



CHAPTER 1

Cisco Unified Communication IOS Services API

This chapter describes the Cisco Unified Communication IOS Services Application Programming Interface (CUCISAPI). The CUCISAPI enables the development of advanced Cisco Unified Communication applications and services on the Cisco Integrated Services Router (ISR) by providing an interface to the Cisco Unified Communication IOS services.

CUCISAPI provides the developer with access to the following unified communication IOS services:

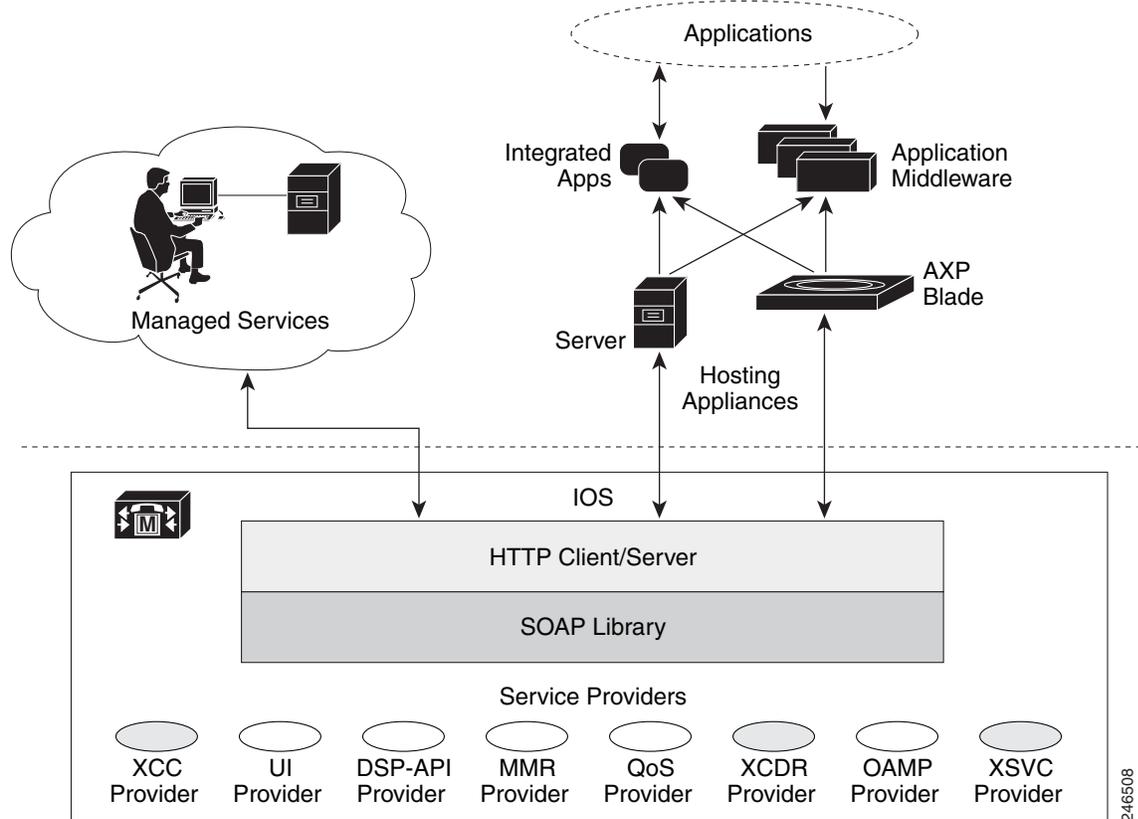
- Extended Call Control Service
- Extended Serviceability Service
- Extended Call Detail Record (CDR) Service

Overview

CUCISAPI allows you to develop an application that interacts with the Cisco Unified Communication IOS services on voice gateways. The application accesses the Cisco Unified Communication IOS services via SOAP messages.

[Figure 1-1](#) illustrates the Cisco Unified Communication IOS service interface. Cisco currently supports the extended call control (XCC) provider, extended call detail record (XCDR) provider, and extended serviceability (XSVC) provider.

Figure 1-1 Cisco Unified Communication IOS Services Interface



Cisco Unified Communication IOS Services

Web service is a standards-based framework that allow applications operating on different platforms to interact over the Internet. Cisco Unified Communication IOS services, like web services, are platform independent and language neutral. With CUCISAPI, you can develop your application in the language and operating system of your choice and communicate directly with the Cisco Unified Communication IOS services running on the voice gateway.

