Dialogic® PowerMedia™ XMS RESTful API

User's Guide

October 2017 05-2703-014

www.dialogic.com

Copyright and Legal Notice

Table of Contents

Revision History

Revision	Release Date	Notes
05-2703-014	October 2017	Updates to support PowerMedia XMS Release 3.5. RESTful API with HTTP Methods: Updated the Response Codes section. Call Resource: Updated the HTTP POST section. Call Sub-Resource: Updated the Accept/Answer Incoming Call, record_track, record, multi_record, update_multi_record, playrecord, and dial sections. Conference Resource: Updated the HTTP POST section. Conference Sub-Resource: Updated the Modify Conference Attributes, record, multi_record, update_multi_record, and playrecord sections. Events: Updated the end_record and end_playrecord sections and added the MRB Events section. Appendix A: Media File Formats: Updated the audio play and audio record tables. Appendix B: Feature Details: Added the section and reorganized the features. Encryption Record: Added the section.
05-2703-013 (Updated)	June 2017	Call Resource: Updated the Media File Locations section in Call Concepts. Call Sub-Resource: Updated the play_source_attributes, recording_audio_mime_params, recording_video_mime_params, and multi_record sections. Appendix A: Media File Formats: Updated the play and record tables. Automatic Deletion of Silence Recordings: Added the section.
05-2703-013	May 2017	Updates to support PowerMedia XMS Release 3.3. Updated the document for Call Progress Analysis (cpa) profile and operation and Global Unique Session Identifier (gusid) support. RESTful API with HTTP Methods: Added the Response Codes section.
05-2703-012	March 2017	Call Resource: Added the hangup_ack_mode

Revision	Release Date	Notes
(Updated)		parameter in calls. Call Sub-Resource: Added the hangup_ack_mode parameter in Accept/Answer Incoming Call, added the audio_track_id parameter in play_source_attributes, added the Multitrack Play examples, added the primary_video_source parameter in add_party/update_party, and updated the transaction_id parameter in stop. Conference Sub-Resource: Added the primary_video_source parameter in Modify Conference Attributes.
05-2703-012	November 2016	Updates to support PowerMedia XMS Release 3.2. Updated the document for WebM support. Call Sub-Resource: Updated the recording_video_type values. Conference Resource: Added the mixing_mode parameter to support SFU and updated the layout parameter. Conference Sub-Resource: Updated the recording_video_type values. Appendix A: Media File Formats: Updated the Audio Record - MKV, Audio Record - WebM, Video Record - MKV, and Video Record - WebM tables. Enhanced Video Conference Layout Sizing: Added the section. WebM Container: Added the section. Joining Separate Audio and Video Streams: Added the section.
05-2703-011 (Updated)	July 2016	Call Sub-Resource: Updated add_party/update_party, remove_party, and join/union to support multiple joins.
05-2703-011 (Updated)	May 2016	Call Sub-Resource and Video Record: Added how to use the input video stream's resolution and framerate as the target encoding resolution and framerate.
05-2703-011 (Updated)	April 2016	Overview: Updated the RESTful API Request/ Response Model section to state that the request and response payloads are encoded to UTF-8.
05-2703-011	March 2016	Updated the values for recording_audio_type. Call Sub-Resource:

Revision	Release Date	Notes
		• Updated the record examples and added a note to show how to record audio and video on a single file.
05-2703-011	March 2016	Updates to support PowerMedia XMS Release 3.1. Call Resource: • Added a note about the URI parameters for play, record, and playrecord. • Added rtcp_feedback to the Request Payload Attributes table. Call Sub-Resource: • Added rtcp_feedback to the Request Payload Attributes table. Appendix A: Media File Formats: • Added MKV and MP4 Video Play and Video Record capabilities. • Updated the Video Record - Dialogic VID table and the Video Record - 3GP table. Use Case for Dialogic Proprietary Header Tags in a SIP/WebRTC INVITE: • Added the section.
05-2703-010 (Updated)	October 2015	Call Sub-Resource: • Added video/3gpp to the video_type parameter of play_source_attributes. • Updated recording_video_type values. Conference Sub-Resource: • Updated recording_video_type values.
05-2703-010	October 2015	Updates to support PowerMedia XMS Release 3.0. RESTful API Description: • Updated information to include HTTPS. API Resources: • Updated the all resource base URL to include HTTPS. Call Resource: • Updated the Request Payload Attributes table in the HTTP Post section. Call Sub-Resource: • Updated the Request Payload Attributes tables

Revision	Release Date	Notes
		in the HTTP PUT section. • Updated the play_source_attributes, recording_audio_mime_params, recording_video_mime_params, and content_element_attributes tables in the HTTP PUT section. • Removed overlay from the HTTP PUT section. • Added send_event to the HTTP PUT section. • Updated the join/unjoin Request Payload Attributes table in the HTTP PUT section. • Updated the send_info/send_info_ack Request Payload Attributes table in the HTTP PUT section. • Added send_prack/send_prack_ack/ send_answer_ack to the HTTP PUT section. • Added dial to the HTTP PUT section. • Updated the get_call_info Response Payload Attributes table in the HTTP PUT section. Conference Resource: • Updated layout_regions and region_overlays in the Request Payload Attributes table in the HTTP Post section. Conference Sub-Resource: • Updated layout_regions and region_overlays in the Modify Conference Attributes in the HTTP PUT section. • Added playcollect, playrecord, send_dtmf, and send_event to the HTTP PUT section. • Updated the update_play Request Payload Attributes table in the HTTP PUT section. Events: • Added start_play and end_event to Media Events. • Updated ringing in Call Events. • Added info and info_ack sections in Call Events. XML Schema Definition of Elements: • Updated the XML definition of elements. Dynamic Text and Image Generation: • Removed section.

Revision	Release Date	Notes
		Text and Image Overlays: • Added section. Appendix A: Media File Formats: • Updated the media file formats.
05-2703-009 (Updated)	June 2015	Call Resource: • Added local_rtp_address and headers to the Request Payload Attributes table for HTTP POST. Call Sub-Resource: • Added local_rtp_address to the Request Payload Attributes table for HTTP PUT.
05-2703-009 (Updated)	April 2015	Call Resource: • Added audio, video, message, setup, content_type, and content to the Request Payload Attributes table for HTTP POST. Call Sub-Resource: • Updated send_message section. Appendix A: Media File Formats: • Updated the Video Play - Dialogic VID (proprietary) table in Media File Formats section to correct RESTful attribute format for jpeg codec.
05-2703-009	February 2015	Updates to support PowerMedia XMS Release 2.4. Call Sub-Resource: • Updated send_info/send_info_ack section to change content_data to content.
05-2703-008	January 2015	Updates to all the resource response payload example sections for IPv6 support. Call Sub-Resource: • Added note to mode parameter in add_party/update_party section. Conference Resource: • Updated the value auto to automatic for layout_size parameter. Conference Sub-Resource: • Updated the value auto to automatic for layout_size parameter.

Revision	Release Date	Notes
		Event Handler Resource: • Added RESTful Event Streaming Data Format Change section. Events: • Updated end_play, end_playcollect, end_record, and end_playrecord sections in Media Events. • Updated end_recognize section in MRCP Events. • Updated sections to change call_id to id in MRCP Events. Appendix A: Media File Formats: • Updated the Video Record table in Media File Formats section.
05-2703-007	October 2014	Updates to support PowerMedia XMS Release 2.3. Call Sub-Resource: • Added add_ice_candidate, send_hangup_ack, and send_message sections. • Added msg_payload attributes, msg_payload_content attribute, and msg_payload_uri attributes tables. • Added content_element parameter to send_message section and added content_element_attributes table. • Added get_last_action and get_last_event sections. Conference Sub-Resource: • Added get_last_action and get_last_event sections. MRCP Sub-Resource: • Added get_last_action and get_last_event sections. Event Handler Resource: • Added speech_marker value to type parameter. Event Handler Sub-Resource: • Added speech_marker value to type parameter. Events: • Added end_send_message section in Media

Revision	Release Date	Notes
		Events. • Updated end_play, end_playcollect, and end_record sections in Media Events. • Added message, alarm, stream, prack, accepted, and answered sections in Call Events. • Updated dtmf and tone sections in Call Events. • Added speech_marker section in MRCP Events. • Added conf_overlay_expired section in Conference Events. XML Schema Definition of Elements: • Added updated schema definition. Appendix A: Media File Formats: • Added new section.
05-2703-006	May 2014	Updates to dtmf event in Call Events section.
05-2703-005	April 2014	Updates to support PowerMedia XMS Release 2.2.
05-2703-004	October 2013	Updates to support PowerMedia XMS Release 2.1. Global change: • Renamed this document from Developer's Guide to User's Guide. • Reorganized the resource-based elements and payload attributes tables. XML Schema Definition of Elements: • Added updated schema definition.
05-2703-003	January 2013	Updates to support PowerMedia XMS Release 2.0. Call Resource: • Updated http post/put request payload, single call instance response payload, call_action, add_party, and update_party in XML Schema Definitions for Call. • Added new clamp_dtmf, auto_gain_control, echo_cancellation, digits, interval, level, content_type, and content to the <call_action> Parameters table. • Added new send_dtmf, send_info, send_info_ack, transfer, redirect, and hangup

Revision	Release Date	Notes
		to the <call_action> Parameters table. • Added new display_name, accept, early_media, and info_ack_mode to the <call> Element Attributes table. • Added new section for <call> Element Attribute Notes. Conference Resource: • Added new playrecord conference action in XML Schema Definition for Conference. • Added new barge, cleardigits, beep, and recording_uri to the <conf_action> Parameters table. XML Schema Definition of Elements: • Added updated schema definition.
05-2703-002	July 2012	Updated to support PowerMedia XMS Release 1.1. This includes adding additional information to all sections and reorganizing the layout of the Resource sections. Global change: • Renamed PowerMedia XMS RESTful web service to PowerMedia XMS RESTful server. Call Resource: • Added new interdigit_timeout parameter to the <call_action> Parameters table. Conference Resource: • Added new region parameter to the <conf_action> Parameters table. XML Schema Definition of Elements: • Added updated schema definition. Dynamic Text and Image Generation: • Added new section. XMSTool RESTful Utility: • Added new section.
05-2703-001	February 2012	Initial release of this document.
Last modified: October 2017		

Refer to www.dialogic.com for product updates and for information about support policies, warranty information, and service offerings.

