The Cisco TelePresence Exchange System is an integrated video service-creation platform that enables service providers and strategic partners to offer secure cloud-based managed and hosted Cisco TelePresence and business video services. The Cisco TelePresence Exchange System is a software environment that simplifies end-to-end subscriber service provisioning; optimizes intelligent call routing for endpoints and network bandwidth; manages the call processing and allocation of media resources for conferencing; consolidates a centralized control point for management, billing, and administration; and exposes a set of application programming interface (API) for integration across business and operational support systems. These standards-based web services APIs facilitate application integration by providing access to functions such as meeting scheduling, management of active meetings, and billing.

This chapter provides a general description of the APIs and includes the following sections:

- API Overview, page 1-1
- Development Guidelines, page 1-2
- About Web Services, page 1-2
- About the API Architecture, page 1-3
- Security, page 1-3
- API Versions, page 1-3
- Error Handling, page 1-3
- Queries, page 1-4
- Required Parameters, page 1-4
- Date/Time Fields, page 1-4
- Pagination, page 1-4
- Common API Methods, page 1-5

**API Overview**

The Cisco TelePresence Exchange System provides the following APIs:

- Scheduling

  The Scheduling API provides web services to control scheduling of services such as Meet-Me and two-party scheduled meetings on the Cisco TelePresence Exchange System.
By using the Scheduling API, you can schedule, modify, or cancel meetings and retrieve information about meetings and other objects required during scheduling.

- **Active Meeting Management**
  The Active Meeting Management API enables real-time management of meetings that are currently in progress. (In contrast, the Scheduling API enables you to schedule and modify future meetings.) By using the Active Meeting Management API, you can develop client applications for monitoring and controlling active meetings, typically by concierge or service desk personnel.

- **Call Detail Record (CDR)**
  The CDR API provides web services to retrieve and manage call detail records for services provided by the Cisco TelePresence Exchange System.


The Cisco TelePresence Exchange System software image includes all components that are required to use the APIs. The APIs require no additional software download or installation.

## Development Guidelines

Cisco requires Cisco Technology Developer Program member applications to be retested and updated as necessary to maintain compatibility with each new major release of Cisco TelePresence Exchange System.

Cisco recommends the following practices to reduce the number and extent of any updates that may be necessary:

- New interface events, methods, responses, headers, parameters, attributes, elements, or new values of existing elements, will most likely be introduced in new versions of the APIs. Each new version of the API includes a separate Web Services Description Language (WSDL). When developers upgrade to the new API version, they may need to provide additional generic or null parameters to existing service methods or call additional methods to achieve the same result.

- Previous interface events, methods, responses, headers, parameters, attributes, and other elements will remain defined in the API, and will appear in the API in the form of separate WSDLs for each supported release.

- Applications must not be dependent on interface behavior that is the result of defects (behavior not consistent with published interface specifications) because the behavior can change when the defect is fixed.

- Cisco recommends that developers have a strategy for migration to newer Cisco TelePresence Exchange System API versions. Developers must note cases in which Cisco removes items such as methods, parameters, responses, and attributes in newer API versions, and ensure that they remove these from their application as soon as possible.

## About Web Services

The web services standards define a framework for clients to request services over a network by using XML-based messaging. Web services operations use an XML-based protocol such as Simple Object Access Protocol (SOAP), which defines the envelope structure, encoding rules, and conventions for representing web service requests and responses. These requests and responses are transmitted as XML-based SOAP messages over HTTP.
Although SOAP messages can be complex, a web services framework hides the complexity from the client developer. A client creates a proxy (a local object that represents the service) and then invokes methods on the proxy. The web services framework converts the API method calls and responses to and from SOAP messages. Cisco recommends that developers use powerful web services frameworks such as Axis for Java developers to simplify development and avoid direct XML document manipulation.

