



Integration Platform APIs

Technical Overview

WebEx Communications, Inc.
3979 Freedom Circle
Santa Clara, CA 95054

Corp: +1-408-435-7000
Sales: +1.877.50.WEBEX
Web: www.webex.com

Introduction

The WebEx™ Developer Program is designed to support the efforts of partners and end-user customers to integrate a variety of enterprise systems and applications with WebEx web-conferencing services. The core elements of the program include:

- ❑ The WebEx Integration Platform that exposes a set of Application Programming Interfaces (APIs)
- ❑ The WebEx Developer Connection (WDC) extranet containing numerous resources to support developers efforts
- ❑ Dedicated developer support and services staff
- ❑ Partner marketing and sales education programs

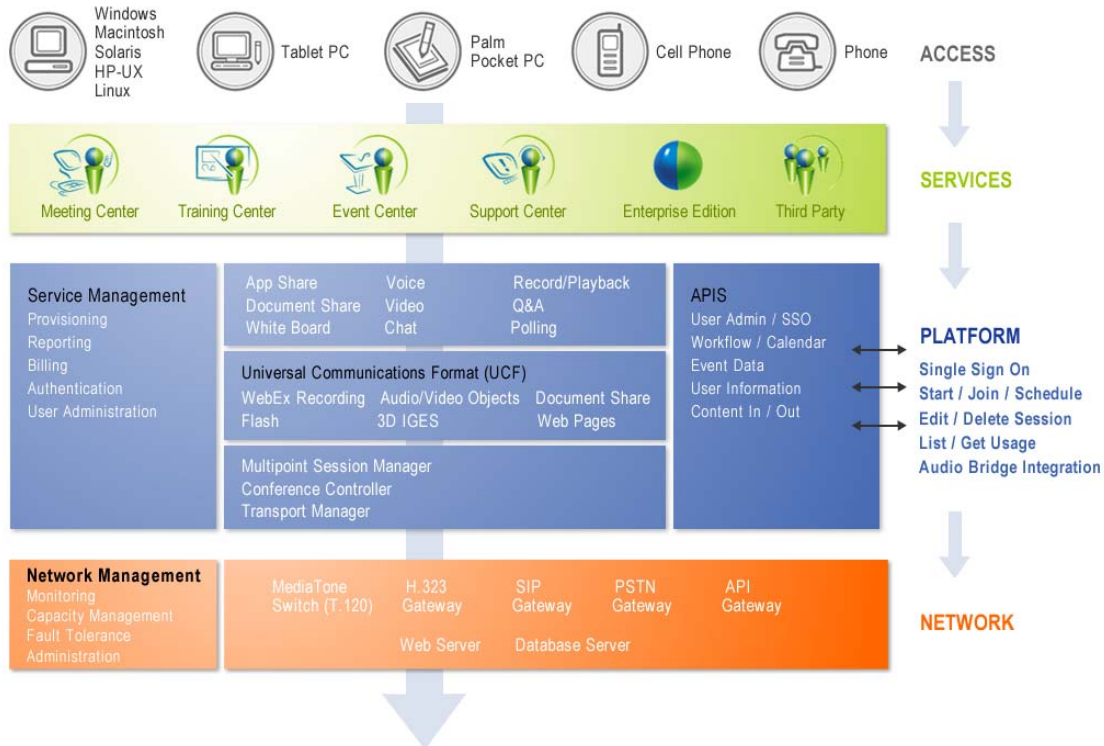
Application partner companies can utilize the WebEx Integration Platform to integrate their enterprise applications and make them “WebEx Enabled”, while system integrators can offer integration services to WebEx customers. Typical enterprise applications benefiting from a tight integration with WebEx include CRM and Sales Force Automation (SFA) systems, Enterprise Marketing Automation (EMA) systems, ERP and HR systems, Learning Management Systems (LMS), Telephony Service Providers, and more. By enabling users of these systems to seamlessly schedule, launch, and join interactive meetings, partners can significantly increase the value of their offering, and can tap into thousands of customers currently using WebEx.

Similarly, regular WebEx end-customers’ IT departments can utilize the Integration Platform to perform in-house integrations with their systems and enterprise applications as well. The WebEx APIs, together with support for voice/data standards and protocols, provide many benefits for corporate customers and end users of WebEx services. Upon integrating their applications with WebEx services, corporations gain powerful web-collaboration capabilities that drive productivity and reduce travel costs. In addition, the ability to integrate WebEx services with corporate accounting, CRM, ERP, human resources and other key applications streamlines maintenance and upkeep. For end users, participating in WebEx online meetings is easy and efficient; simple integrations such as Single Sign-On (SSO) eliminates the need for multiple passwords, and the user can easily move between applications or documents, as each meeting requires.

This document assumes the reader has a basic familiarity with the concept of web conferencing, the WebEx services available for meetings, training, support, events, and so on, and familiarity with application integration based on the use of web services and standards such as XML and HTTP(S).