1. Welcome

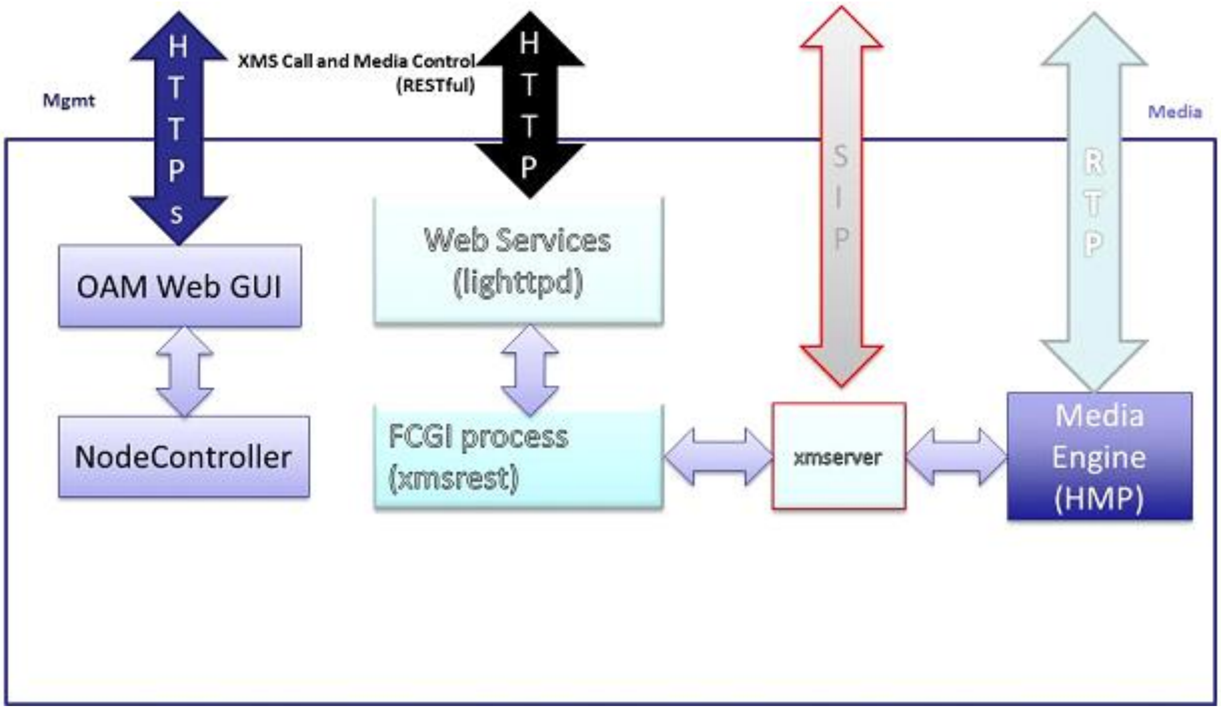
This User's Guide provides information about the Dialogic® PowerMedia™ Extended Media Server (also referred to herein as "PowerMedia XMS" or "XMS") RESTful API interface, including available features and resource-based component definitions.

The PowerMedia XMS RESTful API is one of several APIs that can be used to drive the PowerMedia XMS.

2. Overview

This section provides information about the PowerMedia XMS RESTful API interface, including available features and resource-based component definitions.

The PowerMedia XMS RESTful API is one of several APIs that can be used to drive the PowerMedia XMS. The architectural diagram below shows how the RESTful interface fits into PowerMedia XMS.



Two web servers are used in PowerMedia XMS:

- Apache (httpd) server
  Controls a web-based interface for operations, administration and maintenance.
- lighttpd server
  Processes call control and media commands delivered via the RESTful API as described in this guide. The lighttpd server includes a Fast Common Gateway Interface (FCGI) process, which allows efficient interfacing between PowerMedia XMS processes.

The PowerMedia XMS translates RESTful commands into the PowerMedia HMP media engine's low-level API. The media engine itself handles SIP calls, plays/records multimedia, and mixes multimedia conferences.

PowerMedia XMS provides two call control models:

- First party call control (1PCC)
  The application sends commands to the PowerMedia XMS to establish SIP or WebRTC calls on the application's behalf. In this model, the application does not need to be involved in making or receiving SIP calls and related SDP negotiation.
- Third party call control (3PCC)
  The application handles SIP calls or WebRTC signaling and SDP negotiation, and the PowerMedia XMS only performs media processing operations.

RESTful API Description

The PowerMedia XMS RESTful API uses a Representational State Transfer (RESTful) web service. This web service is a software system designed to support interoperable machine-to-machine interactions over a network, using the HTTP or HTTPS protocol. For more information on using HTTPS, refer to API Resources for port and authentication information.

The RESTful API consists of a series of requests and responses built around the transfer of representations of "resources". These resources are accessed through Universal Resource Indicators (URIs).

RESTful client-server architecture is where clients initiate requests to servers and servers process the requests and return appropriate responses.

In a RESTful application, the http client is the application which contains the business logic and PowerMedia XMS is the http server which handles the client request and processes the media commands.

Client Side Technologies

The "client side" refers to the client that communicates with the PowerMedia XMS and directs the session with the caller. Essentially, any language or operating system may be used to build a client. The main requirement is that the client supports HTTP and XML.

Listed below are some possible examples of client-side development platforms that can be used to command PowerMedia XMS services. Comments are included on multithreading, which is important for the event handler.

Java – This object-oriented, operating system-independent programming environment is fully multithreaded. Several XSD/XML parsers are available, as well as HTTP client class libraries. See the XML Schema Definition section for information on XSDs.

Note: The Dialogic Verification Demo used with the PowerMedia XMS is a Java application. Refer to the *Dialogic® PowerMedia™ XMS Quick Start Guide* for information about the Demo.

Python – This operating system-independent interpreted scripting language has POSIX threading available. HTTP protocol client library and Python XML/Schema processing tool are also available.

.NET – This Integral Microsoft Windows component supports the building and running of applications and XML web services. HTTP module and XSD schema definition tools are available.

Ruby – This open source scripting language contains a multiprocessing model that may be needed for the event handler. An HTTP client API and XSD validation tools are available.

C/C++ – These general purpose programming languages are fully multithreaded. cURL library (http://curl.haxx.se) is used for HTTP processing and Xerces C++ XML parser (http://xerces.apache.org/xerces-c) is used for XML. For a proof of concept, see http://www.dialogic.com/support/helpweb/helpweb.aspx/3584/powermedia_xms_restful_C_Sharp_demo/PM_XMS.

RESTful API with HTTP Methods

In the RESTful API, the four HTTP methods are translated to the actions shown in the following table.

HTTP Method	Request	Response
POST	Create a new resource	Contents of a newly created resource
PUT	Modify an existing resource	Contents of an updated resource
GET	Retrieve information for all instances of a specific resource type, or information regarding a specific resource	Contents of resource information
DELETE	Delete an existing resource	N/A

RESTful API Request/Response Model

The HTTP request/response model is the mechanism by which media control functionality is invoked. A RESTful HTTP request is sent to the PowerMedia XMS. The HTTP response carries the resulting response code of the operation, as well as a response body if it applies to the specific operation. The payload type used for the message body is XML. The request and response payloads are encoded to UTF-8.

All Call Resources

If a client wished to retrieve a list of all call resources currently active on the PowerMedia XMS, it would issue an HTTP **GET** request. The HTTP **GET** request would be sent on the web service with the IP address of the PowerMedia XMS <server>. For example:

```
http://<server>/default/calls?appid=app
```

If successful, the response code to the HTTP **GET** would be `200 OK`. The response body would resemble the following example:

```
<web_service version="1.0">
   <calls_response size="2">
   <call_response appid="master"
    identifier="123zdasdkz"
    href=/default/calls/123zdasdkz
    cpa = "yes"
    signaling = "yes"
    source_uri=sip:frank@10.20.34.3
    call_type="inbound" />
   <call_response
    identifier="178zdasdkz"
    href=/default/calls/178zdasdkz
    cpa = "no"
    signaling = "no"
    sdp=[sdp]
    call_type="3pcc" />
  </calls_response>
</web_service>
```

The above example shows a client requesting information for all calls with a response of two active identifiers along with the attributes of each call resource.