The Cisco Unified Communication IOS Services API supports the following standards and protocol:

- XML 1.0
- Web Services Description Language (WSDL) 1.1
- SOAP, version 1.2
- HTTP, version 1.1

Providers

The providers on the voice gateway provide services on the voice gateway for remote applications. Cisco Unified Communication IOS Services API enables applications to interact with the providers and is comprised of the following provider objects:

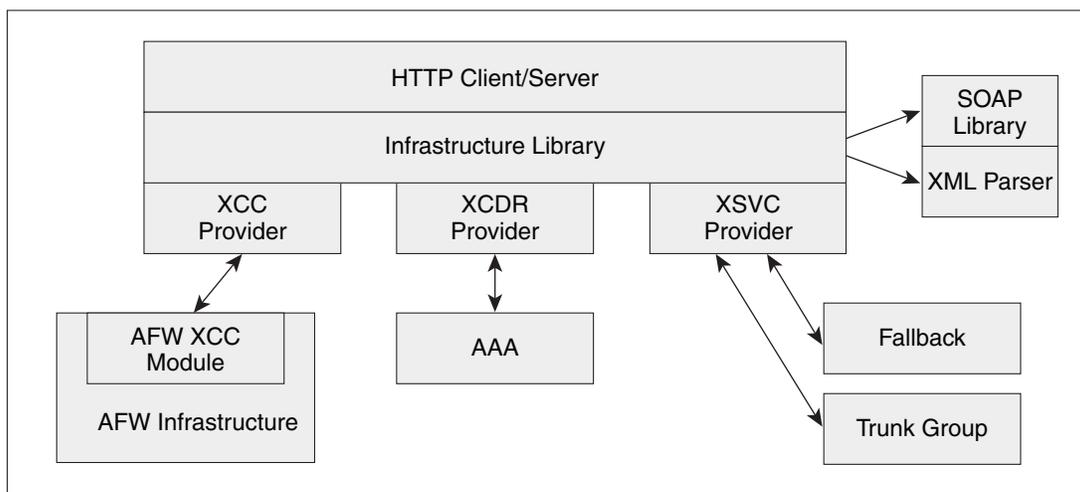
- XCC Provider—Extended Call Control (XCC) provider supports operations that allow an application to perform call control and real-time call monitoring.
- XCDR Provider—Extended Call Detail Record (XCDR) provider supplies CDR information to the application and notifies the application when calls have ended.
- XSVC Provider—Extended Serviceability (XSVC) provider monitors trunk status, and provides real-time link status and configuration change notification to application.

Each provider has a unique URL identifier and communicates with the application via SOAP messages. The providers can be in one of two states:

- In-service—Provider is active and available for use.
- Shutdown—Provider is disabled and no longer available. The API methods associated with this provider are invalid in this state.

Figure 1-2 illustrates the relationship between the IOS components.

Figure 1-2 Cisco Unified Communication IOS Services Components



When a provider is configured and enabled on the voice gateway, it performs the following functions:

- Manages the registration process between the application and the provider.
- Sends notification to the application when a provider changes its status.
- Passes incoming messages to the appropriate provider.
- Notifies the provider when there is a message exchange failure.
- Sends probing messages to maintain an active registration session.
- Sends negative probing messages to detect the status of an application. If the number of failed responses exceeds a configured number of negative probing messages, the voice gateway unregisters the application.

WSDL Files

CUCISAPI uses the WSDL specification to define the services that are available on the voice gateway. These services are represented as providers on the voice gateway.

Table 1-1 lists the namespace for the Cisco Unified Communications IOS services

Table 1-1 Cisco Unified Communication IOS Services Namespace

Service	Location
XCC	http://www.cisco.com/schema/cisco_xcc/v1_0
XCDR	http://www.cisco.com/schema/cisco_xcdr/v1_0
XSVC	http://www.cisco.com/schema/cisco_xsvc/v1_0

Inbound Ports

Table 1-2 lists the URL and inbound location that the application uses to communicate with the server.

Table 1-2 Location of the Inbound Port

Service	Namespace
XCC	http://<access_router>:8090/cisco_xcc ¹
XCDR	http://<access_router>:8090/cisco_xcdr ¹
XSVC	http://<access_router>:8090/cisco_xsvc ¹

1. The access_router is the hostname or IP address of the router that with Cisco Unified Communications IOS services.

Registering an Application

Before an application can register with the voice gateway, you must first configure the application's service URL on the router. The URL is used to authenticate messages from the application. When the router first boots up, the provider sends status messages to the applications that are in its configuration. The router sends status messages when the provider changes its status.

The application initiates registration by sending a registration message to the appropriate provider. The provider generates a unique registration ID and sends it back to the application. The unique registration ID identifies the registered session and is used in all messages that are sent during the registered session.

States of a Registered Session

The state of the registered session and the status of the messages that are sent between the provider and application have one of the following value:

- **Steady State**—This state is the normal state of the registered session. Messages and acknowledgements are exchanged regularly in this state.
- **Keepalive State**—When the provider does not have messages to send, the voice gateway sends keepalive probing messages to the registered application. This keeps the connection between the application and the provider active. The messages that are sent in this state contain information on the health and connectivity status of the provider.

