router(config)# trunk group SJ_SIP	Enters VOIP trunk-group configuration mode to define a trunk group.
Step 13	description Example: Router(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: Router(config-voip-trk-gp)# xsvc	Enables xsvc monitoring on the VOIP trunk group.
Step 15	session target ipv4: <i>destination address</i> Example: Router(config-voip-trk-gp)# session target ipv4:9.10.31.254	Configures the IP address of the remote router.

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

Configuring the XCDR Provider on the Router

Perform this procedure to configure the XCDR provider on the router.

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: Router> enable	Enables privileged EXEC mode. Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	uc wsapi Example: Router(conf)# uc wsapi	Enters Cisco Unified Communication IOS Service configuration mode.
Step 4	provider xcdr Example: Router(config-uc-wsapi)# provider xcdr	Enters XCDR provider configuration mode.

	Command or Action	Purpose
Step 5	<code>no shutdown</code> Example: Router(config-uc-wsapi-xcdr)# no shutdown	Activates XCDR provider.
Step 6	<code>remote-url [url-number] url</code> Example: Router(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 <i>url-number</i> identifies the unique url. Range is 1 to 8.
Step 7	<code>exit</code> Example: Router(config-uc-wsapi-xcdr)# exit	Exits XCDR configuration mode.
Step 8	<code>end</code> Example: Router(config-uc-wsapi)# end	Returns to privileged EXEC mode.

Configuration Example

The following example sets up the router for Cisco Unified Communication IOS 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 Communication IOS Services

Use the following show commands to gather information on the performance of the Cisco Unified Communication IOS 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 infrastructure [CR | all | function | default | detail | error | inout | event]**

Command Reference

This section documents the CLI commands that are used on the router.

- [debug wsapi, page 2-11](#)
- [message-exchange max-failures, page 2-14](#)
- [probing interval, page 2-15](#)
- [probing max-failures, page 2-16](#)
- [provider, page 2-17](#)
- [remote-url, page 2-18](#)
- [show call media forking, page 2-19](#)
- [show voip trunk group, page 2-20](#)
- [show wsapi, page 2-21](#)
- [source-address \(uc-wsapi\), page 2-24](#)
- [uc wsapi, page 2-25](#)
- [voip trunk group, page 2-26](#)
- [xsvc, page 2-27](#)

debug wsapi

To collect and display traces for the Cisco Unified Communication IOS services application programming interface, use the **debug wsapi** command in privileged EXEC mode. To disable debugging, use the **no** form of this command.

```
debug wsapi {infrastructure | xcc | xcdr | xsvc } [all | default | detail | error | event | function | inout | messages]
```

```
no debug wsapi {infrastructure | xcc | xcdr | xsvc } [all | default | detail | error | event | function | inout | messages]
```

Syntax Description

infrastructure	Enables debugging traces on the infrastructure.
xcc	Enables debugging traces on the xcc provider.
xcdr	Enables debugging traces on the xcdr provider.
xsvc	Enables debugging traces on the xsvc provider.
all	Enables all debugging traces.
default	Enables default debugging traces.
detail	Enables detailed debugging traces.
error	Enables error debugging traces.
event	Enables event debugging traces.
function	Enables function debugging traces.
inout	Enables inout debugging traces.
messages	Enables API message traces.

Command Modes

Privileged EXEC

Command History

Release	Modification
15.2(2)T	This command was introduced.

Usage Guidelines

Use this command to enable debugging traces for the Cisco Unified Communication IOS services subsystems.