WebEx Services Architecture

WebEx delivers web-conferencing services through a hosted service offering. The primary service offerings for meetings, training, support, and events sit on top of a dedicated, global, switched data network called MediaTone™. The figure below illustrates the overall architecture and service offerings, including APIs.



The APIs in this architecture provide access to each of the application-level services, as well as to underlying core capabilities for things like user administration, telephony integration, scheduling, and usage reporting.

WebEx Integration Capabilities

The WebEx Integration Platform enables a wide variety of integration capabilities for enterprise customers, enterprise application developers, and telephony service providers. The following is a summary of some of the key integration areas:

User data management:

- Sign up new user (creating user accounts)
- Login/logout
- Activate/deactivate users
- Edit users

Meeting scheduling and registration:

- Schedule/host/join/impromptu
- Edit/delete
- List/add/delete attendees
- Create/get registration form

- Register attendee

Manage and access the history of online sessions:

- List/get usage history
- List recorded access history

Integrate audio conferencing networks:

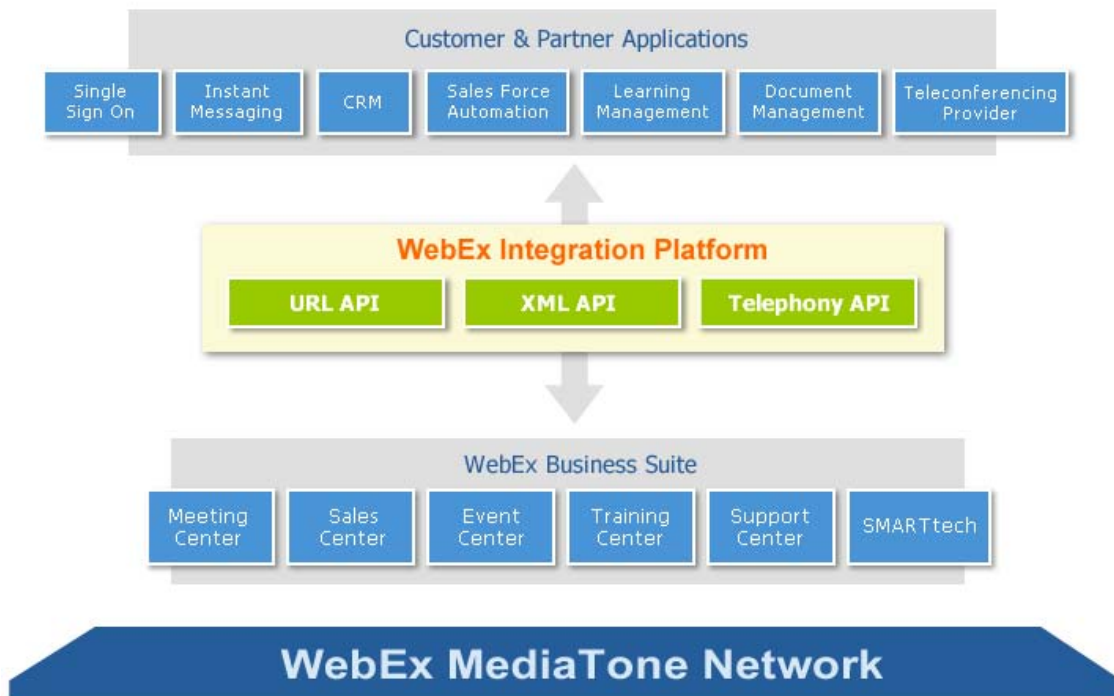
- Synchronize meetings and audio conference network provisioning
- Adaptor-based plug-in architecture for supporting teleconferencing bridge equipment
- Coordinate call-in and call-out options for meeting users, and notification of users entering/exiting the meeting
- Support muting and dropping calls

WebEx API Overview

There are three distinct APIs exposed by the Integration Platform that are available to developers in the WebEx Developer Program.

- ❑ **URL API:** The URL API is HTTP(S) - based, and provides a convenient, lightweight mechanism to provide browser-based, external hooks into the WebEx services. The URL API is typically used in enterprise portal integrations to support basic interactions such as Single Sign-On (SSO), scheduling meetings, starting and joining simple meetings, and inviting attendees and presenters.
- ❑ **XML API:** The XML API is a comprehensive set of services supporting most aspects of the WebEx services, including detailed user management, comprehensive scheduling features, attendee management and reporting, and more. The service responds to XML “request” messages sent via HTTP Post operations from external applications, and returns an XML “response” message back, indicating an action was taken or returning requested data. This is the preferred mechanism for application-level integration. Because data is returned in XML messages, the receiving application is free to parse, utilize, and present the data any way desired; in contrast, for data retrieval commands, the URL API will generally present a WebEx service page back to the user showing the requested information.
- ❑ **Telephony API:** The Telephony API is a full-featured XML-based interface to allow integration with audio conferencing networks. The integration architecture supports redundancy, fail-over, and load balancing, and provides robust features to tightly integrate the audio conferencing capabilities with telephony management and provisioning in the WebEx environment.