- **Negative Probe State**—When the number of failed responses exceeds the maximum number of failed responses, the registered session enters the negative probe state. In the negative probe state, the voice gateway sends negative-probing messages in an attempt to reestablish the steady state or the keepalive state with the application. The only message sent in a negative probe state is a negative probe message. The registered session returns to a steady state or keepalive state upon receipt of a successful response to a negative probe message, and regular messages resume.
- **Unregistered State**—The session is unregistered and no messages are exchanged between the provider and the application. The session enters an unregistered state under the following conditions:
 - When the application unregisters with the provider
 - When an application fails to respond to probing messages
 - When the administrator shuts down the provider service on the voice gateway

XCC Provider

The XCC provider gives an application the capability to control all the legs of a standard call. With the XCC provider, the application can perform auxiliary call control and can control some network elements.

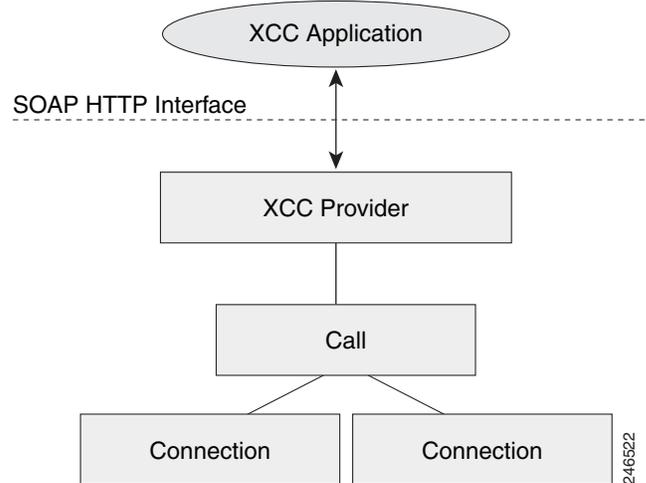
Characteristics of the XCC Provider

The XCC provider has the following characteristics:

- The XCC provider allows the application to maintain stateful control on a call over the entire life cycle of the call.
- The XCC provider allows the application to subscribe and receive mid-call event notification. The application can change event subscription over the life of the call.
- The XCC provider allows services to be invoked on a network triggered event. The provider reports on notifications from a direct application request.
- The XCC provider follows a generic call model in which the underlying communication protocol and architecture is hidden from the developer. XCC provider uses a high-level call control model for maintaining and managing the state of a call session.

Figure 1-3 illustrates the XCC call control abstraction.

Figure 1-3 XCC Call Control



XCC Provider API

When an application registers with the XCC provider, the application configures event filter parameters that the application is interested in monitoring, and the XCC provider installs a connection listener to monitor the calls. XCC notifies the application when a call or connection event matches the event filters that were configured. When the application updates event filter parameters, the updates only apply to new calls, not existing calls.

The XCC provider API is described in [Table 1-3](#). For additional information, see the XCC Provider API.

Table 1-3 XCC Provider API

XCC Provider API	Direction	Description
RequestXccRegister	Application to XCC Provider	Registration request sent with event filter settings for the blocking timeout, connection event, or media filters in the message.
RequestXccUnRegister	Application to XCC Provider	Message sent from the application requesting to be unregistered.
RequestXccControlUpdate	Application to XCC Provider	Message sent with updated connection or media event filters and updated blocking timeout setup.
ResponseXccRegister	XCC provider to the application	Message sent in response to a registration request.
ResponseXccUnRegister	XCC provider to the application	Message sent in response to an unregistration request.
ResponseXccControlUpdate	XCC provider to the application	Message sent in response to a updated event filter request.
NotifyXccStatus	XCC provider to the application	Operation-triggered message sent reporting on the XCC provider status. The following statuses are valid: <ul style="list-style-type: none"> IN_SERVICE SHUTDOWN

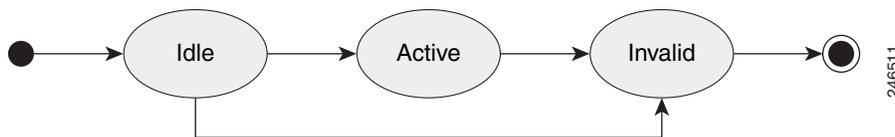
Table 1-3 XCC Provider API

XCC Provider API	Direction	Description
SolicitXccProbing	XCC provider to the application	Blocking message sent to keep the registration session alive and to check on the health of the application.
SolicitXccProviderUnRegister	XCC provider to the application	Blocking message sent to inform the application that the XCC provider has entered the shutdown state and the registration session is now unregistered.
ResponseXccProbing	Application to XCC Provider	Message sent in response to the XCC probing message.
ResponseXccProviderUnRegister	Application to XCC Provider	Message sent in response to the XCC unregistered message.