Examples

The following is the debug output from the **debug wsapi infrastructure** command for an XCC registration.

```
Router# debug wsapi infrastructure
23:25:09: //WSAPI/INFRA/wsapi_https_urlhook:
23:25:09: //WSAPI/INFRA: app_name cisco_xcc in url /cisco_xcc in port 8090
23:25:09: //WSAPI/INFRA/wsapi_https_urlhook: Exit
23:25:09: //WSAPI/INFRA/wsapi_https_post_action:
```

```

23:25:09: wsapi_https_data_read: <soapenv:Envelope
xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope"><soapenv:Body><RequestXccRegister
xmlns="http://www.cisco.com/schema/cisco_xcc/v1_0"><applicationData><name>myapp</name><url
>http://sj22lab-as2:8090/xcc</url></applicationData><blockingEventTimeoutSec>1</blockingEv
entTimeoutSec><blockingTimeoutHandle>CONTINUE_PROCESSING</blockingTimeoutHandle><connectio
nEventsFilter>CREATED AUTHORIZE_CALL REDIRECTED ALERTING CONNECTED TRANSFERRED
CALL_DELIVERY DISCONNECTED HANDOFFLEAVE
HANDOFFJOIN</connectionEventsFilter><mediaEventsFilter>MODE_CHANGE DTMF TONE_BUSY
TONE_DIAL TONE_SECOND_DIAL TONE_RINGBACK TONE_OUT_OF_SERVICE
MEDIA_ACTIVITY</mediaEventsFilter><msgHeader><transactionID>txID001</transactionID></msgHe
ader><providerData><url>http://10.1.1.1:8090/cisco_xcc</url></providerData></RequestXccReg
ister></soapenv:Body></soapenv:Envelope>
23:25:09: //WSAPI/INFRA/27/0/wsapi_https_recv:
23:25:09: //WSAPI/INFRA/27/0/txID001/wsapi_ph_request_msg_handle:
23:25:09: //WSAPI/INFRA/27/0/txID001: prov_type 0 msg_type 6 prov_state 1
23:25:09: //WSAPI/INFRA/wsapi_create_common_msg:
23:25:09: //WSAPI/INFRA/wsapi_create_common_msg: Exit
23:25:09: //WSAPI/INFRA/27/0/txID001/wsapi_send_outbound_response:
23:25:09: wsapi_dump_msg: type 8
23:25:09: transactionID txID001
23:25:09: registrationID 50674FC:XCC:myapp:9
23:25:09: ResponseXccRegister:
23:25:09: providerStatus 1
23:25:09: //WSAPI/INFRA/27/0/txID001/wsapi_send_outbound_response: Exit
23:25:09: wsapi_send_ResponseRegister:mem_mgr_mempool_free: mem_refcnt(3CA18B8)=0 -
mempool cleanup
23:25:09: //WSAPI/INFRA/27/0/txID001/wsapi_https_recv: Exit
23:25:09: wsapi_https_data_write: <?xml version="1.0" encoding="UTF-8"?><SOAP:Envelope
xmlns:SOAP="http://www.w3.org/2003/05/soap-envelope"><SOAP:Body><ResponseXccRegister
xmlns="http://www.cisco.com/schema/cisco_xcc/v1_0"><msgHeader><transactionID>txID001</tran
sactionID><registrationID>50674FC:XCC:myapp:9</registrationID></msgHeader><providerStatus>
IN_SERVICE</providerStatus></ResponseXccRegister></SOAP:Body></SOAP:Envelope>
23:25:09: //WSAPI/INFRA/wsapi_https_post_action: Exit

```

The following is a partial debug log from the **debug wsapi xcc all** command for a call.

```
Router# debug wsapi xcc all
```

```

23:27:20: //WSAPI/XCC/check_xccp_active:177:
23:27:20: //WSAPI/XCC/provider_base_get_state:248:
23:27:20: //WSAPI/XCC/provider_base_get_registration_count:212:
23:27:20: //WSAPI/XCC/check_xccp_active:177:
23:27:20: //WSAPI/XCC/provider_base_get_state:248:
23:27:20: //WSAPI/XCC/provider_base_get_registration_count:212:
23:27:20: //WSAPI/XCC/xccp_sessStore_call_add:271:
23:27:20: //WSAPI/XCC/xccp_sessStore_get_db:145:
23:27:20: //WSAPI/XCC/xccp_session_call_add:353: xcc session successfully added
23:27:20: //WSAPI/XCC/xccp_sessStore_call_add:285: xcc call successfully added
23:27:20: //WSAPI/XCC/check_xccp_active:177:
23:27:20: //WSAPI/XCC/provider_base_get_state:248:
23:27:20: //WSAPI/XCC/provider_base_get_registration_count:212:
23:27:20: //WSAPI/XCC/xccp_create_outbound_msg_space:677:
23:27:20: //WSAPI/XCC/xccp_sessStore_get_callData:225:
23:27:20: //WSAPI/XCC/xccp_sessStore_get_db:145:
23:27:20: //WSAPI/XCC/xccp_session_get_callData:445:
23:27:20: //WSAPI/XCC/check_xccp_active:177:
23:27:20: //WSAPI/XCC/provider_base_get_state:248:
23:27:20: //WSAPI/XCC/provider_base_get_registration_count:212:
23:27:20: //WSAPI/XCC/xccp_notify_events:434:
23:27:20: //WSAPI/XCC/xccp_queue_events:304:
23:27:20: //WSAPI/XCC/provider_base_event_new:335:
23:27:20: //WSAPI/UNKNOWN/event_base_new:267:
23:27:20: //WSAPI/XCC: magic [0xBABE] state[EVENT_STATE_ACTIVE] owner [0x1148C178]
evSize[56] debFlag[3] evHdlr[0x894D834] evHdlFree[0x894DB00]