The diagram below illustrates the relationship of these APIs with the WebEx application tier and third-party applications that integrate with WebEx application services via the APIs. The remainder of this document will explore each of these interfaces in detail.



URL API

The URL API is ideal for lightweight integration with corporate portals and other web-based applications in the enterprise to support integration of features such as Single Sign-On, and scheduling, starting, or joining meetings. While the URL API has fairly comprehensive support for more advanced activities such as user management, meeting management, reporting, remote access, and so on, WebEx strongly recommends that advanced integrations utilize the XML API which provides more comprehensive feature support, and will have a more capable feature set going forward as new capabilities are added to the underlying services.

The WebEx URL API is based on HTTP URL requests that invoke service requests on the WebEx server. The URLs invocation submits the request to the WebEx server. The request is processed, and returns a web page that is determined by a “Back URL” parameter in the initial request. Every command request in the API specifies a Back URL, which is typically either a specific page served by the WebEx service being invoked, or a page external to WebEx residing, for example, on the corporate portal being integrated.

Command Structure

URL requests are formed by constructing a base URL and appending “AT” commands with parameters specified in the API documentation for each command. The base URL is based on the URL for your WebEx-hosted web site. For example:

```
http://yourWebExHostedName.webex.com
```

In this URL, *yourWebExHostedName* represents the first part of the domain name of your WebEx-hosted Web site, such as your company name, assigned by WebEx. This is the WebEx-hosted Web site to which your server must connect.

For example, if your company name is “XYZ_Corp”, and if WebEx uses XYZ_Corp in your WebEx-hosted site’s server name, your WebEx home page would be:

```
http://XYZ_Corp.webex.com/
```

The URL command set is organized across a few PHP command pages that support certain categories of operations. To invoke a URL command on your organization’s WebEx-hosted Web site, you must append your AT command to the following URL:

```
http://yourWebExHostedName.webex.com/yourWebExHostedName/
```

For the XYZ_Corp company example, the base URL would be:

```
http://XYZ_Corp.webex.com/XYZ_Corp/
```

Next, you append the desired command page and the actual URL command and arguments. Following this example, the command to log out of the WebEx site for the current XYZ_Corp user would be:

```
http://XYZ_Corp.webex.com/XYZ_Corp/p.php?AT=LO
```

In this example, the logout command (AT=LO) is specified from the Partner page (*p.php*), and an optional Back URL is not specified (the default Back URL is the home page of your WebEx-hosted web site).

Typical commands have numerous arguments, some of which are required and some of which are optional. Below is an example of the complete syntax for the Host Meeting command which is supported on the Meetings page (*m.php*). Arguments in brackets are optional:

```
m.php?AT=HM
    &MK=meetingKey
    [&AS=autostartFeature]
    [&AH=windowsAppHandle]
```

```
[ &AL=ApplicationLocation ]
[ &AP=ApplicationParameter ]
[ &DL=DocumentLocation ]
[ &WL=url ]
[ &BU=BackURL ]
```

Log In Requirements

A user normally does not need to log in to join a meeting or event as an attendee, or to join a support session. However, an authorized user must log in to your organization's WebEx-hosted Web site to request any other services. The Partner page (p.php) allows an authorized user to log in. After a user has logged in, an authenticated connection to your WebEx-hosted web site is established and the user's subsequent requests for services will use his or her WebEx identity (WID). The login command (AT=LI) on the Partner page service is commonly used from corporate portals to support Single Sign-On for the users of the portal.

For additional security, WebEx recommends that you use the form POST method to request services—especially for login requests—to avoid sending sensitive information in a URL. Additionally, the secure HTTPS protocol provides added security.

URL API Services

The various service requests available are broken down into specific service areas. These are summarized below:

- **Partner Page (p.php):** This service page allows you to create a new user account on your organization's WebEx-hosted Web site. It also allows an existing authorized user to log in to or out from your WebEx-hosted Web site, and verifies that the URL for the domain from which these calls originate is valid. Additional user and account management requests are also provided. After an authorized user logs in, he or she can access other PHP services that require an authorized user to be logged in.
- **Meeting Page (m.php):** This service allows an authorized meeting host to schedule, delete, edit, or start a meeting (for Meeting Center), a training session (Training Center), an event (Event Center), or a support session (Support Center). Additionally, the page supports requests to list all meetings that a host user has scheduled, or manage attendees for the meeting or session. After an authorized user logs in, he or she can access the Meeting page to request these meeting services.
- **My WebEx and Enterprise Edition Page (o.php):** This service allows a user to access My WebEx and Enterprise Edition site features. Some of these features include modification to a user's My WebEx profile, listing and managing attendees from the user's My Contacts, list and download files from the user's My Files area, and others.
- **Hands-on Lab Page (h.php):** This service allows an authorized Training Center host to list lab names for a site, list the schedule of a lab, get detailed lab information, or check the availability of the computers in a lab.
- **Remote Access Page (c.php):** This service allows a user to get a list of remote access computers (for a My WebEx user's My Computers) and to connect to a remote computer.