The XCC call APIs describe the endpoints and trunks that are associated with a call. The APIs in XCC call API, and the associated XCC connection describes the control and media flow in a call. The provider notifies the application when there is a change to the state of a call and sends update information on the call, address, and connections.

A call abstraction is represented in [Figure 1-4](#) on the voice gateway in one of the following three states:

- IDLE—Initial state of all calls. A call in an idle state has zero connections.
- ACTIVE—Call with ongoing activity. A call in an active state has one or more associated connections.
- INVALID—Final state of all calls. Calls that lose all their connections are moved into this state. A call in the invalid state has zero connections.

Figure 1-4 Call Abstraction Model

The XCC call API is described in [Table 1-4](#).

Table 1-4 Xcc Call API

Operation	Direction	Description
RequestXccCallRelease	Application to XCC Provider	Message sent requesting that a call session be released
ResponseXccCallRelease	XCC provider to the application	Message sent in response to the application's call release request.
RequestXccCallMediaForking	Application to XCC Provider	Message sent to enable media forking for the call session.
RequestXccCallMediaSetAttributes	Application to XCC Provider	Message sent to notify the XCC provider that the media attributes for the call session has changed, for example if a call is changed from "voice" to "fax".

Table 1-4 Xcc Call API

Operation	Direction	Description
ResponseXccCallMediaForking	XCC provider to the application	Message sent in response to the application's media forking request.
ResponseXccCallMediaSetAttributes	XCC provider to the application	Message sent in response to the application's media set attribute request.
NotifyXccCallData	XCC provider to the application	Operation-triggered message sent to notify the application when one of the following conditions occurs in a call session: <ul style="list-style-type: none"> • The mode has changed. • DTMF¹ digit was detected. • Media inactive or active is detected.

1. DTMF = dual-tone multi-frequency

XCC Call Media Set Attributes

External applications can enable the voice gateway to detect changes to the call media set attributes on a call and have the voice gateway send a notify event message. Table 1-5 lists the call media set attributes that the gateway can detect.

Table 1-5 Call Media Set Attributes

Call Media Set Attributes	Description
Call Mode Change	Enables the voice gateway to detect when a call changes between the following call modes: <ul style="list-style-type: none"> • Voice Call • Fax Call • Video Call • Modem Call • Data Call <p>Note ISDN calls with an unrestricted bearer capability value are reported as data calls.</p>
DTMF	Enables the voice gateway to detect a DTMF digit in the media stream or a DTMF relay. <p>Note The notify event message includes the timestamp if the DTMF event is detected in IOS.</p> <p>Note For notify event messages, the application should use the voice gateway as the NTP¹ server for synchronizing clocks.</p>
Media Activity	Enables the voice gateway to detect when the media activity state changes from "Active" to "Inactive" or vice versa.

Table 1-5 Call Media Set Attributes

Call Media Set Attributes	Description
Tone	<p>Enables the voice gateway to detect the following specified tones:</p> <ul style="list-style-type: none"> • Busy Tone • Dial Tone • Ringback Tone • Out-of-Service Tone • Second Dial Tone <p>Note Tone detection is not supported for a FXO voice port if the supervisory tone detection feature is enabled.</p>
Media Forking	<p>Enables media forking on a connected call to target a RTP address. For more information on media forking, see the “XCC Call Media Forking” section on page 1-9.</p>

1. NTP = network time protocol.

XCC Call Media Forking

External applications can request media forking for a call. When the application requests media forking, it must provide the XCC provider with two unique remote RTP ports (nearEndAddr and farEndAddr). The XCC provider identifies the incoming connection of a call, forks both the transmit (TX) and receive (RX) packets, and sends the packets to the targeted RTP ports. The XCC provider uses the nearEndAddr element for the forked TX media stream and the farEndAddr XCC element to record the RX media stream. The two forked media streams are sent from the voice gateway in a “SEND ONLY” direction.

Media forking has the following limitations:

- Supports only voice media stream.
- Supports only IPv4 RTP forked media stream.
- Media mixing on forked media streams is not supported.
- Media negotiation is not supported on the forked media streams. In other words, the codec of the forked media stream cannot be changed. If the targeted media server supports a dynamic codec format in the forked media stream, you must configure a supported codec, such as G.711, in the voice gateway.
- Media renegotiation is not supported.
- Media forking ends when the connection is disconnected.
- Supplementary services are not supported.
- Only one media forking request per session is supported. The XCC provider rejects additional media forking request from the application.

The XCC provider updates the application on the status of the media forking by including one of the following states in the NotifyXccCallData message.