Single Call Resource

If a client wanted to retrieve information for only a single specific call resource, it would invoke the following HTTP **GET** request. The specific call identifier is part of the GET URL.

```
http://<server>/default/calls/1279697438?appid=app
```

If successful, the response code to the HTTP **GET** would be `200 OK`. The response body would be as follows:

```
<web_service version="1.0">
   <call_response appid="master"
    identifier="1279697438"
    href=/default/calls/1279697438
    cpa = "yes"
    signaling = "yes"
    source_uri=sip:frank@10.20.34.3
    call_type="inbound">
   </calls_response>
</web_service>
```

Additional request/response examples are contained within Resource-Based Components.

All XML sent to the PowerMedia XMS should have proper XML escape codes within string content. For example, when a <call> contains SDP info in the sdp="" attribute, newlines need to be converted to the proper XML escape code "
" and "&#xD".

```
<call sdp="v=0&#xA;o=sipclient 1376422095 1376422096 IN IP4
10.20.129.100&#xA;s=sipclient&#xA;c=IN IP4 10.20.129.100&#xA;t=0
0&#xA;m=audio 49162 RTP/AVP &#xA;a=rtpmap:0 pcmu/8000&#xA;a=sendrecv&#xA;" media="audiovideo"
signaling="no"/>
```

XML Schema Definition

PowerMedia XMS uses an XML schema definition (also referred to herein as "XSD"). The XSD formally describes the structure, content, and semantics of the XML payload for the PowerMedia XMS RESTful API call and media commands.

An XSD may be used to generate client-side code, allowing contents of XML documents to be treated as objects. The generated code usually enforces type-checking, thus supporting client-side validation of the XML payload before it is sent to the PowerMedia XMS.

Definitions of individual elements are referenced throughout this guide. The full XSD is provided in XML Schema Definition of Elements.

PowerMedia XMS RESTful API is designed using the following XML Schema declarations:

Element
An element describes the data it contains. It consists of a name and data type. When an element definition contains additional elements or attributes, it is a complex type.

```
<xs:element name="call_response">
```

Attribute
An attribute is a simple type definition that cannot contain other elements. Attribute names are always within quotation marks.

```
<xs:attribute name="media">
```

Sequence
Specifies the order in which attributes or elements within a complex type must be listed.

```
<xs:element name="call_response">
       <xs:complexType>
              <xs:sequence>
                    <xs:element ref="call_action" minOccurs="0" />
              </xs:sequence>
              <xs:attribute name="signaling" type="boolean_type" />
              <xs:attribute name="media" type="media_type" />
```

Complex Type
Defines an element containing other elements and attributes or mixed content (elements and text).

```
<xs:element name="call_response">
	<xs:complexType>
		<xs:sequence>
			<xs:element ref="call_action" minOccurs="0" />
		</xs:sequence>
		<xs:attribute name="signaling" type="boolean_type" />
		<xs:attribute name="destination_uri" type="xs:string" />
		<xs:attribute name="source_uri" type="xs:string" />
		<xs:attribute name="call_type" type="call_type_option" />
		<xs:attribute name="sdp" type="xs:string" />
		<xs:attribute name="cpa" type="boolean_type" />
		<xs:attribute name="media" type="media_type" />
		<xs:attribute name="dtmf_mode" type="dtmf_mode_option" />
		<xs:attribute name="async_dtmf" type="boolean_type" />
		<xs:attribute name="async_tone" type="boolean_type" />
		<xs:attribute name="cleardigits" type="boolean_type" />
		<xs:attributeGroup ref="response_attrgroup" />
	</xs:complexType>
</xs:element>
```

Simple Type
Creates a constrained data type for an element or attribute value.

```
<xs:simpleType name="call_type_option">
	<xs:restriction base="xs:string">
		<xs:enumeration value="inbound" />
		<xs:enumeration value="outbound" />
		<xs:enumeration value="3pcc" />
	</xs:restriction>
</xs:simpleType>
```

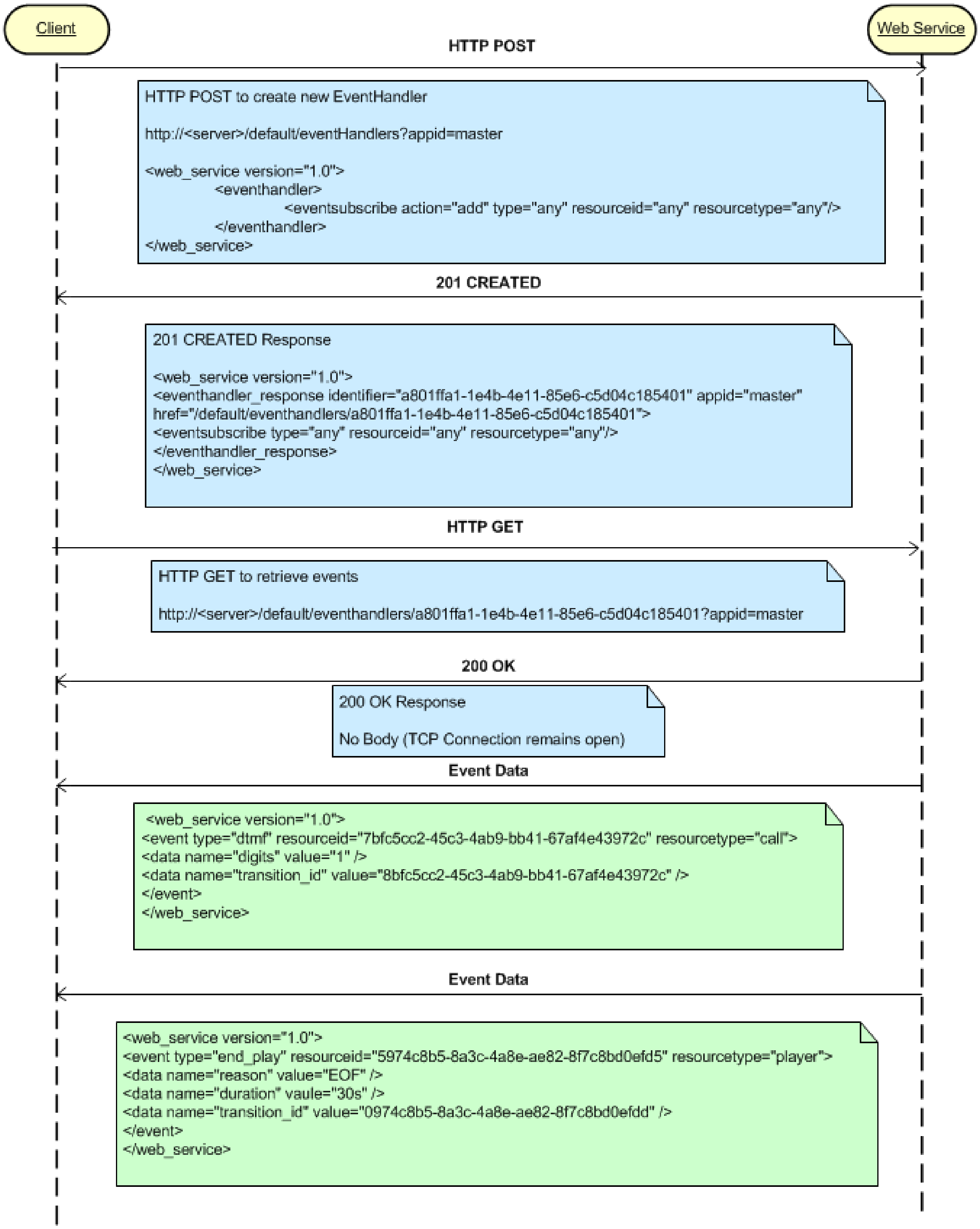
Refer to the specific resource-based element sections for more information.

Event Streaming

While most RESTful applications fit well into the HTTP request/response model, telephony applications must be able to handle unsolicited events such as digit detection and play completion. This concept is called Comet or HTTP event streaming. In a normal HTTP interaction, the client sends a request to the server, which processes it and sends the HTTP response. The connection between the client and server is then closed. This process will take place continuously as long as the web service is running; however, with HTTP event/data streaming, a reliable TCP connection remains open after the response is sent from the server, allowing the server to continue to send raw data to the client without solicitation or without client request.

For PowerMedia XMS, HTTP event streaming is implemented in the eventhandler resource. When the client wishes to receive asynchronous events, it uses an HTTP **POST** to create an eventhandler and subscribe to specific event types. The client then performs an HTTP **GET** on the newly created eventhandler and the PowerMedia XMS RESTful API responds with a `200 OK`; however, the TCP connection remains open. Any event data related to resources and event types are pushed to the client until the Eventhandler is deleted by the client.

The following diagram provides an example scenario where a client creates an eventhandler and receives digit detection and play completion events:



Response Codes

The following section lists the possible HTTP and RESTful error codes returned by RESTful service and their descriptions.