Command Summaries

The following sub-sections provide summaries of the available commands for each of the service pages. Detailed information on the command behavior and syntax is available in the URL API documentation.

Partner Page

<u>Command Descriptor</u>	<u>Command</u>	<u>Description</u>
Activate User	AC	Reactivate a deactivated host account

Edit a User	EU	Change an existing user's attributes
Deactivate User	IN	Deactivate a host account
Partner Log In	LI	Log in to your WebEx-hosted web site as an authorized user
Partner Log Out	LO	Log off your WebEx-hosted web site
New User Sign-Up	SU	Create a new user account and set the user's attributes, using an authenticated server-to-server connection

Meeting Page

<u>Command Descriptor</u>	<u>Command</u>	<u>Description</u>
Add Attendees	AA	Add attendees to a scheduled meeting and send email invitations
Add a Presenter	AP	Add a presenter to a scheduled training session
Assist in Support Session	AS	Join a technical support session in Assist Mode by providing the required support session number
Create a Registration Form	CF	Choose which fields are required and which are optional on a meeting's attendee registration form
Add Check Box	CFAC	Add a check box option into a registration form
Add Drop-Down List	CFAD	Add a drop-down list option into a registration form
Add Radio Button	CFAR	Add a option button into a registration form
Add Text Box	CFAT	Add a text box option into a registration form
Create a Recorded Session	CR	Add a previously recorded training session to the list of training sessions available for viewing
Delete an Attendee	DA	Remove one attendee at a time from a scheduled meeting for which the current user is the host
Delete an Event	DE	Delete a scheduled Event Center event
Delete Meeting	DM	Delete a scheduled meeting for which the current user is the host
Delete a Presenter	DP	Delete a presenter from a scheduled training session

<u>Command Descriptor</u>	<u>Command</u>	<u>Description</u>
Edit Event	EE	Change the settings of a scheduled event
Edit a Meeting	EM	Change the attributes of a scheduled meeting for which the current user is the host
Enroll in an Event	EN	Enroll in a scheduled event as an attendee
Get Registration Form	GF	Retrieve the current settings for registration form fields that determine whether the fields are required, optional, or do not display on the registration form
Host a Meeting	HM	Start a scheduled meeting for which the current user is the host
Host Support Session	HS	Start a technical support session that was scheduled with the SK command and for which the current user is the host
Impromptu Meeting	IM	Set up and immediately start a meeting for which the current user is the host
Join Event	JE	Join an event as an attendee by providing the required event number (Meeting Key)
Join Meeting	JM	Join a meeting as an attendee by providing the required meeting number (Meeting Key)
Join a Support Session	JS	Join a technical support session in progress by providing the required support session number
List Meetings	LM	List all scheduled meetings for which the current user is the host
Send Meeting Details	MD	Allows the host of a meeting to send an email message to himself that contains registration information about the meeting's registered attendees
List all Open Meeting	OM	For authorized server-to-server use only. List all meetings that are currently in progress on your WebEx-hosted Web site.
Playback Recorded Event	PR	Get a list of recorded events for a specified attendee
Register for Meeting	RM	Register an attendee for a meeting using some required fields and some fields whose names you can define
Report Attendee Details	RPAD	Report for a list of all events that an attendee has joined

<u>Command Descriptor</u>	<u>Command</u>	<u>Description</u>
Report on Enrollment and Attendees	RPEA	Report for a list of enrollees and attendees for a scheduled event
Report on Recording Playback	RPRP	Report for a list of people who have downloaded files for a specific recorded event
Schedule an Event	SE	Schedule an Event Center event
Schedule Support Session	SK	Schedule a technical support session for which the current user is the host
Schedule Meeting	SM	Schedule a meeting for which the current user is the host
Start Support Session	SS	Start a technical support session immediately
Start a Scheduled Event	TE	Start an event previously scheduled with the SE command

My WebEx and Enterprise Edition Page

<u>Command Descriptor</u>	<u>Command</u>	<u>Description</u>
Add Attendees	AA	Adds one or more contacts to the user's My Contacts.
Delete Attendees	DA	Removes a contact from the user's My Contacts.
Delete a File	DF	Download the specified file from the user's My Files area
Edit Attendees	EA	Edits contact information in the user's My Contacts.
Find Attendees	FA	Allows a user to search for contacts in the user's My Contacts
List Files	LF	Obtain a file list from the current directory of the user's My Files area
My Office	MO	Modify an authorized user's User Profile
Display End-User Report	QR	Request a list of training-session-related events, searching by dates and by topic
Set Service Type	ST	Set the WebEx Service Type for Enterprise Edition

Training Center Hands-On Lab Page

<u>Command Descriptor</u>	<u>Command</u>	<u>Description</u>
Check Availability	CA	Check the availability of the computers in a lab
Get Lab Details	LD	Get detailed information of a lab
List Lab Names	LL	List lab names for a site
List Schedule	LS	List reservation schedule for a lab