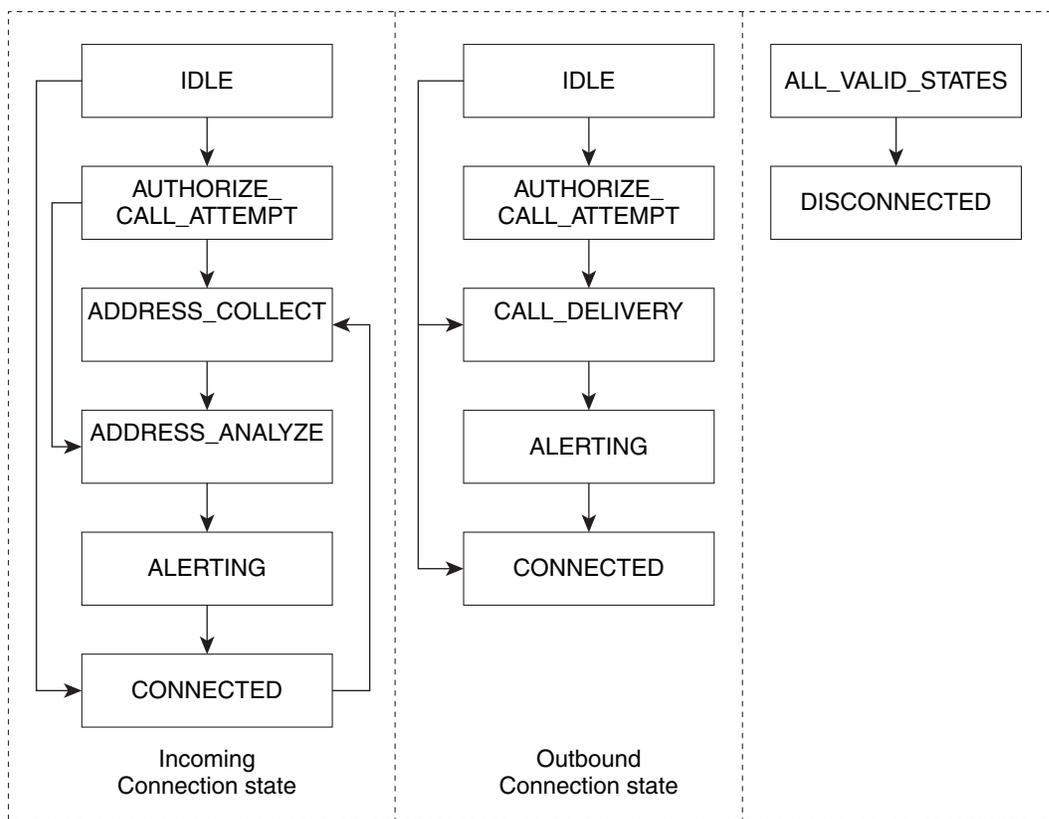
- **FORK_FAILED**—Setup for media forking failed. Forked RTP connections cannot be established with the targeted RTP addresses.

- FORK_STARTED—Media forking was successful. Both the TX and RX forked RTP connections are established and connected to the targeted RTP addresses.
- FORK_DONE—Media forking has completed. Both the TX and RX forked RTP connections are released.

XCC Connection

The XCC connection describes the relationship in a XCC call and the endpoint or trunk in the call. [Figure 1-5](#) illustrates the connection states.

Figure 1-5 Connection States



[Table 1-6](#) describes the connection states and the activity and exchanges that can occur between the voice gateway and application when the application sets up event notifications for a particular connection state.

246516

Table 1-6 Connection States

Connection States	Description	Activity and Messages sent between the Voice Gateway and Application
IDLE	Initial state of all new connections. In this state, the connection is not an active part of the call, but references to the call and address are valid.	<p>Voice Gateway</p> <p>The voice gateway sends a NotifyXccConnectionData(CREATED) message for inbound calls.</p> <p>No messages are sent for outbound calls.</p>
AUTHORIZE_CALL_ATTEMPT	Originating endpoint is waiting for authorization.	<p>Voice Gateway</p> <p>The voice gateway places the call in a suspended state, sends a SolicitXccConnectionAuthorize() message, and waits for a response from the application.</p> <p>Application</p> <p>The application sends the ResponseXccConnectionAuthorize() message directing the gateway to either continue processing or release the call.</p>
ADDRESS_COLLECT	Gateway is collecting information from the originating party.	No messages are sent.
ADDRESS_ANALYZE	Gateway has finished collecting the originating party information and is analyzing and translating the information according to a dial plan.	<p>Voice Gateway</p> <p>The voice gateway places the call in a suspended state, sends a SolicitXccConnectionAddressAnalyze() message, and waits for a response from the application.</p> <p>Application</p> <p>The application sends either the call route back to the gateway or delegates the voice gateway to make the route selection in the ResponseXccConnectionAddressAnalyze() message.</p>
CALL_DELIVERY	On an outbound call, the voice gateway selects the route and sends a request that a call be setup at the specified called endpoint.	<p>No messages are sent for inbound calls.</p> <p>Voice Gateway</p> <p>The voice gateway sends a NotifyXccConnectionData(CREATED) and a NotifyXccConnectionData(CALL) DELIVERY) message for outbound calls.</p>
ALERTING	Endpoint is being notified of the incoming call.	<p>Voice Gateway</p> <p>The voice gateway sends a NotifyXccConnectionData(ALERTING) message.</p>

Table 1-6 Connection States (continued)

Connection States	Description	Activity and Messages sent between the Voice Gateway and Application
CONNECTED	Connection and address for the call active.	Voice Gateway The voice gateway sends a NotifyXccConnectionData (CONNECTED) message.
DISCONNECTED	Connection is no longer active.	Voice Gateway The voice gateway sends a NotifyXccConnectionData(DISCONNECTED) message.