```

```

23:27:20: //WSAPI/UNKNOWN/event_base_new:292: event base new succ
23:27:20: //WSAPI/XCC/provider_base_event_new:360: provider base eventNew success
23:27:20: //WSAPI/XCC/provider_base_add_ev_to_q:393:
23:27:20: //WSAPI/XCC/check_xccp_active:177:
23:27:20: //WSAPI/XCC/provider_base_get_state:248:
23:27:20: //WSAPI/XCC/provider_base_get_registration_count:212:
23:27:20: //WSAPI/XCC/xccp_create_outbound_msg_space:677:
23:27:20: //WSAPI/XCC/xccp_sessStore_get_callData:225:
23:27:20: //WSAPI/XCC/xccp_sessStore_get_db:145:
23:27:20: //WSAPI/XCC/xccp_session_get_callData:445:
23:27:20: //WSAPI/XCC/check_xccp_active:177:
23:27:20: //WSAPI/XCC/provider_base_get_state:248:
23:27:20: //WSAPI/XCC/provider_base_get_registration_count:212:
23:27:20: //WSAPI/XCC/xccp_solicit_events:359:
23:27:20: //WSAPI/XCC/xccp_queue_events:304:
23:27:20: //WSAPI/XCC/provider_base_event_new:335:
23:27:20: //WSAPI/UNKNOWN/event_base_new:267:
23:27:20: //WSAPI/XCC: magic [0xBABE] state[EVENT_STATE_ACTIVE] owner [0x1148C178]
evSize[56] debFlag[3] evHdlr[0x894D834] evHdlFree[0x894DB00]
23:27:20: //WSAPI/UNKNOWN/event_base_new:292: event base new succ
23:27:20: //WSAPI/XCC/provider_base_event_new:360: provider base eventNew success
23:27:20: //WSAPI/XCC/provider_base_add_ev_to_q:393:
23:27:20: //WSAPI/XCC/provider_base_process_events:444:
23:27:20: //WSAPI/XCC/xccp_handle_events:153:
23:27:20: //WSAPI/INFRA/wsapi_send_outbound_message:
23:27:20: //WSAPI/INFRA/wsapi_send_outbound_message_by_provider_info:
23:27:20: //WSAPI/XCC/wsapi_xcc_encode_outbound_msg:
23:27:20: //WSAPI/XCC/wsapi_xcc_encode_outbound_msg: Exit
23:27:20: //WSAPI/INFRA/0/1527/50875A4:319:out_url http://sj22lab-as2:8090/xcc
23:27:20: wsapi_send_outbound_message_by_provider_info: <?xml version="1.0"
encoding="UTF-8"?><SOAP:Envelope
xmlns:SOAP="http://www.w3.org/2003/05/soap-envelope"><SOAP:Body><NotifyXccConnectionData
xmlns="http://www.cisco.com/schema/cisco_xcc/v1_0"><msgHeader><transactionID>50875A4:319</
transactionID><registrationID>50674FC:XCC:myapp:9</registrationID></msgHeader><callData><c
allID>9</callID><state>ACTIVE</state></callData><connData><connID>1527</connID><state>IDLE
</state></connData><event><created><connDetailData><connData><connID>1527</connID><state>I
DLE</state></connData><guid>7A1E678F-8259-11E0-8FF1-D29982DCA129</guid><callingAddrData><t
ype>E164</type><addr>5522101</addr></callingAddrData><calledAddrData><type>E164</type><add
r>6001</addr></calledAddrData><origCallingAddrData><type>E164</type><addr>5522101</addr></
origCallingAddrData><origCalledAddrData><type>E164</type><addr>6001</addr></origCalledAddr
Data><connIntfType>CONN_SIP</connIntfType><mediaData><type>VOICE</type></mediaData><connIn
tf>1.3.45.2</connIntf><connDirectionType>INCOMING</connDirectionType></connDetailData></cr
eated></event></NotifyXccConnectionData></SOAP:Body></SOAP:Envelope>
23:27:20: //WSAPI/INFRA/0/1527/50875A4:319/wsapi_send_outbound_message_by_provider_info:
Exit
.
.
.