HTTP Error Code Mapping with RESTful Error Code

RESTful Error Code	HTTP Error Code	Description
1	400	Bad request
2	403	Access denied
3	404	Not found
4	501	Not implemented
5	500	Already exists
6	410	Unavailable
7	500	API interrupted
8	503	Resource busy
9	409	Bad state
10	503	Service unavailable
11	500	Server error
12	503	Resource unavailable
13	503	Device error
14	412	Preconditions error
15	406	Not acceptable
16	415	Media not supported
17	408	Timeout
18	400	Invalid parameter
19	500	Connection error
901	405	Method not allowed
902	404	Invalid URI

RESTful Error Code	HTTP Error Code	Description
903	400	Invalid XML payload
904	400	Invalid XML miss node
905	400	Payload version mismatch
906	400	Invalid XML miss web_service node
907	401	Invalid App ID
999	500	Unknown error

Error Response

If a request fails, there is an HTTP error code returned to the caller with XML payload as seen in the following example:

```
400 Bad Request

<web_service version="1.0">
    <error code="903" description="904 Invalid XML payload"/>
</web_service>
```

Success (200)

Code	Description
200	OK - request successful
201	Created - resource created
204	No Content - resource removed/deleted

Request Error (4xx)

Code	Description
400	Bad Request - request failed

Server Error (5xx)

Code	Description
500	Internal Server Error - RESTful service internal error
501	Not Implemented - request is not implemented/supported

3. Resource-Based Element Introduction

There are four resource-based elements used by the PowerMedia XMS RESTful web service:

- Call Resource
- Conference Resource
- Event Handler Resource
- MRCP Resource

These elements are used in conjunction with one another to direct the PowerMedia XMS to make and receive calls, handle media during a call, manipulate audio and video conferences, invoke ASR/TTS speech services, and to catch events relating to calls, conferences and their media.

Each element makes use of the various HTTP methods POST, PUT, DELETE and GET. The resource-based element chapters in this document contain HTTP method tables that define the request body content type if a request body is allowed. In addition, the tables supply the possible return code values as well as the response body content. The tables also contain an example payload for that specific resource-based element type. This XML content is used in both HTTP requests and responses. Refer to the specific resource-based element sections for information about each of the elements.

Applications and Application IDs (appid)

The *appid* shown in the URL request examples is for identifying the resources used, owned, and created by a specific application. For example:

```
<web_service version="1.0">
  <call_response
    appid="[appid]"
    identifier="[call_id]"
    href="/default/calls/[call_id]?appid=[appid]"
    signaling = "yes"
    source_uri=[uri]
    sdp=[sdp]
    call_type="inbound" />
</web_service>
```

Discrete appids are defined so that multiple applications may be simultaneously run on a single PowerMedia XMS. The appid indicates the ownership of a RESTful resource so that each resource that is created has an associated appid. The resources can only be viewed, modified, or deleted by an application with a matching appid. The appid is used throughout the PowerMedia XMS RESTful API to identify the intended application.

Note: The appid is pre-defined on the Routing page of the PowerMedia XMS Admin Console (also referred to herein as "Console"), which is used for post-operating system installation and configuration tasks. New appids may be added, or unwanted appids removed on the Routing page. Refer to the *Dialogic® PowerMedia™ XMS Installation and Configuration Guide* for detailed information about the Console.

Time Values

Values that represent time in the RESTful API are specified in whole numbers of seconds ("s") or milliseconds ("ms") whichever is appropriate. For example:

```
<send_dtmf digits="1234" duration="100ms" interval="100ms" level="-10dB"/>
<play offset="2s" repeat="3" delay="0s" terminate_digits="#" max_time="infinite"
skip_interval="10s">
```

4. API Resources

The *appid* shown in the following URL request is for identifying the resources used, owned, and created by the application. All resource base URL:

```
http or https://[ipaddress:port]/default/
http default port: 81
https default port: 444
The authentication (basic method) for HTTPS: username:"admin" password: "admin"
```

Note: "/default/" is a placeholder.

The resources provided by the RESTful API will differ between the various modes.

```
default appid=app
```

The table below lists all available resources, their sub-resources, valid HTTP methods that may be used with them, and attributes for which they are valid.

Clicking on the resource or sub-resource will:

- Provide its definition
- Provide valid values for the parameters that can be set
- Define how each valid method affects the resource
- Give an example of a request and a response payload

Resource	Sub-Resource	HTTP Methods Supported
Call Resource /calls?appid=[app_id]		GET, POST
	Call Sub-Resource /calls/[call_id]?appid=[app_id]	GET, PUT, DELETE
Conference Resource /conferences?appid=[app_id]		GET, POST
	Conference Sub-Resource /conferences/[conference_id]?appid=[app_id]	GET, PUT, DELETE
MRCP Resource /mrcps?appid=[app_id]		GET, POST
	MRCP Sub-Resource /mrcps/[mrcp_id]?appid=[app_id]	GET, PUT, DELETE
Event Handler Resource /eventhandlers?appid=[app_id]		GET, POST
	Event Handler Sub-Resource /eventhandlers/[eventhandler_id]?appid=[app_id]	GET, PUT, DELETE

5. List of Available Resources

Call Resource

The Call Resource creates and manages the media/signal connection between the remote media endpoint (typically a SIP or WebRTC endpoint) and the PowerMedia XMS.

The Call Resource has the following types:

- **Inbound**
  This call resource is created by the PowerMedia XMS RESTful server when an incoming call is received. The application is then informed of the inbound call via the eventhandler resource.
- **Outbound**
  This call resource is created by an application that wishes to make a media stream connection from the PowerMedia XMS RESTful server to a SIP or WebRTC endpoint.
- **3PCC**
  This call resource is requested by the application without requesting the PowerMedia XMS RESTful server to provide signaling control. The call resource will be created based on the Session Description Protocol (SDP) info that is provided by the application. PowerMedia XMS can handle SDP from standard SIP or WebRTC endpoints. 3PCC outbound calls should have SDP="" so that PowerMedia XMS can generate the SDP for the app to include in the INVITE while 3PCC incoming calls should have SDP set to whatever is in the incoming INVITE.

  Note: In order to set the PowerMedia XMS to 3PCC mode, you need to set the signaling="no".

Media-related properties and actions associated with the media connection are defined in this section. These include play, playcollect, playrecord, join/unjoin, and stop.

Note: When using playrecord and record, playrecord executes the play list first. Once all of the specified plays are complete, the record functionality is executed.

For call sub-resources, see the Call Sub-Resource section.

The following tables show the HTTP methods that can be used with one or more calls.

Note: The payloads shown are examples only as there are many possible variations.

Note: The URI parameters for play, record, and playrecord functions can be both required and optional because a mininimum of one URI parameter (audio and/or video) must be included when issuing those functions. For example, you can have one audio URI for play, record, and playrecord implementation; you can have one video URI for play, record, and playrecord implementation; or you can have both audio and video URIs for play, record, and playrecord implementation.

calls

Resource URI

`/calls?appid=[app_id]`

HTTP GET

Retrieves all available call resources.

`GET /calls?appid=[app_id]`

Response Payload Example


```
<web_service version="1.0">
   <calls_response size="2">
         <call_response appid="app" async_dtmf="yes" async_tone="yes" audio="sendrecv"
call_type="inbound"
              cleardigits="no" connected="yes" cpa="no" destination_uri="sip:sip@10.20.129.100"
dtmf_mode="rfc2833"
              href="/default/calls/8ae87129-b334-4d8a-bec6-5d4ddeba5649"
              identifier="8ae87129-b334-4d8a-bec6-5d4ddeba5649" info_ack_mode="automatic"
media="audiovideo" signaling="yes"
              source_uri="sip:Username@10.20.129.113:5060" video="sendrecv"
gusid="b9a5ef23398749bfa6809fae974378f2">
         </call_response>
         <call_response appid="app" async_dtmf="yes" async_tone="yes" audio="sendrecv"
call_type="inbound"
              cleardigits="no" connected="yes" cpa="no" destination_uri="sip:sip@10.20.129.100"
dtmf_mode="rfc2833"
              href="/default/calls/f1e9040f-4ca2-4dd6-81e6-c665385ffde8"
              identifier="f1e9040f-4ca2-4dd6-81e6-c665385ffde8" info_ack_mode="automatic"
media="audiovideo" signaling="yes"
              source_uri="sip:Username@10.20.129.113:5060" video="sendrecv"
gusid="b9a5ef23398749bfa6809fae97437abc">
         </call_response>
   </calls_response>
</web_service>
```


HTTP POST

Creates a call resource.

`POST /calls?appid=[app_id]`

Create Call Types

- Outbound
- 3PCC (application handles SIP or WebRTC call signaling)

Request Payload Attributes

Parameter	Default	Optional	Description
signaling	"yes"	*	Specifies if signaling is done by this media server ("yes") or a third party application server ("no").
sdp	Set by system	*	Session Description Protocol data. Only used for third party call control (3PCC). **Note:** If the data contains newlines or carriage returns, make sure that they are replaced with the XML equivalent of "&#xA" prior to sending.
media	"audio"	*	Sets the media type supported by the call. Values: • "audio" • "audiovideo" • "video" • "message" • "audiofax" • "image"
dtmf_mode	"rfc2833"	*	Specifies the signaling mode for DTMF digits. Values: • "inband" • "outofband" • "rfc2833"
cpa	"no"	*	Specifies if call progress detection is used for an outbound call (signaling only). Values: • "yes" • "no"

Parameter	Default	Optional	Description
cpa_profile	(system default)	*	Use configuration profile for call progress analysis. See the *Dialogic® PowerMedia™ XMS Installation and Configuration Guide* for more information.
codec_profile	(system default)	*	Use codec profile for codec prioritization. See the *Dialogic® PowerMedia™ XMS Installation and Configuration Guide* for more information.
info_ack_mode	"automatic"	*	Specifies how INFO events are acknowledged (signaling only). Values: • "automatic" • "manual"
rx_volume	(none)	*	Volume adjustments are allowed between +31dB and -32dB. Both absolute (default) and relative adjustments are supported. Values: • "+3dB;relative" • "+3dB;absolute"
tx_volume	(none)	*	Volume adjustments are allowed between +31dB and -32dB. Both absolute (default) and relative adjustments are supported. Values: • "+3dB;relative" • "+3dB;absolute"

Parameter	Default	Optional	Description
async_dtmf	(none)	*	Specifies if DTMF digits should be reported as events instead of being buffered internally (default). When active, the application must ensure that when using the any of the play APIs, any unused digit processing parameters are cleared. This is to avoid digits being processed both internally and by the application. Values: • "yes" • "no"
cleardigits	(none)	*	The parameter is considered when async_dtmf is set to "yes" and specifies whether previous, buffered, input should be discarded. Values: • "yes" • "no"
async_tone	(none)	*	Specifies if tones are reported as events outside of a playcollect action. Values: • "yes" • "no"
destination_uri	(none)	*	Destination address. For SIP, this is the Request-URI. Requires RFC 3986 Section 2 encoding rules.
source_uri	(none)	*	Caller address. For SIP, this is the From header. Requires RFC 3986 Section 2 encoding rules.
called_uri	(none)	*	Logical destination address. For SIP, this is the To header.