The XCC connection API is described in [Table 1-7](#)

Table 1-7 XCC Connection API

Connection	Direction	Description
RequestXccConnectionRelease	Application to XCC Provider	Message sent requesting that the connection for a call be released.
ResponseXccConnectionRelease	XCC provider to the application	Message sent in response to the application's connection release request.
SolicitXccConnectionAuthorize	XCC provider to the application	Blocking message sent requesting call authorization from the application.
SolicitXccConnectionAddressAnalyze	XCC provider to the application	Blocking message sent with address information for the application to analyze.
ResponseXccConnectionAuthorize	Application to XCC Provider	Message sent in response to the XCC provider's connection authorization request.
RequestXccConnectionAuthorizeDone	Application to XCC Provider	Message sent instructing the XCC provider to either continue processing the call or to release the call.
ResponseXccConnectionAddressAnalyze	Application to XCC Provider	Response message sent instructing the XCC provider to either continue processing the call or to release the call.
RequestXccConnectionAddressAnalyzeDone	Application to XCC Provider	Message sent when the application has completed the address analysis. The message provides information on how the provider should process the call and lists the connection event filters that the application is interested in monitoring.
ResponseXccConnectionAuthorizeDone	XCC provider to the application	Response message sent when the XCC provider has completed the application's XccConnectionAuthorizeDone request.

Table 1-7 XCC Connection API (continued)

Connection	Direction	Description
ResponseXccConnectionAddressAnalyzeDone	XCC provider to the application	Response message sent when the XCC provider has completed the application's XccConnectionAddressAnalyzeDone request.
NotifyXccConnectionData(connection_state)	XCC provider to the application	Operation-triggered message sent to notify the application of the following connection states: <ul style="list-style-type: none"> • CREATED • REDIRECTED • ALERTING • CONNECTED • TRANSFERRING • DISCONNECTED • HANDOFFLEAVE • HANDOFFJOIN

XSVC Provider

The extended serviceability provider (XSVC provider) monitors the health of the trunk and provides the application with real-time trunk status.

The XSVC provider can monitor both traditional public switched telephone network (PSTN) trunks and VoIP trunks. You must configure the XSVC provider and install a route listener for XSVC on the interested trunk group to begin monitoring the trunk status. The route listener communicates with the trunk group resource manager to obtain information on the trunks, including alarm information for T1/E1 trunks.

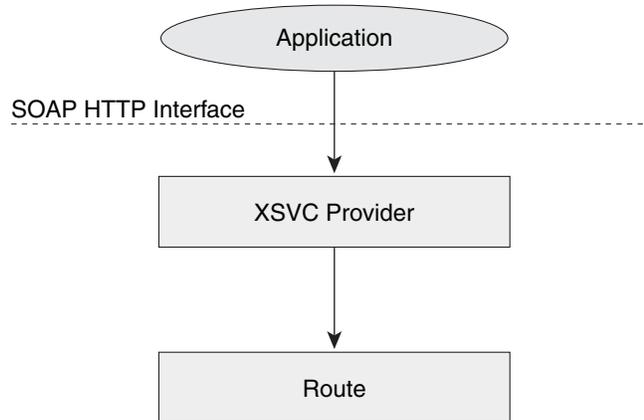
For PSTN trunks, the trunk group is a logical grouping of interfaces with the same signaling characteristics, such as DS1, FXO, or PRI interfaces. The trunk group can have more than one PRI interface and can also support FXO, but you cannot mix FXO and T1/E1 interfaces. The trunk group resource manager supports the logical configuration of trunk groups.

For VoIP trunks, the trunk manager monitors a VoIP trunks by using Internet Control Message Protocol (ICMP) pings. The trunk manager supports up to 1000 trunks.

When the application registers with the XSVC provider, the application obtains a handler that the application uses to receive snapshot information on all the routes or specific routes. The XSVC provider can support up to 8 different applications, with each application able to monitor a particular group of trunks.

Figure 1-6 illustrates the relationship between the application, XSVC route, and XSVC provider.