```

message-exchange max-failures

To configure the maximum number of failed message that is exchanged between the application and the provider before the provider stops sending messages to the application, use the **message-exchange max-failures** command. To reset the maximum to the default number, use the **no** form of this command.

message-exchange max-failures *number*

no message-exchange max-failures *number*

Syntax Description	<i>number</i>	Maximum number of messages allowed before the service provider stops sending messages to the application. Range is from 1 to 3. Default is 1.
---------------------------	---------------	---

Command Default	The default is 1.
------------------------	-------------------

Command Modes	uc wsapi configuration mode
----------------------	-----------------------------

Command History	Release	Modification
	15.2(2)T	This command was introduced.

Usage Guidelines	Use this command to set the maximum number of messages that can fail before the system determines that the application is unreachable and the service provider stops sending messages to the application.
-------------------------	---

Examples The following example sets the maximum number of failed messages to 2.

```
Router(config)# uc wsapi
Router(config-uc-wsapi)# message-exchange max-failures 2
```

Related Commands	Command	Description
	probing interval	Sets the time interval between probing messages.
	probing max-failure	Sets the number of messages that the system will send without receiving a reply before the system unregisters the application.

probing interval

To configure the time interval between probing messages sent by the router, use the **probing interval** command. To reset the time interval to the default number, use the **no** form of this command.

probing interval [keepalive | negative] seconds

no probing interval keepalive [negative] seconds

Syntax Description

keepalive	(optional) Configures the time interval between probing messages when the session is in a keepalive state. Range is from 1 to 255 seconds. Default is 5 seconds.
negative	(optional) Configures the time interval between probing messages when the session is in a negative state. Range is from 1 to 20 seconds. Default is 5 seconds.
<i>seconds</i>	Number of seconds between probing message.

Defaults

The default is 120 seconds between probing messages when the session is in a normal state and 5 seconds between probing messages when the session is in a negative state.

Command Modes

uc wsapi configuration mode.

Command History

Release	Modification
15.2(2)T	This command was introduced.

Usage Guidelines

Use this command to configure the time interval between probing messages sent by the router.

Examples

The following example sets an interval of 180 seconds during a normal session and 10 seconds when the session is in a negative state:

```
Router(config)# uc wsapi
Router(config-uc-wsapi)# probing interval keepalive 180
Router(config-uc-wsapi)# probing interval negative 10
```

Related Commands

Command	Description
message-exchange	Sets the maximum number of failed message responses before the provider stops sending messages.
probing max-failure	Sets the number of messages that the system will send without receiving a reply before the system unregisters the application.

probing max-failures

To configure the maximum number of probing messages that the application fails to respond to before the system stops the session and unregisters the application, use the **probing max-failures** command. To reset the maximum to the default number, use the **no** form of this command.

probing max-failures *number*

no probing max-failures *number*

Syntax Description	<i>number</i>	Maximum number of messages allowed before the system stops the session and unregisters the application. Range is from 1 to 5. Default is 3.
---------------------------	---------------	---

Command Default The default is 3.

Command Modes uc wsapi configuration mode.

Command History	Release	Modification
	15.2(2)T	

Usage Guidelines Use this command to set the maximum number of probing messages sent by the system that the application does not respond to before the system stops the session and unregisters the application session.

Examples The following example sets the maximum number of failed messages to 5.

```
Router(config)# uc wsapi
Router(config-uc-wsapi)# probing max-failures 5
```

Related Commands	Command	Description
		message-exchange
	probing interval	Sets the time interval between probing messages.

provider

To configure and enable a service provider, use the **provider** command. To remove the provider, use the **no** form of this command.

provider [XCC | XSVC | XCDR]

no provider [XCC | XSVC | XCDR]

Syntax Description		
XCC	(optional)	Enables the XCC service provider.
XSVC	(optional)	Enables the XSVC service provider.
XCDR	(optional)	Enables the XCDR service provider.

Defaults No default behavior.

Command Modes uc wsapi configuration mode

Command History	Release	Modification
	15.2(2)T	This command was introduced.

Usage Guidelines Use this command to enable the service provider.