Parameter	Default	Optional	Description
display_name	(none)	*	Caller's display name.
dial_timeout	"30s"	*	Maximum time to wait for the call to be answered by the called party.
encryption	"none"	*	Media stream (RTP) encryption. Values: • "none" • "dtls"
ice	"no"	*	Use ICE (Interactive Connectivity Establishment) to configure media streams (RTP). Values: • "no" • "yes"
audio	"sendrecv"	*	Direction of audio media. Values: • "inactive" • "sendonly" • "recvonly" • "sendrecv"
video	"sendrecv"	*	Direction of video media. Values: • "inactive" • "sendonly" • "recvonly" • "sendrecv"
message	"sendrecv"	*	Direction of message media (MSRP). Values: • "inactive" • "sendonly" • "recvonly" • "sendrecv"

Parameter	Default	Optional	Description
setup	"active"	*	MSRP message role. Values: • "active" • "passive"
content_type	(none)	*	Mime type describing content (outbound call only).
content	(none)	*	Data (outbound call only).
local_rtp_address	(none)	*	Sets the local IP address to use for RTP.
hangup_ack_mode	"automatic"	*	Specifies how HANGUP events are acknowledged (signaling only). Values: • "automatic" • "manual" **Note:** If hangup_ack_mode is set to "manual", the app/user will be responsible for deleting the call resource, and the RESTful service will not automatically delete the call resource after it receives the "hangup" event. Refer to send_hangup_ack.
answer_ack_mode	"automatic"	*	Specifies how CONNECTED events are acknowledged. Values: • "automatic" • "manual"
prack_mode	"automatic"	*	Specifies how PRACK messages are sent. Values: • "automatic" • "manual"

Parameter	Default	Optional	Description
prack_level	"automatic"	*	Specifies to the remote if PRACK messages are supported or required. Values: • "supported" • "required"
headers	(none)	*	Use sip_headers instead. Raw SIP headers, delimited by the <CR><LF> end-of-line characters (outbound call only). **Note:** This is deprecated since PowerMedia XMS Release 3.0.
sip_headers	(none)	*	Refer to sip_headers_attributes.
rtcp_feedback		*	Specifies the RTCP feedback for media in the offering SDP (AVPF/SAVPF). Leave empty to select the configuration mode. Values: • "video" • "audio" • "audiovideo" • "none" **Note:** RTCP feedback is currently only implemented for video. The audio value has no effect.
gusid	(auto generated if not defined by user)	*	Specifies the Global Unique Session Identifier (gusid).

Request Payload Example

Outbound

```
<web_service version="1.0">
  <call media="audiovideo" signaling="yes" dtmf_mode="rfc2833" async_dtmf="yes" async_tone="yes"
       rx_delta="+0dB"
                tx_delta="+0dB" destination_uri="sip:xmstool@10.20.129.102"
       source_uri="sip:xmstool@146.152.124.182" cpa="no" />
</web_service>
```

3PCC

```
<web_service version="1.0">
  <call sdp="[sdpinfo]" media="audiovideo" signaling="no"/>
</web_service>
```

Response Payload Example

Outbound

```
<web_service version="1.0">
     <call_response identifier="7f0e358f-5786-41df-bad3-866f73d044a7" appid="app"
               href="/default/calls/7f0e358f-5786-41df-bad3-866f73d044a7"
               connected="no" signaling="yes" cpa="no" call_type="outbound"
               media="audiovideo"
               dtmf_mode="rfc2833"
               destination_uri="sip:xmstool@10.20.129.102"
               source_uri="sip:xmstool@146.152.124.182"
               async_dtmf="yes" async_tone="yes" cleardigits="no" info_ack_mode="automatic">
               gusid="b9a5ef23398749bfa6809fae974378f2">
     </call_response>
</web_service>
```

3PCC

```
<web_service version="1.0">
<call_response identifier="ca494fe7-17d7-4b6a-a7d0-e89592eef262" appid="app"
href="/default/calls/ca494fe7-17d7-4b6a-a7d0-e89592eef262"
sdp="v=0
o=xmserver 1376426725 1376426726 IN IP4 10.20.129.61
s=xmserver
c=IN IP4 10.20.129.61
t=0 0
m=audio 49152 RTP/AVP 9 0 8 96 97 4 18 98 101
a=rtpmap:9 g722/8000
a=rtpmap:0 pcmu/8000
a=rtpmap:8 pcma/8000
a=rtpmap:96 g726-32/8000
a=rtpmap:97 amr/8000
a=fmtp:97 octet-align=0
a=rtpmap:4 g723/8000
a=fmtp:4 annexa=yes
a=rtpmap:18 g729/8000
a=fmtp:18 annexb=no
a=rtpmap:98 amr-wb/16000
```

```
a=fmtp:98 octet-align=0
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
a=sendrecv
m=video 57344 RTP/AVP 100 98 34 96 97 102
b=AS:1000
a=rtpmap:100 h264/90000
a=fmtp:100 profile-level-id=42001F; packetization-mode=1; max-br=768
a=rtpmap:98 mp4v-es/90000
a=fmtp:98 profile-level-id=3 MaxBR=3840
a=rtpmap:34 h263/90000
a=fmtp:34 CIF=1; CIF=2; CIF=3; QCIF=1; QCIF=2
a=rtpmap:96 h263-1998/90000
a=fmtp:96 CIF=1; CIF=2; CIF=3; QCIF=1; QCIF=2
a=rtpmap:97 h263-2000/90000
a=fmtp:97 CIF=1; CIF=2; CIF=3; QCIF=1; QCIF=2
a=rtpmap:102 vp8/90000
a=fmtp:102 max-fr=30; max-fs=1200
a=sendrecv
"
signaling="no" cpa="no" call_type="3pcc"
media="audiovideo"
dtmf_mode="rfc2833"
source_uri="sip:xmstool@146.152.124.182"
async_dtmf="yes" async_tone="yes" cleardigits="no" encryption="none" ice="no"
info_ack_mode="automatic">
gusid="b9a5ef23398749bfa6809fae974378f2"
</call_response>
</web_service>
```

Note: If you send a request in 3PCC mode with an empty SDP="", PowerMedia XMS will allocate an internal device and return back the SDP. This allows SDP to initiate and outbound call with early media.

Call Concepts

This section contains a higher-level look at various aspects of PowerMedia XMS call behavior.

Asynchronous Tones

- Audio tones can be used to both terminate operations and can also be delivered to the application as asynchronous events.
- To terminate a playrecord or play operation, set terminate_digits to the desired DTMF digit value (0-9, * and #). If the operation is terminated this way, the end_playrecord or end_play event will reference the digit collected to end the operation.
- To have an async DTMF event delivered to your application outside of its use as a play/record termination, async_dtmf=yes should be set for a call resource.

- User-defined tone detection is set up using tone templates, which are created in the Tones screen through the Console (see the *Dialogic® PowerMedia™ XMS Installation and Configuration Guide* for more information). Set async_tones=yes for a call resource and detection of the defined tones is activated.
- DTMF events are delivered as event type "dtmf", with a name of "digits" and a value corresponding to the digit collected.
- Async tone events are delivered as event type "tone", with a name of "tone" and a value corresponding to the name assigned when the user-defined tone was created.

Playcollect and User-Defined Tones

A playcollect operation usually is used to collect DTMF tones but may collect user-defined tone as well.

To do this, the attribute tone_detection=yes must be set when the playcollect is initiated. The end_playcollect event will reference the user-defined name of the tone collected and the reason parameter will be set to tone. Duration (in milliseconds) refers to the length of the play operation, not the duration of the tone.

Early Media

Early media refers to the ability to play media (audio and/or video) on an inbound call to the caller before a call is answered. The call must first be Accepted before the media play operation can be started. To fully complete the connection, the call must be answered. A PUT should be used for these messages.

Media File Locations

The default location for media files used by the PowerMedia XMS RESTful API is displayed when the Media menu is chosen through the Console. See the Media section in the *Dialogic® PowerMedia™ XMS Installation and Configuration Guide*.

Files for a given RESTful application are usually grouped under a directory reflecting the application name. Directories/applications delivered with the system are "verification" for the RESTful Verification Demo and "xmstool" for the XMSTool RESTful Utility.