Figure 1-6 XSVC Provider



246517

Characteristics of the XSVC Provider

The XSVC provider has the following characteristics:

- When the XSVC provider cannot reach the remote application, the XSVC provider discards event information messages.
- The application must register with the XSVC provider or use a snapshot to obtain the most updated trunk information.
- During the registration, the application can configure event filters for a registered session. The event filters only applies for that registered session.
- The XSVC provider reports on the current status of the trunk. The XSVC provider does not report on changes to a trunk configuration until the change has taken effect.

XSVC Provider API

When the application registers with the XSVC provider, a route listener is installed on the trunk interfaces. If filters are not specified in the registration message, the XSVC provider does not filter out any events. For the application to receive the most current trunk configuration, we recommend that you do not filter out the ROUTE_CONF_UPDATED event.

The XSVC provider API is described in [Table 1-8](#). For additional information, see the XSVC Provider API.

Table 1-8 XSVC Provider API

XSVC Provider	Direction	Description
RequestXsvcRegister	Application to XSVC Provider	Registration request sent with event filters settings in the message.
RequestXsvcUnRegister	Application to XSVC Provider	Message sent from the application requesting to be unregistered.

Table 1-8 XSVC Provider API (continued)

XSVC Provider	Direction	Description
ResponseXsvcRegister	XSVC provider to the application	Message sent in response to the application's registration request.
ResponseXsvcUnRegister()	XSVC provider to the application	Message sent in response to the application's unregistration request.
NotifyXsvcStatus	XSVC provider to the application	Operation-triggered message sent to notify the application when the XSVC provider changes state.
SolicitXsvcProbing	XSVC provider to the application	Blocking message sent to keep the registration session alive and to check on the health of the application.
SolicitXsvcProviderUnRegister	XSVC provider to the application	Blocking message sent to inform the application that the XSVC provider has entered the shutdown state and the registration session is now unregistered.
ResponseXsvcProbing	Application to XSVC Provider	Message sent in response to the XSVC probing message.
ResponseXsvcProviderUnRegister	Application to XSVC Provider	Message sent in response to the XSVC unregistered message.

XSVC Route

With the route snapshot API, the application can request and receive a summary from the voice gateway on all the routes that are currently being monitored in a compact format. The application can also set up a filter to listen to specific routes. The application can also request that the XSVC provider send detail information for a specific route. For T1/E1 trunks, the XSVC provider sends additional information, such as channels, total available channels, alarm, and error statistics.

The XSVC Route API is described in [Table 1-9](#).

Table 1-9 XSVC Route API

XSVC Route	Direction	Description
RequestXsvcRouteSetFilter	Application to XSVC Provider	Message specifying the routes that the application is interested in monitoring.
RequestXsvcRouteSnapshot	Application to XSVC Provider	Message requesting compact information on all monitored routes.
RequestXsvcRouteStats	Application to XSVC Provider	Message requesting statistics on specific routes.
RequestXsvcRouteData	Application to XSVC Provider	Message sent requesting detail information on specific routes.
ResponseXsvcRouteSetFilter	XSVC provider to the application	Message sent in response to the application's route filter request message.
ResponseXsvcRouteSnapshot	XSVC provider to the application	Message sent with the compact information (Name and Link information only) on all the routes that are being monitored.
ResponseXsvcRouteStats	XSVC provider to the application	Response message sent with the statistical information on a route.

Table 1-9 X SVC Route API (continued)

X SVC Route	Direction	Description
ResponseXsvcRouteData	X SVC provider to the application	Response message sent with the detailed information on a route.
NotifyXsvcRouteConfiguration	X SVC provider to the application	Operation-triggered message sent when the X SVC option is enabled or disabled on a trunk group, or if the following route configuration changes occur on a trunk group where the X SVC option is enabled: <ul style="list-style-type: none"> • When a new trunk or VoIP trunk is added • When a trunk or VoIP trunk is deleted • When trunks in an existing trunk group are modified • When a trunk or VoIP trunk is modified
NotifyXsvcRouteStatus	X SVC provider to the application	Operation-triggered message sent to notify the application when there is a route status change, for example when the link status changes from UP to DOWN or vice versa. The information sent is in a compact format. Note This event is also triggered when there is a change in the alarm status.

Alarm Definition

Table 1-10 describes the alarm definition that can be found in X SVC route messages.

Table 1-10 Alarm Definition

Alarm	Definition
NoAlarm	No alarm present
RcvFarEndLOF	Far end LOF ¹ indication (a.k.a. Yellow Alarm)
XmtFarEndLOF	Near end sending LOF indication
RcvAIS	Far end sending AIS ²
XmtAIS	Near end sending AIS
LossOfFrame	Near end LOF (a.k.a. Red Alarm)
LossOfSignal	Near end loss of signal
LoopbackState	Near end has a loop back
T16AIS	E1 TS16 AIS
RcvFarEndLOMF	Far end is sending TS16 LOMF ³
RcvFarEndLOMF	Near end is sending TS16 LOMF
RcvTestCode	Near end detects a test code
OtherFailure	Line status that is not defined here
UnavailSigState	Near end is in an unavailable signal state

Table 1-10 Alarm Definition (continued)

Alarm	Definition
NetEquipOOS	Carrier equipment is out of service
RcvPayloadAIS	DS2 payload AIS
Ds2PerfThreshold	DS2 performance threshold

1. LOF = loss of frame.
2. AIS = alarm indication signal.
3. LOMF = loss of multiframe.

Statistics Definition

Table 1-10 defines the statistics that are collected and can be found in XSVC route messages.