Examples The following example enables the XCC service provider.

```
Router(config)# uc wsapi
Router(config-uc-wsapi)# provider xcc
Router(config-uc-wsapi-xcc)# no shutdown
```

Related Commands	Command	Description
	remote-url	Specifies the URL of the application.
	source-address	Specifies the IP address of the provider.
	uc wsapi	Enters Cisco Unified Communication IOS services configuration mode.

remote-url

To configure the url of the application that will be used by the service provider, use the **remote-url** command. The provider will use this url to authenticate and communicate with the application. To delete the configured url, use the **no** form of this command.

remote-url [*url-number*] *url*

no remote-url [*url-number*] *url*

Syntax Description

<i>url-number</i>	(optional) URL number. Range is from 1 to 8.
<i>url</i>	Specifies the URL that the service provider will be using in the messages.

Command Default

None

Command Modes

uc wsapi configuration mode.

Command History

Release	Modification
15.2(2)T	This command was introduced.

Usage Guidelines

Use this command to configure the remote URL (application) that the service provider uses in messages.

Examples

The following example configures the remote url that the the xcc service provider will use in messages.

```
Router(config)# uc wsapi
Router(config-uc-wsapi)# provider xcc
Router(config-uc-wsapi-xcc)# no shutdown
Router(config-uc-wsapi-xcc)# remote-url 1 http://209.133.85.47:8090/my_route_control
```

Related Commands

Command	Description
provider	Enables a provider service.
source-address	Specifies the IP address of the provider.
uc wsapi	Enters Cisco Unified Communication IOS services configuration mode.

show call media forking

To display currently active media forking sessions, use the **show call media forking** command in user EXEC or privileged EXEC mode.

show call media forking

Syntax Description

This command has no arguments or keywords.

Command Modes

User EXEC (>)

Privileged EXEC (#)

Command History

Release	Modification
15.2(2)T	This command was introduced.

Usage Guidelines

Use this command to verify that media forking was successful for relevant anchor legs.

Examples

The following example is a sample output from the **show call media forking** command.

```
Router# show call media forking
```

```
Warning: Output may be truncated if sessions are added/removed concurrently!
```

```
Session    Call    n/f  Destination (port address)
7          6       far  1234 1.5.35.254
8          6       near 5678 1.5.35.254
```

[Table 2-1](#) describes the fields that are displayed.

Table 2-1 Show Call Media Forking Field Descriptions

Field	Description
Session	Session Identifier.
Call	Call Leg identifier in hexadecimal. It must match the Call ID from the show call leg active command.
n/f	Direction (Near End or Far End) of the voice stream that was forked.
Destination (port address)	Destination for the forked packets. It consists of the following: <ul style="list-style-type: none"> RTP Port IP Address

show voip trunk group

To display the internal list of voip trunk groups, use the **show voip trunk group** command in user EXEC or privileged EXEC mode.

show voip trunk group

Syntax Description This command has no arguments or keywords.

Command Modes User EXEC (>
Privileged EXEC (#)

Command History	Release	Modification
	15.2(2)T	This command was introduced.

Usage Guidelines Use this command to display VOIP trunk groups.

Examples The following example is a sample output from the **show voip trunk group** command.

```
Router# show voip trunk group
=====
name:          1
protocol:      cisco
ip:            1.3.45.2
xsvc:         TRUE
```

show wsapi

To display information on the Cisco Unified Communication IOS services, including registration, statistics, and route information, use the **show wsapi** command in user EXEC or privileged EXEC mode.

show wsapi {http-client | http-server | registration {all | xcc | xcdr | xsvc } | xsvc route }

Syntax Description	Option	Description
	http-client	Displays the statistics that have been collected on the http client interface.
	http-server	Displays the statistics that have been collected on the http server interface.
	registration	Displays the currently registered applications on the WSAPI subsystem.
	all	Displays all registered applications.
	xcc	Displays the applications that are registered to the XCC provider.
	xcdr	Displays the applications that are registered to the XCDR provider.
	xsvc	Displays the applications that are registered to the XSVC provider.
	xsvc route	Displays the internal route information in the XSVC provider.

Command Modes	Mode
	User EXEC
	Privileged EXEC

Command History	Release	Modification
	15.2(2)T	This command was introduced.

Usage Guidelines Use this command to display information on the Cisco Unified Communication IOS services.