The file:// URL used in a RESTful Play command points at a set of audio and video files named *Dialogic.mp4*, which are located in the verification directory in the default media location:

```
<call>

    <call_action>

      <record terminate_digits="#" max_time="30s"
recording_audio_uri="file://./verification/Dialogic.mkv" recording_audio_type="audio/mkv"
recording_video_uri="file://./verification/Dialogic.mkv" recording_video_type="video/mkv">

        <recording_audio_mime_params codec="OPUS" rate="16000"/>

        <recording_video_mime_params codec="vp8" profile="66" framerate="15" maxbitrate="768000"
height="480" width="640"/>

      </record>

    </call_action>

</call>
```

Similarly, the location of an audio file for a recording is given in the same way:

```
<call>
    <call_action>
      <play offset="0s" delay="0s" repeat="0" terminate_digits="#">
        <play_source audio_uri="file://./verification/Dialogic.mkv" audio_type="audio/mkv"
audio_rate="8000" video_uri="file://./verification/Dialogic.mkv" video_type="video/mkv"/>
      </play>
    </call_action>
</call>
```

Call Sub-Resource

For details on call resources, see the Call Resource section.

call

Resource URI

`/calls/[call_id]?appid=[app_id]`

HTTP GET

Retrieves an available call resource.

`GET /calls/[call_id]?appid=[app_id]`

Response Payload Example

```
<web_service version="1.0">
    <call_response appid="app" async_dtmf="yes" async_tone="yes" audio="sendrecv"
call_type="inbound"
              cleardigits="no" connected="yes" cpa="no" destination_uri="sip:sip@10.20.129.100"
dtmf_mode="rfc2833"
              href="/default/calls/8ae87129-b334-4d8a-bec6-5d4ddeba5649"
              identifier="8ae87129-b334-4d8a-bec6-5d4ddeba5649" info_ack_mode="automatic"
media="audiovideo" signaling="yes"
              source_uri="sip:Username@10.20.129.113:5060" video="sendrecv">
              gusid="b9a5ef23398749bfa6809fae974378f2">
    </call_response>
</web_service>
```

HTTP PUT

Updates a call resource.

`PUT /calls/[call_id]?appid=[app_id]`

Accept/Answer Incoming Call

Request Payload Attributes

Parameter	Default	Optional	Description
answer	(none)	*	Answer an incoming call. Values: • "yes" • "no"
accept	(none)	*	Accept an incoming call. Values: • "yes" • "no"
media	"audio"	*	Sets the media type supported by the call. Values: • "audio" • "audiovideo" • "video" • "message" • "audiofax" • "image" This parameter has no effect if the call has been accepted already.
dtmf_mode	"rfc2833"	*	Specifies the signaling mode for DTMF digits. Values: • "inband" • "outofband" • "rfc2833"
early_media	"no"	*	Enable early media. Values: • "yes" • "no" This parameter has no effect if answer=yes.

Parameter	Default	Optional	Description
info_ack_mode	"automatic"	*	Specifies how INFO events are acknowledged. Values: • "automatic" • "manual"
hangup_ack_mode	"automatic"	*	Specifies how HANGUP events are acknowledged (signaling only). Values: • "automatic" • "manual" **Note:** If hangup_ack_mode is set to "manual", the app/user will be responsible for deleting the call resource, and the RESTful service will not automatically delete the call resource after it receives the "hangup" event. Refer to send_hangup_ack.
async_completion	"no"	*	Values: • "yes" the API will return immediately and completion is indicated by an ANSWERED event. • "no" the API will block until the call has been fully answered. This parameter has effect if answer=yes.

Parameter	Default	Optional	Description
encryption	(none)	*	Media stream (RTP) encryption. Values: • "none" • "dtls"
ice	(none)	*	Use ICE (Interactive Connectivity Establishment) to configure media streams (RTP). Values: • "no" • "yes"
content_type	(none)	*	Mime type describing content (answer only).
content	(none)	*	Data content (answer only).
media	(none)	*	The media types supported by the call. Values: • "audio" • "audiovideo" • "video"
audio	(none)	*	Direction of audio media. Values: • "inactive" • "sendonly" • "recvonly" • "sendrecv"
video	(none)	*	Direction of video media. Values: • "inactive" • "sendonly" • "recvonly" • "sendrecv"

Parameter	Default	Optional	Description
message	"sendrecv"	*	Direction of message media (MSRP). Values: • "inactive" • "sendonly" • "recvonly" • "sendrecv"
image	"sendrecv"	*	Direction of message media (MSRP). Values: • "inactive" • "sendonly" • "recvonly" • "sendrecv"
local_rtp_address	"sendrecv"	*	Sets the local IP address to use for RTP.
prack_ack_mode	"automatic"	*	Specifies how PRACK events are acknowledged. Values: • "automatic" • "manual" For accepting a call only.
prack_level	"none"	*	Specifies if the prack mode sent in sip 1XX (RINGING or PROGRESS), in response to a remote INVITE with prack mode "supported," is required or not. Values: • "required" • "none" For accepting a call only.

Parameter	Default	Optional	Description
rtcp_feedback		*	Specifies the RTCP feedback for media in the offering SDP (AVPF/SAVPF). Leave empty to select the configuration mode. Values: • "video" • "audio" • "audiovideo" • "none" **Note:** RTCP feedback is currently only implemented for video. The audio value has no effect.
cpa	(none)	*	Use call progress detection: Values: • "yes" • "no"
cpa_profile	(system default)	*	Use configuration profile for call progress analysis. See the *Dialogic® PowerMedia™ XMS Installation and Configuration Guide* for more information.
codec_profile	(system default)	*	Use codec profile for codec prioritization. See the *Dialogic® PowerMedia™ XMS Installation and Configuration Guide* for more information.

Request Payload Example

accept

```
<web_service version="1.0">
  <call accept="yes" early_media="yes" />
</web_service>
```

answer

```
<web_service version="1.0">
```

```
  <call answer="yes" async_completion="yes"/>
</web_service>
```

Response Payload Example

accept

```
<web_service version="1.0">
     <call_response identifier="8fe4c7f4-067c-455d-afee-52f57e00314b" appid="app"
            href="/default/calls/8fe4c7f4-067c-455d-afee-52f57e00314b"
            connected="yes" signaling="yes" cpa="no" call_type="inbound"
            media="audio"
            dtmf_mode="rfc2833"
            source_uri="sip:Username@10.20.129.113:5060"
            async_dtmf="no" async_tone="no" cleardigits="no" info_ack_mode="automatic"
early_media="yes">
            gusid="b9a5ef23398749bfa6809fae974378f2">
     </call_response>
</web_service>
```

answer

```
<web_service version="1.0">
      <call_response identifier="8fe4c7f4-067c-455d-afee-52f57e00314b" appid="app"
             href="/default/calls/8fe4c7f4-067c-455d-afee-52f57e00314b"
             connected="yes" signaling="yes" cpa="no" call_type="inbound"
             media="unknown"
             dtmf_mode="rfc2833"
             source_uri="sip:Username@10.20.129.113:5060"
             async_dtmf="no" async_tone="no" cleardigits="no" info_ack_mode="automatic"
early_media="yes">
             gusid="b9a5ef23398749bfa6809fae974378f2">
      </call_response>
</web_service>
```

Modify Call Attributes

Request Payload Attributes

Parameter	Default	Optional	Description
sdp	(none)	*	Only used in 3rd party call control (3PCC).
rx_delta	(none)	*	Volume adjustments are allowed between +31dB and -32dB. Both absolute (default) and relative adjustments are supported. Usage: • "+3dB;relative" • "+3dB;absolute"

Parameter	Default	Optional	Description
tx_delta	(none)	*	Volume adjustments are allowed between +31dB and -32dB. Both absolute (default) and relative adjustments are supported. Usage: • "+3dB;relative" • "+3dB;absolute"
async_dtmf	(none)	*	Specifies if DTMF digits should be reported as events instead of being buffered internally (default). When active, the application must ensure that when using any of the play APIs, any unused digit processing parameters are cleared. This is to avoid digits being processed both internally and by the application. Values: • "yes" • "no"
cleardigits	(none)	*	The parameter is considered when async_dtmf is set to "yes" and specifies whether previous, buffered, input should be discarded. Values: • "yes" • "no"
async_tone	(none)	*	Specifies if tones are reported as events outside of a playcollect action. Values: • "yes" • "no"

Parameter	Default	Optional	Description
media	(none)	*	Sets the media type supported by the call. Values: • "audio" • "audiovideo" • "video" • "message" • "audiofax" • "image"
audio	(none)	*	Direction of audio media. Values: • "inactive" • "sendonly" • "recvonly" • "sendrecv"
video	(none)	*	Direction of video media. Values: • "inactive" • "sendonly" • "recvonly" • "sendrecv"
message	(none)	*	Direction of message media (MSRP). Values: • "inactive" • "sendonly" • "recvonly" • "sendrecv"
image	"sendrecv"	*	Direction of image media (MSRP). Values: • "inactive" • "sendonly" • "recvonly" • "sendrecv"

Request Payload Example

```
<web_service version="1.0">
  <call async_dtmf="yes" async_tone="yes" rx_delta="+3dB" tx_delta="+3dB" />
</web_service>
```

Response Payload Example

```
<web_service version="1.0">
     <call_response identifier="8fe4c7f4-067c-455d-afee-52f57e00314b" appid="app"
            href="/default/calls/8fe4c7f4-067c-455d-afee-52f57e00314b"
            connected="yes" signaling="yes" cpa="no" call_type="inbound"
            media="audio"
            dtmf_mode="rfc2833"
            source_uri="sip:Username@10.20.129.113:5060"
            async_dtmf="yes" async_tone="yes" cleardigits="no" info_ack_mode="automatic"
early_media="yes">
            gusid="b9a5ef23398749bfa6809fae974378f2">
     </call_response>
</web_service>
```

Perform Call Action

- play
- update_play
- record
- multi_record
- update_multi_record
- playrecord
- playcollect
- overlay
- send_dtmf
- send_event
- stop
- add_party/update_party
- remove_party
- join/unjoin
- send_info/send_info_ack
- send_prack/send_prack_ack/send_answer_ack
- send_message
- dial
- transfer
- redirect
- hangup
- send_hangup_ack

- get_call_info
- add_ice_candidate
- get_last_action
- get_last_event
- cpa_operation

play_source_attributes

Note: At least one of the following parameters are required: location (deprecated), audio_uri, or video_uri. For example, if a play_source includes audio_uri, location (deprecated) and video_uri are not required.