**About the API Architecture**

API clients access the Cisco TelePresence Exchange System APIs by using a standards-based web services infrastructure that is implemented on the administration server. These clients, which can run on different OS platforms, communicate with the administration server by using SOAP-based web services. The CTX API conforms to the SOAP Specification 1.1 and the WSDL Specification 1.1.

The SOAP messages (between client and server) are transported over HTTP to a unique URL that is associated with each of the APIs.

The web services provided by the API are specified by a set of WSDL files. Each web service is defined as a request-response operation (each request results in a correlated response message from the Cisco TelePresence Exchange System). The response message contains information that is relevant to the requested action or data query.

Each request (and the associated response) is a complete transaction. There is no requirement for session or state information to be maintained on the server between requests from a given client.

**Security**

The current API uses HTTP basic access authentication. API clients must include authentication credentials with each API request. The mechanism is HTTP basic access authentication, using Base64 encoding of username and password.

**API Versions**

The Cisco implementation of Cisco TelePresence Exchange System APIs may change over time in response to the evolving needs of our partner community.

The APIs provide a unique URL for each supported version of the API, so that clients can control the timing of their migration to newer versions of the API.

**Error Handling**

The Cisco TelePresence Exchange System API communicates an error condition to the client by returning an exception message instead of a response message. The exception message is an HTTP 500 response that contains a SOAP fault. The fault contains an error code and string field that provide additional details about the exception.
Queries

For services that retrieve information about data objects in the Cisco TelePresence Exchange System (such as endpoints or meetings), the API provides a generalized query mechanism to allow clients to flexibly construct the desired queries. Simple and complex queries are supported. A null query is interpreted as a request to return all of the entities.

Required Parameters

Most API methods have one or more required parameters. When the client provides an empty or null value for any required parameter, the Cisco TelePresence Exchange System will throw a missing-parameter exception, which notes the missing parameter.

Date/Time Fields

In each API, the date and time fields are in ISO 8601 format. Specifically, a calendar date has the following format: YYYY-MM-DD and the time of day employs a 24-hour time period. The letter T is used to separate the date and time fields. The time zone information is represented as an offset to UTC.

For example, an API would store the date of February 11, 2011 and the time of 12:00 PM PDT as follows:

2011-11-02T19:00:00-07:00

Pagination

For many methods where large numbers of records may be returned, you can define pagination parameters to limit the number of records that the Cisco TelePresence Exchange System returns to the API client, to adapt to a web display or a client buffer.

For example, to limit the system to return only 100 records per response to the API client, you would set the numberOfRecords parameter to 100 and set the firstIndex to the following sequence:

firstIndex = 0 for the first group of records, 100 for the second group of records, 200 for the third group of records, and so on for each subsequent group of records.

As long as the Cisco TelePresence Exchange System returns the 100 records in the response as the API client expects, the client will request the next portion of records. When the system returns fewer than 100 records in the response, the client can assume that it has received the last block of records and that no more requests are necessary.
Common API Methods

Each of the Cisco TelePresence Exchange System APIs supports a common set of methods, which are described in the following sections:

- echo, page 1-5
- getVersion, page 1-5

**echo**

The Echo service allows the system to confirm that the requested API service is active. The client includes an arbitrary string in the echo request and the response message includes the same string. Table 1-1 describes the input parameters for the Echo service request.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>echoString</td>
<td>String</td>
<td>Enter an arbitrary string. The same string is returned in the response message.</td>
</tr>
</tbody>
</table>

Table 1-2 describes the parameters in the Echo service response.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>return</td>
<td>String</td>
<td>The value of the string is identical to the string that was sent in the request message.</td>
</tr>
</tbody>
</table>

**getVersion**

The Get Version service returns the software version of the Cisco TelePresence Exchange System. The service request contains no input parameters. Table 1-3 describes the parameters in the service response.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>return</td>
<td>String</td>
<td>The value of the string is the build version of the Cisco TelePresence Exchange System.</td>
</tr>
</tbody>
</table>