Examples The following example is a sample output from the **show wsapi http-client** command.

```
Router# show wsapi http-client

WSAPI Outgoing Notify/Solicit Message Statistics
=====

wsapi_show_httpc_callback_context_invalid: 0
wsapi_show_httpc_callback_context_error: 0
wsapi_show_httpc_callback_no_reg: 5
wsapi_show_httpc_callback_notify_OK: 85
wsapi_show_httpc_callback_notify_error: 0
wsapi_show_httpc_callback_client_error: 0
wsapi_show_httpc_callback_error: 7
wsapi_show_httpc_callback_client_error: 0
wsapi_show_httpc_callback_decode_error: 28
wsapi_show_httpc_callback_no_txID: 0
wsapi_show_httpc_callback_OK: 655
wsapi_show_httpc_create_msg_error: 0
wsapi_show_httpc_context_active: 0
wsapi_tx_context_freeq depth: 4
```

The following example is a sample output from the **show wsapi http-server** command.

```
Router# show wsapi http-server

WSAPI Incoming Request Message Statistics
=====

wsapi_show_https_urlhook: 23
wsapi_show_https_post_action: 23
wsapi_show_https_post_action_fail: 0
wsapi_show_https_xml_fault: 0
wsapi_show_https_post_action_done: 23
wsapi_show_https_service_timeout: 0
wsapi_show_https_send_error: 0
wsapi_show_https_invalid_context: 0
wsapi_show_https_data_active: 0
wsapi_https_data_q depth: 1
wsapi_show_https_internal_service_error: 0
wsapi_show_https_service_unavailable_503: 0
wsapi_show_https_not_found_404: 0
wsapi_show_https_registration_success: 9
wsapi_show_https_not_registered: 0
wsapi_show_https_registration_auth_fail: 1
wsapi_show_https_registration_fail: 0
wsapi_show_https_un_registered: 0
```

The following example is a sample output from the **show wsapi registration all** command.

```
Router# show wsapi registration all

Provider XCC
=====
registration
  id: 4FA11CC:XCC:myapp:5
  appUrl:http://sj22lab-as2:8090/xcc
  appName: myapp
  provUrl: http://10.1.1.1:8090/cisco_xcc
  prober state: STEADY
  connEventsFilter:
  CREATED|AUTHORIZE_CALL|ADDRESS_ANALYZE|REDIRECTED|ALERTING|CONNECTED|TRANSFERRED|CALL_DELI
  VERY|DISCONNECTED|HANDOFF_JOIN|HANDOFF_LEAVE
  mediaEventsFilter:
  DTMF|MEDIA_ACTIVITY|MODE_CHANGE||TONE_DIAL|TONE_OUT_OF_SERVICE|TONE_RINGBACK|TONE_SECOND_D
  IAL
  blockingEventTimeoutSec: 1
  blockingTimeoutHandle: CONTINUE_PROCESSING

Provider XSVC
=====
registration index: 2
  id: 4FA0F8C:XSVC:myapp:3
  appUrl:http://sj22lab-as2:8090/xsvc
  appName: myapp
  provUrl: http://10.1.1.1:8090/cisco_xsvc
  prober state: STEADY
  route filter:
  event filter: off

Provider XCDR
=====
registration index: 1
  id: 4FA10A0:XCDR:myapp:1
  appUrl:http://sj22lab-as2:8090/xcdr
```

```

appName: myapp
provUrl: http://10.1.1.1:8090/cisco_xcdr
prober state: STEADY
cdr format: COMPACT
event filter: off

```

The following example is a sample output from the **show wsapi xsvc route** command.

```
Router# show wsapi xsvc route
```

```

Route SANJOSE_SIP
=====
Type: VOIP
Description: OUT
Filter:
Trunk:
    Trunk Name:      1.3.45.2
    Trunk Type:      SIPV2
    Trunk Status:    UP

Route SANJOSE_PRI
=====
Type: PSTN
Description: IN
Filter:
Trunk:
    Trunk Name:      Se0/1/0:23
    Trunk Type:      ISDN PRI
    Trunk Status:    UP
    Total channels   2
    Channel bitmap   0x01FFFFFFE 1-24
    Link bitmap      0x00000006
    Alarm            0x00000001
    Time elapsed     516
    Interval         92
    CurrentData
0 Line Code Violations, 0 Path Code Violations
0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
    TotalData

49 Line Code Violations, 7 Path Code Violations,
0 Slip Secs, 1 Fr Loss Secs, 1 Line Err Secs, 0 Degraded Mins,
0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 2 Unavail Secs

    Trunk Name:      Se0/1/1:23
    Trunk Type:      ISDN PRI
    Trunk Status:    UP
    Total channels   2
    Channel bitmap   0x01FFFFFFE 1-24
    Link bitmap      0x00000006
    Alarm            0x00000001
    Time elapsed     516
    Interval         92
    CurrentData
0 Line Code Violations, 0 Path Code Violations
0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
    TotalData

42 Line Code Violations, 4 Path Code Violations,
0 Slip Secs, 1 Fr Loss Secs, 1 Line Err Secs, 0 Degraded Mins,
0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 2 Unavail Secs