Note: If the media has a header, the audio type/rate is optional. Otherwise, the audio type/rate is required. Refer to Play Media.

Parameter	Default	Optional	Description
location	(none)		The URI of the media to be played e.g., "file://...", "rtsp://...", "image:", "http(s)://...". **Note:** This is deprecated. Use audio_uri and video_uri.
base_audio_uri	(none)	*	Base URI prefix for audio_uri URIs e.g., "file://...", "rtsp://...", "http(s)://...". **Note:** This is deprecated. Use audio_base_uri.
audio_base_uri	(none)	*	Base URI prefix for audio_uri URIs e.g., "file://...", "rtsp://...", "http(s)://...".

Parameter	Default	Optional	Description
audio_uri	(none)		The URI of the audio media to be played e.g., "file://...", "rtsp://...", "http(s)://..." or <filename>.<ext> if the base_audio_uri has been set. Multiple URIs can be specified by using the newline character '\n' as a separator. Only URIs of type "file://..." are currently supported when playing multiple files.
audio_type	(none)	*	The mime type of the audio media. Values: • "audio/x-wav" • "audio/basic" • "audio/x-alaw-basic" • "audio/L8" • "audio/L16" • "audio/x-aud" • "audio/AMR" • "audio/AMR-WB" • "audio/3gpp" • "audio/mp4" • "audio/mkv" • "audio/webm" • "text/uri-list"
audio_rate	(none)	*	The mime type of the audio media. Values: • "8000" • "11025" • "16000"

Parameter	Default	Optional	Description
audio_track_id	"0"	*	The audio track to play (multitrack files). Values: 0…n Only "0 " or "1" currently supported.
base_video_uri	(none)	*	Base URI prefix for video_uri URIs e.g., "file://...", "rtsp://...", "http(s)://...". **Note:** This is deprecated. Use video_base_uri.
video_base_uri	(none)	*	Base URI prefix for video_uri URIs e.g., "file://...", "rtsp://...", "http(s)://...".
video_uri	(none)		The URI of the video media to be played e.g., "file://...", "rtsp://...", "http(s)://...or <filename>.<ext> if the base_video_uri has been set. Multiple URIs can be specified by using the newline character '\n' as a separator. Only URIs of type "file://..." are currently supported when playing multiple files.
video_type	(none)	*	The mime type of the video media. Values: • "video/x-vid" • "video/3gpp" • "image/jpeg" • "video/mp4" • "video/mkv" • "video/webm"

recording_audio_mime_params

Note: The required recording audio mime parameters are based on the recording audio type. Refer to Record Audio.

Parameter	Default	Optional	Description
codec	(none)		Sets the audio codec. Values: • "L16" • "mulaw" • "alaw" • "L8" • "AMR" • "AMR-WB" • "OPUS" • "native"
rate	(none)		Sets the audio rate. Values: • "8000" • "11025" • "16000 (L16 only)"
mode	(none)		Sets the audio AMR/AMR-WB mode. Values: • "mode=0..7 (AMR only)" • "mode=0..8 (AMR-WB only)"

recording_video_mime_params

Note: The required recording video mime parameters are based on the recording video type. Refer to Record Video.

Parameter	Default	Optional	Description
codec	(none)		Sets the video codec. Values: • "h263" • "h264" • "mp4v-es" • "vp8" • "vp9"
profile	(none)		Sets the video profile.
level	(none)		Sets the video level.
framerate	(none)		Sets the video framerate. If framerate is set to 0 when using the H.264, VP8, or VP9 video codec, all frames are encoded.
maxbitrate	(none)		Sets the video maxbitrate.
height	(none)		Sets the video height. If height and width are set to 0 when using the H.264, VP8, or VP9 video codec, the input video stream's resolution parameters are identified by the system and used as the recording resolution parameters.
width	(none)		Sets the video width. If height and width are set to 0 when using the H.264, VP8, or VP9 video codec, the input video stream's resolution parameters are identified by the system and used as the recording resolution parameters.

dvr_setting_attributes

Parameter	Default	Optional	Description
forward_key	"1"	*	Defines the DTMF key [0-9,*,#] used to skip forwards.

Parameter	Default	Optional	Description
backward_key	"2"	*	Defines the DTMF key [0-9,*,#] used to skip backwards.
pause_key	"3"	*	Defines the DTMF key [0-9,*,#] used to pause playback.
resume_key	"4"	*	Defines the DTMF key [0-9,*,#] used to resume playback.
restart_key	"5"	*	Defines the DTMF key [0-9,*,#] used to restart playback.

param_attributes

Parameter	Default	Optional	Description
name	(none)		Sets parameter name.
value	(none)		Sets parameter value.

record_track

Parameter	Default	Optional	Description
id	(none)		Call or conference ID.
media	(none)	*	Media to be recorded. Values: • "audio" • "video" (not currently supported)
direction	recv	*	For calls only. This specifies the direction of the stream to be recorded. For conference, only "recv" direction is supported which is the output of the conference. Values: • "send" • "recv"

Parameter	Default	Optional	Description
track_id	(none)		The sequence number (0…n) for record_track for response payload only. It will be ignored in the request payload.

content_element_attributes

Parameter	Default	Optional	Description
id	(none)		Content identifier.
type	(none)		Mime type describing content.
content	(none)		Data content.

sip_headers_attributes

Parameter	Default	Optional	Description
raw_sip_headers	(none)		Raw SIP headers, delimited by the <CR><LF> end-of-line characters.
params	(none)		Refer to param_attributes and get_call_info for examples.

msg_payload_attributes

Parameter	Default	Optional	Description
msg_payload_content	(none)		Refer to msg_payload_content.
msg_payload_uri	(none)	* (mutually exclusive with msg_payload_content)	Refer to msg_payload_uri.
content_type	(none)		Mime type describing content.
content_disposition	"attachment"	*	The intended disposition of the file. Values: • "attachment" • "render"

Parameter	Default	Optional	Description
content_id	(none)		Content identifier.

msg_payload_content attributes

Parameter	Default	Optional	Description
content	(none)		Data content.

msg_payload_uri attributes

Parameter	Default	Optional	Description
uri	(none)		File to transfer ("rfc5547" only) e.g., "file:///tmp/foo.txt".

play

Request Payload Attributes

Parameter	Default	Optional	Description
play_source	(none)		Refer to play_source_attributes and get_call_info for examples.
dvr_setting	(none)	*	Refer to dvr_setting_attributes.
offset	"0"	*	Specifies the time offset from where the play should start. Note, the "offset" is applied to the initial play only.
repeat	"0"	*	Number of times to repeat the play. Use "infinite" to repeat indefinitely. "file://" URIs only.
delay	"1s"	*	Time delay between repeated plays.
terminate_digits	"#"	*	The digit or digits [0-9,*,#] used to terminate the play (call only).

Parameter	Default	Optional	Description
max_time	"infinite"	*	Limit the playback time to this value.
skip_interval	"1s"	*	Defines the amount of time to skip on the "forward" and "backwards" actions (call only).
no_cache		*	Cache-control for HTTP URIs. Values: • "yes" does not use caching. • "no" uses caching. The default is set by the configuration.
max_age		*	Cache-control for HTTP URIs, the maximum age, in seconds, of a cached file. The default is set by the configuration.
max_stale		*	Cache-control for HTTP URIs, the number of seconds that a cached file may exceed its expiration time by and still be a considered as fresh. The default is set by the configuration.
fetch_timeout	"300s"	*	HTTP URIs, the maximum time in seconds to retrieve the file. This is the overall period of the transaction.

Parameter	Default	Optional	Description
region	"0"	*	Conference only. The ID of the region that will display the video media. The value "0" causes the video to be shown full screen with the current layout being restored automatically when the play back completes.