Remote Access Page

<u>Command Descriptor</u>	<u>Command</u>	<u>Description</u>
Disconnect Remote Computer	DC	Disconnect from a remote computer
Connect to Remote Computer	RA	Allows the user to connect to a remote computer that the user has set up earlier
Remove Remote Computer	RC	Remove remote computers from the list of available remote computers for your site
List Remote Computers	RL	Display a list of all remote computers that the user is setting up

XML API

The WebEx XML API employs a services oriented architecture (SOA) to provide comprehensive services to external applications wishing to interact with one or more WebEx services. WebEx highly recommends the use of this API when application-level integration that requires a full-featured integration with WebEx functionality is desired. WebEx XML services, such as creating a user, modifying information about a user, creating a meeting, exchanging files, and so on, are implemented by a set of XML operations. For example, a set of meeting operations facilitate specific interactions and the configuration of WebEx online meetings.

XML Server Communications

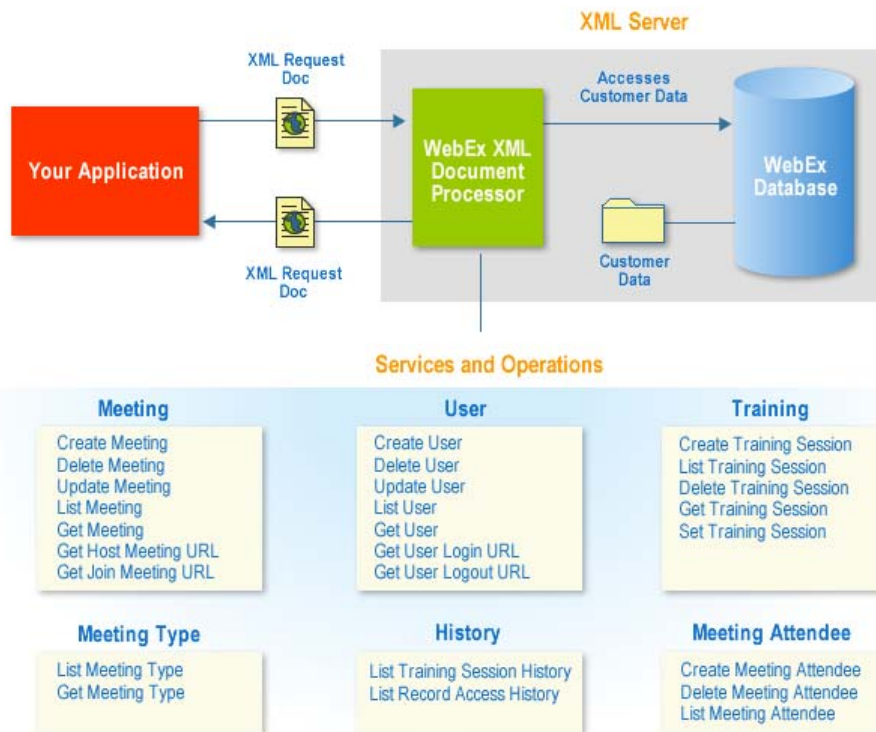
Developers integrate WebEx services through the exchange of well-formed XML documents between enterprise applications and a WebEx XML server. WebEx services are accessed through a protocol of exchanges of XML documents in which an application sends an XML request document to the WebEx XML server, and then processes the received XML response document. WebEx maintains a database of user and site data. External applications and the WebEx XML server exchange this data via XML documents that allow applications to interact with WebEx, no matter how WebEx stores the information internally.

To help ensure that the meanings and the boundaries for each unit of data are well preserved, WebEx standardizes the format of data exchanges between your application and the WebEx XML servers by requiring them to conform to a WebEx-defined XML Schema Definition (XSD), which is the only type of XML document supported in the two-way flow of XML between your application and the WebEx XML servers. Whether an XML expression of a service operation is inbound or outbound, it is always expressed using markup language that conforms to the WebEx XSD. Any information expressed in

elements not declared in the XSD either is ignored by the WebEx XML servers or causes the requested operation to fail.

XML Services Architecture

As shown in the figure below, your application sends a request XML document to perform a specific request—such as creating a user—to the WebEx XML server. This request XML document describes the state of the elements associated with the operation for this request. The WebEx XML server then interacts with the WebEx database and returns the revised element values for this operation to the original application via a response XML document.



The WebEx XML services are predicated on operations involving two primary entities:

- **WebEx Sessions:** Offer interactive comprehensive, multimedia communications capabilities on the Web, so you can effectively interact in a Web-based business meeting. From sharing presentations to sharing any type of document, to Web tours, to full application and desktop remote control, WebEx Meetings offer a spontaneous, dynamic, and rich environment on the Web.
- **Users:** Authorized users who can log in to a WebEx Web service and create meetings.