```

source-address (uc-wsapi)

To specify the source IP address or hostname for the Cisco Unified Communication IOS services in the NotifyProviderStatus message, use the **source-address** command in uc wsapi configuration mode. To disable the router from sending NotifyProviderStatus message, use the **no** form of this command.

source-address *ip-address*

no source-address

Syntax Description	<i>ip-address</i>	IP address identified as the source address by the service provider in the NotifyProviderStatus message.
---------------------------	-------------------	--

Defaults	No IP address.
-----------------	----------------

Command Modes	uc wsapi
----------------------	----------

Command History	Release	Modification
	15.2(2)T	This command was introduced.

Usage Guidelines	This command enables the service provider on the router to send messages to the application via the NotifyProviderStatus message.
-------------------------	---

Examples The following example shows how to set the IP source address and port:

```
Router(config)# uc wsapi
Router(config-register-global)# source-address 172.1.12.13
```

Related Commands	Command	Description
	provider	Enables a provider service.
	remote-url	Specifies the URL of the application.
	uc wsapi	Enters Cisco Unified Communication IOS services configuration mode.

uc wsapi

To configure the Cisco Unified Communication IOS services environment for a specific application, use the **uc wsapi** command.

uc wsapi

Syntax Description This command has no arguments or keywords.

Command Default None

Command Modes EXEC mode.

Command History	Release	Modification
	15.2(2)T	This command was introduced.

Usage Guidelines Use this command to enter the Cisco Unified Communication IOS services configuration environment.

Examples The following example enters the Cisco Unified Communication IOS services configuration environment.

```
Router(config)# uc wsapi
Router(config-uc-wsapi)#
```

Related Commands	Command	Description
	provider	Enables a provider service.

voip trunk group

To define or modify a VOIP trunk group and to enter trunk group configuration mode, use the **voip trunk group** command in global configuration mode. To delete the VOIP trunk group, use the **no** form of this command.

voip trunk group *name*

no voip trunk group *name*

Syntax Description	<i>name</i>
	Name of the voip trunk group. Valid names contain a maximum of 63 alphanumeric characters.

Command Default	No voip trunk group is defined.
-----------------	---------------------------------

Command Modes	Global configuration
---------------	----------------------

Command History	Release	Modification
	15.2(2)T	This command was introduced.

Usage Guidelines	Use the voip trunk group command to define the VOIP trunk and extend serviceability to the trunk. By default, the session protocol of the IP trunk is h323. Up to 1000 trunk groups can be configured on the gateway provided that the gateway has sufficient memory to store the profiles
------------------	---

Examples The following example enables creates a VOIP trunk group and enables monitoring.

```
Router(config)# voip trunk group siptrk1
Router(config-voip-trk)# session protocol sipv2
Router(config-voip-trk)# target ipv4: 10.1.1.15
Router(config-voip-trk)# xsvc
```

Command	Description
show voip trunk group	Displays the internal list of voip trunk groups.
xsvc	Enables monitoring on the trunk.

XSVC

To add support for extended serviceability (xsvc) on TDM, (ISDN-PRI/BRI, DS0-group, analog voice-port) voice interfaces, which are defined as a trunk group, use the **xsvc** command. To disable support for extended serviceability, use the **no** form of this command.

xsvc

no xsvc

Syntax Description This command has no arguments or keywords.

Command Default Extended serviceability is disabled on trunk groups.

Command Modes Trunk group configuration

Command History	Release	Modification
	15.2(2)T	This command was introduced.

Usage Guidelines Use this command to add support for extended serviceability on voice interfaces which are defined as a trunk group.

Examples The following example enables monitoring on a trunk group.

```
Router(config)# trunk group tdm-tg1
Router(config-trunk-group)# xsvc
```

Related Commands	Command	Description
	provider	Enables a provider service.