Table 1-11 Statistics Definition

Statistics	Definition
LCV	Line Coding Violation Error Event
PCV	Path Coding Violation Error Event
CSS	Controlled Slip Seconds
SEFS	Severely Errored Frame Seconds
LES	Line Errored Seconds
DM	Degraded Minutes
ES	Errored Seconds
BES	Bursty Errored Seconds
SES	Severely Errored Seconds
UAS	Unavailable Seconds

XCDR Provider

The XCDR provider sends information on a call detail record (CDR) to the registered application when a call ends. The CDR contains statistics on the call and calling party and called party information in a CSV format. The XCDR provider can support up to eight remote application.

When the application registers with the XCDR provider, it obtains a handler that the application can use to receive CDR records. The application can choose to receive either the compact or detailed CDR format.

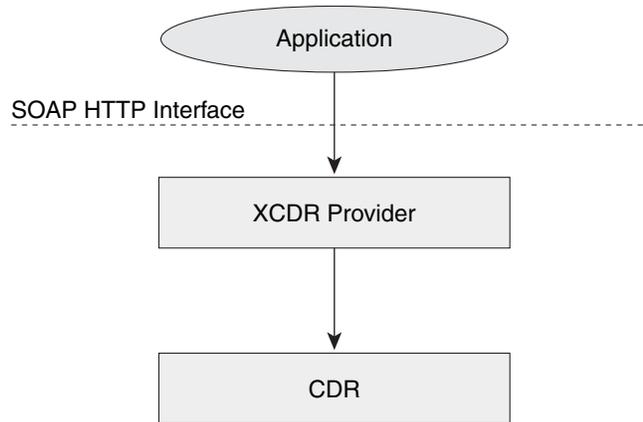


Note

By default, the XCDR provider sends out the CDR record in a compact format to save bandwidth.

Figure 1-7 illustrates the relationship between the application, CDR, and XCDR provider.

Figure 1-7 XCDR



246521

XCDR Provider API

The XCDR provider API is described in [Table 1-12](#). For additional information, see the XCDR provider API.

Table 1-12 XCDR Provider API

XCDR Provider	Direction	Description
RequestXcdrRegister	Application to XCDR Provider	Registration request message. The application can specify whether it wants to receive the route configuration change notification or route status changes.
RequestXcdrUnRegister	Application to XCDR Provider	Unregistration request message sent from the application to the XCDR provider.
ResponseXcdrRegister	XCDR Provider to application	Message sent in response to the application's registration request.
ResponseXcdrUnRegister	XCDR Provider to application	Message sent in response to the application's unregistration request.
NotifyXcdrStatus	XCDR Provider to application	Operation triggered message to notify the application when the XCDR provider changes state.
SolicitXcdrProbing	XCDR Provider to application	Blocking message sent to keep the registration session alive and to check on the health of the application.
SolicitXcdrProviderUnRegister	XCDR Provider to application	Blocking message sent from the voice gateway informing the application that the XCDR provider has entered the shutdown state and the registration session is now unregistered.
ResponseXcdrProbing	Application to XCDR Provider	Message sent in response to the XCDR probing message.
ResponseXcdrProviderUnRegister	Application to XCDR Provider	Message sent in response to the XCDR unregistered message.

XCDR CDR

XCDR CDR is responsible for collecting CDR information and generating events that are sent to the application. The application can specify whether it wants the CDR record in compact or detailed format by using the RequestXcdrSetAttribute message.

The XCDR CDR API is described in [Table 1-12](#).

Table 1-13 XCDR CDR API

XCDR CDR	Direction	Description
RequestXcdrSetAttribute	Application to XCDR Provider	Request message sent to specify the CDR format.
ResponseXcdrSetAttribute	XCDR Provider to application	Message sent in response to the application's CDR format request.
NotifyXcdrRecord	XCDR Provider to application	Message with the Call Detail Record.

Call Detail Record

For detail information on the name and order of the call detail record fields, see [CDR Accounting for Cisco IOS Voice Gateways](#).

Where to Go Next

For more information on the interactions between the providers and the application and examples of messages, see the [“Provider and Application Interactions”](#) section on page A-1

For more information on the elements in the API, see the XCC, XCDR, and XSVC Provider API reference guide.