Each WebEx service typically uses a number of service instances of different types. For example, the features of a meeting are partly configured by one *Meeting* instance, which helps to establish per-meeting preferences; and partly by one *MeetingType* instance, each of which helps establish a well-coordinated set of meeting features. WebEx maintains the *MeetingType* instances as state information for you. Your application's programming, along with your meeting host and meeting proposers, help to establish the per-meeting configuration parameters moderated by the *Meeting* instance.

Examples of WebEx services are scheduling meetings, creating users, and adding attendees. Subscription and Pay-Per-Use meeting services are two distinct types of meeting services that can be represented (and precisely configured) by specific instances of *MeetingType*.

Categories of Request Messages

The WebEx XSD defines many request and response messages with varying constraints. Each message fits into one of the following request categories:

- **Single-item query (getXXX):** A single query resulting in a single item from the WebEx database. For example, `getMeeting`.
- **Multiple-item query (lstsummaryXXX):** Query resulting in a list of items from the WebEx database. For example, `lstsummaryMeeting` and `lstsummarySalesSession`.
- **Add or update (createXXX, setXXX, and deleteXXX):** Creates, updates, or deletes an item in the WebEx database. For example, `setMeeting` changes meeting information in the WebEx database; `createMeetingAttendee` adds a meeting attendee to the WebEx database; and `delMeeting` deletes a meeting from the WebEx database.

WebEx Security Authentication

For every request message sent to the WebEx XML API server, an authentication is conducted as the first step in processing the request. If a request message cannot be authenticated, no further processing occurs. A request message is authorized using the data contained in the security context section of the header. WebEx recommends using HTTPS Post operations to ensure authentication data is securely transmitted.

The WebExID, password, SiteID, and PartnerID are used in combination to validate a request message. You must get these identifiers from WebEx or from your site administrator.

Service Roles

There are three WebEx service roles. After your service request is authenticated to the WebEx server, your WebEx user name is mapped to one of the following three roles.

- **Site Admin:** A site administrator is usually an IT department employee who will administer the sites, set up the site, create user accounts, and assign user privileges.
- **Site Admin—View Only:** A site administrator with view-only privileges can generate reports and view information of the site, but cannot create, delete, or modify user accounts.
- **Hosts:** Hosts can schedule and start meetings. Hosts are usually the presenters as well.

Privileges

There are four classifications of privileges for what each role can do for each XML API operation, as follows:

- **Site Privilege:** A user's ability to access information about all users and meetings on that site.
- **Self Privilege:** A user's ability to view, delete, modify, or create his or her own data, but not the data of others.
- **Public Privilege:** A user's ability to allow others to access the data.
- **No Privilege:** No such privileges.

Example XML Request/Response

The example below shows the XML request and response messages for obtaining the URL for a user to join a meeting. The first segment of the request message is the `securityContext` for authentication purposes. The segment below that is the actual operation request to return the Join URL for an attendee named James Kirk.

Get Host URL Request:

```
<?xml version="1.0" encoding="UTF-8"?>
<serv:message
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <header>
    <securityContext>
      <webExID>hostid</webExID>
      <password>hostpassword</password>
      <siteID>0000</siteID>
      <partnerID>9999</partnerID>
    </securityContext>
  </header>

  <body>
    <bodyContent
xsi:type="java:com.webex.service.binding.meeting.GetJoinur
lMeeting">
      <meetingKey>48591508</meetingKey>
      <attendeeName>James Kirk</attendeeName>
    </bodyContent>
  </body>
</serv:message>
```

The following XML message is the response from the WebEx XML server indicating a successful operation, and returning the requested Join URL:

```
<?xml version="1.0" encoding="UTF-8" ?>
<serv:message
xmlns:serv="http://WebExServiceSchema"
xmlns:com="http://WebExCommonSchema"
xmlns:use="http://WebExUserSchema"
xmlns:att="http://WebExAttendeeSchema">
  <serv:header>
    <serv:response>
      <serv:result>SUCCESS</serv:result>
      <serv:gsbStatus>PRIMARY</serv:gsbStatus>
    </serv:response>
  </serv:header>

  <serv:body>
    <serv:bodyContent
xsi:type="meet:getJoinurlMeetingResponse"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" />
      <meet:joinMeetingURL>https://JoinURL
      <meet:joinMeetingURL>
    </serv:bodyContent>
  </serv:body>
</serv:message>
```

XML API Service Categories

The various operations available are organized into service categories as follows:

- **User Service:** a collection of operations that support interaction with user identities for creation, updating, deletion, and more.
- **Meeting Service:** a collection of operations to manage and configure meetings in Meeting Center.
- **Training Session Service:** a collection of operations to manage and configure training sessions in Training Center.
- **Event Session Service:** a collection of operations to manage and configure events in Event Center.
- **Sales Session Service:** a collection of operations to manage and configure sales sessions in Sales Center.
- **History Service:** a collection of operations for querying the WebEx database to obtain lists of history information across all the services, such as listing event histories and attendee histories.
- **Meeting Attendee Service:** a collection of operations for creating, deleting, and listing attendees.

XML Command Summaries

The following sub-sections summarize the available command operations within each of the available XML services.

User Service Operations

<u>XML Request Message</u>	<u>Description</u>
createUser	Creates a new user account
delUser	Deletes an existing user account
getloginurlUser	Returns the user's login URL
getlogouturlUser	Returns the user's logout URL
getUser	Returns detailed information about a user
lstsummaryUser	Lists summary information of the users
setUser	Updates a user's account information

Message Service Operations

<u>XML Request Message</u>	<u>Description</u>
createMeeting	Creates a meeting
delMeeting	Deletes a meeting
gethosturlMeeting	Returns the host's URL for starting a meeting
getjoinurlMeeting	Returns the URL for attendees to join a meeting
getMeeting	Returns detailed meeting information

<u>XML Request Message</u>	<u>Description</u>
lstsummaryMeeting	Lists summary information on scheduled meetings
setMeeting	Updates a meeting's information

Training Session Services

<u>XML Request Message</u>	<u>Description</u>
checkLabAvailability	Checks availability of lab computers
createTrainingSession	Creates a Training Session
delTrainingSession	Deletes a Training Session
getIMStestDetails	Returns IMS test details
getIMStestResult	Returns IMS test results
getLabInfo	Returns information on the Hand's On Lab
getLabSchedule	Returns reservation schedule for the Hand's On Lab
getTestInformation	Returns test information
getTrainingSession	Returns detailed training session information
lstScheduledTests	Returns a list of scheduled tests
lstsummaryTrainingSession	Returns training session summary information
setTrainingSession	Updates the information for a training session

Event Session Service

<u>XML Request Message</u>	<u>Description</u>
createEvent	Creates a new, scheduled event
delEvent	Deletes a scheduled event
getEvent	Returns information for an event
lstrecordedEvent	Returns a list of previously recorded events
lstsummaryEvent	Returns summary information for scheduled events
setEvent	Updates information for a scheduled event

Sales Session Service

<u>XML Request Message</u>	<u>Description</u>
addProducts	Adds new products to the list of defined products
createSalesSession	Creates a new sales session
getSalesSession	Returns information on a specified sales session
lstProducts	Returns information on all the defined products with matching names
lstsummarySalesSession	Returns summary information on scheduled sales sessions
setProducts	Updates information for existing products

History Service

<u>XML Request Message</u>	<u>Description</u>
lsteventsessionHistory	Returns usage data for previously hosted events
lstsalesessionHistory	Returns usage data for previously hosted sales sessions
lstsupportsessionHistory	Returns usage data for previously hosted support sessions
lsttrainingsessionHistory	Returns usage data for previously hosted training sessions
lsteventattendeeHistory	Returns detailed attendee history for previously hosted events
lstmeetingattendeeHistory	Returns detailed attendee history for previously hosted meetings
lstsalesattendeeHistory	Returns detailed attendee history for previously hosted sales sessions
lstsupportattendeeHistory	Returns detailed attendee history for previously hosted support sessions
lsttrainingattendeeHistory	Returns detailed attendee history for previously hosted training sessions
lstmeetingusageHistory	Returns detailed data for previously hosted meetings
lstrecordaccessHistory	Returns access details for a previously recorded training session

Telephony API

The Telephony API allows conference network vendors to integrate their telephony services with WebEx meeting services to form a seamless audio/visual collaborative environment. WebEx's services support a host of advanced audio control features both during the scheduling and configuration of a meeting, and during the meeting itself.

The Telephony API and associated tools are generally only used by specific teleconferencing service providers that strategically partner with WebEx. The full range of features and details of the Telephony API is beyond the scope of this document. Interested parties should contact the WebEx telecommunications channels team.

Telephony API Feature Summary

A successful teleconferencing integration can exploit a number of WebEx features on both the scheduling and in-meeting sides of the service.

Key scheduling features include:

- Configure sites to be enabled for toll free call-in, toll call-in, domestic call-back, international call-back
- Schedule teleconferences based on the number of attendees per meeting
- Schedule call-in or call-back per meeting
- WebEx email templates and notification is available, but optional, so that you may use your own email templates or email servers
- A teleconference session can be started from the Meeting window by selecting a function from the Participant menu

Some key in-meeting audio features include:

- Toll free call-in and toll call-in numbers are configurable in the Site Administrator screen, and it is visually displayed in the Meeting window
- Within the existing Meeting window, the host or attendee can use the Meeting Manager to perform the following call control capabilities through the API to the adaptor and onto the teleconferencing bridge network:
 - Associate caller with phone icon identifiable by name
 - Mute and mute all function identifiable by a red/green icon
 - Any attendee may select mute
 - Expel attendees
 - Phone out to another attendee during meeting
 - Restrict access

Telephony API Architecture

The P Telephony AI API is based on a plug-in telephony adaptor architecture that allows plug-in adaptors to be developed for new telephony servers without the need to customize underlying code. This allows providers to integrate with existing or new teleconferencing bridge network equipment.

The developer program also makes available standard bridge adaptors that have already been developed to work with particular bridges. Currently, the available adaptors include a Voyant Bridge version 2035 and a Voyant 2313 Adaptor. WebEx develops and maintains the standard adaptors according to priorities. In cases where you have telephony server expertise and resources, you may elect to develop your own adaptor that you will own and maintain.

Some key features of the Telephony API architecture include:

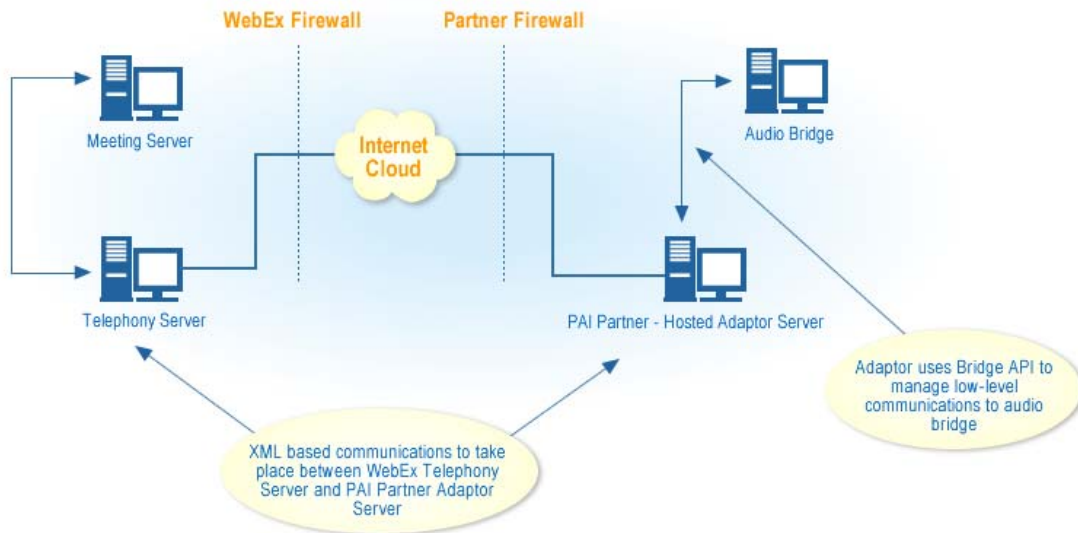
- XML-based protocol with asynchronous request/response behavior
- Employs the SSL (Secure Socket Layer) over IP cloud
- Performance to support up to a 1344 port bridge (one chassis) with 350 simultaneous calls
- Works independently of the WebEx URL API and XML API

- Functions across the public Internet
- Integrates with each user's network elements such as proxy servers, firewalls, security servers, VPN servers, and load balancers
- Supports redundancy and fail-over on both the WebEx side of the interface and the TSP side
- Supports future deployments with dedicated leased lines between a TSP user's network and WebEx's WIN (the WebEx Interaction Network). Dedicated lease line connectivity is considered a custom installation with associated fees.

WebEx servers that provide Web meeting services are hosted by WebEx and located inside the WebEx network. Telephony bridges providing voice conferencing services are hosted by a teleconferencing partner and located inside the teleconferencing partner's network. WebEx servers talk to the bridge through API calls over the internet or a dedicated T1 connection with an HTTP/HTTPS protocol. Each side has their own web server installed to process the calls and requests.

API messages passing between the two sides well-formed XML messages conforming to a WebEx supplied DTD. The API between WebEx and the teleconferencing partner has a generic definition with basic telephony functionalities. Implementations of the functionalities are specific to the kind of bridge used. Thus, an adaptor application (or "adaptor") is required to run between the teleconferencing partner web server and the bridge to convert the generic API calls to the bridge's own API calls.

The diagram below illustrates the high-level architecture for integration using the Telephony API.



Conclusion

The WebEx Integration Platform APIs and associated SDKs provide partners and customers with a powerful set of mechanisms to integrate portals and enterprise applications with WebEx services at a variety of levels. These integrations benefit end-users by creating seamless environments that bring web conferencing functionality to their desktops within the applications and portals they use every day. Partner-level integrations benefit partners by making their applications “WebEx Enabled”, increasing the value proposition they can offer their customers, and creating further leverage through WebEx partnering programs and marketing activities.

Get a start on improving your workflows and application functionality today, and join the rapidly growing WebEx developer community.



Worldwide Sales Offices:

Americas & Canada
Tel: +1.877.50.WEBEX
AmericasInfo@webex.com

Europe, Middle East & Africa
Tel: +31 (0)20.4108.700
EMEAInfo@webex.com

Australia & New Zealand
Tel: +61 (0)3.9653.9581
AsiaPacInfo@webex.com

China (HK)
Tel: +852.8201.0228
AsiaPacInfo@webex.com

Japan
Tel: +81 3 5501 3272
JapanInfo@webex.com

©2004 WebEx Communications, Inc. WebEx, WebEx MediaTone, and the WebEx logo are registered trademarks of WebEx Communications, Inc. All rights reserved. All other trademarks are the property of their respective owners.