Request Payload Example

Individual Play


```
<web_service version="1.0">
  <call>
   <call_action>
      <play offset="0s" repeat="0" delay="1s" terminate_digits="#" skip_interval="1s">
         <play_source audio_uri="file://verification/play_menu.wav" audio_type="audio/x-wav" />

         <dvr_setting forward_key="1" backward_key="2" pause_key="3" resume_key="4"
restart_key="5"/>
      </play>
   </call_action>
</call>
</web_service>
```


Multitrack Play


```
<web_service version="1.0">
 <call>
   <call_action>
      <play offset="0s" repeat="0" delay="0s" terminate_digits="#" max_time="infinite"
skip_interval="10s">
          <play_source audio_uri="file://recorded/multi_recorded_file.wav" audio_type="audio/x-
wav" audio_track_id="0"/>
      </play>
   </call_action>
 </call>
</web_service>
```


Response Payload Example

Individual Play


```
<web_service version="1.0">
     <call_response identifier="8fe4c7f4-067c-455d-afee-52f57e00314b" appid="app"
           href="/default/calls/8fe4c7f4-067c-455d-afee-52f57e00314b"
           connected="yes" signaling="yes" cpa="no" call_type="inbound"
```

```
          media="audio"
          dtmf_mode="rfc2833"
          source_uri="sip:Username@10.20.129.113:5060"
          async_dtmf="no" async_tone="no" cleardigits="no" info_ack_mode="automatic"
early_media="yes">
          gusid="b9a5ef23398749bfa6809fae974378f2">
          <call_action>
             <play transaction_id="9d608231-a164-4102-b9e6-3ba1f0671a53"
                   max_time="infinite"
                   fetch_timeout="300s"
                   offset="0s"
                   delay="1s"
                   repeat="0"
                   skip_interval="1s"
                   terminate_digits="#">
                   <play_source audio_uri="file://verification/play_menu.wav"
audio_type="audio/x-wav" />
                   <dvr_setting forward_key="1"
                                backward_key="2"
                                pause_key="3"
                                resume_key="4"
                                restart_key="5"/>
             </play>
          </call_action>
     </call_response>
</web_service>
```

Multitrack Play

```
<web_service version="1.0">
  <call_response identifier="8fe4c7f4-067c-455d-afee-52f57e00314b" appid="app"
          href="/default/calls/8fe4c7f4-067c-455d-afee-52f57e00314b"
          connected="yes" signaling="yes" cpa="no" call_type="inbound"
          media="audio"
          dtmf_mode="rfc2833"
          source_uri="sip:Username@10.20.129.113:5060"
          async_dtmf="no" async_tone="no" cleardigits="no" info_ack_mode="automatic"
early_media="yes">
          gusid="b9a5ef23398749bfa6809fae974378f2">
      <call>
         <call_action>
            <play offset="0s" repeat="0" delay="0s" terminate_digits="#" max_time="infinite"
skip_interval="10s" transaction_id="9d608231-a164-4102-b9e6-3ba1f0671a53">
               <play_source audio_uri="file://recorded/multi_recorded_file.wav"
audio_type="audio/x-wav" audio_track_id="0"/>
            </play>
         </call_action>
      </call>
  </call_response>
```

```
</web_service>
```

update_play

Request Payload Attributes

Parameter	Default	Optional	Description
transaction_id	(none)		Media identifier, returned by play.
dvr_action	(none)		Values: • "backward" - skip backwards. • "forward" - skip forward. • "pause" - pause playback. • "restart" - jump back to the start. • "resume" - resume paused playback.
region	"0"	*	Conference only. The ID of the region that will display the video media. The value "0" causes the video to be shown full screen with the current layout being restored automatically when the play back completes.

Request Payload Example

```
<web_service version="1.0">
  <call>
     <call_action>
        <update_play transaction_id="9d608231-a164-4102-b9e6-3ba1f0671a53" dvr_action="pause"/>
     </call_action>
  </call>
</web_service>
```

Response Payload Example

```
<web_service version="1.0">
     <call_response identifier="8fe4c7f4-067c-455d-afee-52f57e00314b" appid="app"
           href="/default/calls/8fe4c7f4-067c-455d-afee-52f57e00314b"
           connected="yes" signaling="yes" cpa="no" call_type="inbound"
           media="audio"
```

```
            dtmf_mode="rfc2833"
            source_uri="sip:Username@10.20.129.113:5060"
            async_dtmf="no" async_tone="no" cleardigits="no" info_ack_mode="automatic"
early_media="yes">
            gusid="b9a5ef23398749bfa6809fae974378f2">
            <call_action>
                <update_play transaction_id="9d608231-a164-4102-b9e6-3ba1f0671a53"
dvr_action="pause"/>
           </call_action>
      </call_response>
</web_service>
```

record

Request Payload Attributes

Parameter	Default	Optional	Description
recording_audio_mime_params	(none)		Refer to recording_audio_mime_params.
recording_video_mime_params	(none)		Refer to recording_video_mime_params.
terminate_digits	"#"	*	The digit or digits [0-9,*,#] used to terminate the play_record.
recording_uri	(none)		A filename "file://..." which must refer to an existing directory or "http://" ("https://"). **Note:** This is deprecated.
recording_audio_uri	(none)		The URL of the audio media to be recorded e.g., "file://...", "http(s)://...".

Parameter	Default	Optional	Description
recording_audio_type	(none)		The mime type of the audio media. Values: • "audio/x-wav" • "audio/basic" • "audio/x-alaw-basic" • "audio/G723" • "audio/G726" • "audio/G729" • "audio/mp4" • "audio/3gpp" • "audio/AMR" • "audio/AMR-WB" • "audio/mkv" • "audio/x-aud" • "audio/L8" • "audio/L16" • "audio/webm" • "text/uri-list"
recording_video_uri	(none)		The URL of the video media to be recorded e.g., "file://...", "http(s)://...".
recording_video_type	(none)		The mime type of the video media. Values: • "video/x-vid" • "video/3gpp" • "video/mp4" • "video/mkv" • "video/webm"
max_time	"infinite"	*	The maximum length of time to record. Use "infinite" for no limit.
max_silence	"infinite"	*	The maximum length of silence to record after audio has been detected. Use "infinite" for no limit.

Parameter	Default	Optional	Description
noinput_timeout	"infinite"	*	The maximum time to wait for audio to be detected. Use "infinite" for no limit.
clamp_dtmf	"no", "yes" if termination digit is used	*	Determines if dtmf digits are suppressed. Values: • "yes" • "no"
append	"no"	*	Determines if an audio-only recording should be appended to an existing file (file:// uri only). Values: • "yes" • "no"
public_key	(none)	*	RSA public key in pem/base64 format in each encrypted request.

Request Payload Example


```
<web_service version="1.0">
 <call>
  <call_action>
     <record terminate_digits="#" max_time="10s"
                      recording_audio_uri="file://verification/beta_recorded_file.3gp"
                      recording_audio_type="audio/3gpp"
                      recording_video_uri=file://verification/beta_recorded_file.3gp
                      recording_video_type="video/3gpp"  >
             <recording_video_mime_params codec="h264" level="3.1" height="480" width="640"
maxbitrate="768000" framerate="15"/>
             <recording_audio_mime_params codec="AMR" />
      </record>
  </call_action>
 </call>
</web_service>
```


Response Payload Example


```
<web_service version="1.0">
<call_response identifier="4de2358f-4e2e-41c1-ba65-94a6f400f6b6" appid="app"
           href="/default/calls/4de2358f-4e2e-41c1-ba65-94a6f400f6b6"
           connected="yes" signaling="yes" cpa="no" call_type="inbound"
           media="audiovideo"
           dtmf_mode="rfc2833"
```

```
            destination_uri="sip:app@10.20.123.100"
            source_uri="sip:linphone@10.20.126.15"
            async_dtmf="yes" async_tone="yes" cleardigits="no" encryption="none" ice="no"
info_ack_mode="automatic" hangup_ack_mode="automatic" early_media="no" audio="sendrecv"
video="sendrecv">
            gusid="b9a5ef23398749bfa6809fae974378f2">
            <call_action>
                 <record transaction_id="80fa5484-ba92-4446-9d05-0dc58e6b7369"
                         terminate_digits="#"
                         max_time="10s"
                         max_silence="infinite"
                         noinput_timeout="infinite"
                         recording_audio_uri="file://verification/beta_recorded_file.3gp"
                         recording_audio_type="audio/3gpp"
                         recording_video_type="video/3gpp"
                         recording_video_uri="file://verification/beta_recorded_file.3gp">
                         <recording_audio_mime_params codec="AMR"/>
                         <recording_video_mime_params codec="h264" level="3.1" framerate="15"
maxbitrate="768000" height="480" width="640"/>
                 </record>
            </call_action>
      </call_response>
</web_service>
```


Note: To record audio and video on a single file, recording_audio_uri and recording_video_uri must use the same URI (file name) and type (.3gpp, .mkv, .mp4) as shown in the previous request payload example and response payload example.

Public Key Example


```
<web_service version="1.0">
 <call>
  <call_action>
<record max_time="10s" public_key="-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxq1dDrwpaDhnkYCe7xZS
7qHYcXDo6bNNz54/gipWtL1nH2ArGZvAPMuLS5ADkbQdOSrHEd8tu1ziFpBt6Va8
mzLMFQ7o/q8VmBHFDXCLCWHDFZEVGfBTmryrrtMK1Cxj4r19SRZ3lneWd0HKIfyY
OWADzGDUsnEQXtX+o98xLQHh4sichnTvCLegCZrogZsVsi3uvUlUh1v1L0toje/X
uWuN6ZjloHmWIUS9MLmtQejhrJM9xrKnMbQF0MvMor5Iff3L0OcGmdYi2CFgksV0
ma5FRMYqoGL/LeszhZqwN7E6Zkh56AG5mMNuqB2XwdNv9lBKdROlZFFueHri7HT8
hwIDAQAB
-----END PUBLIC KEY-----"
recording_audio_type="audio/opus"
recording_audio_uri="file:///testing/mxml/recordings/mt_caller_1.webm" terminate_digits="#">
      </record>
  </call_action>
 </call>
</web_service>
```

multi_record

Record media from multiple sources to a multitrack container. The recording is terminated when the last source is destroyed or after an optional timeout. The recording may be cancelled using the stop API.

Note: Only 2 track audio/x-wav is currently supported for multi_record.

Request Payload Attributes

Parameter	Default	Optional	Description
recording_audio_mime_params	(none)		Refer to recording_audio_mime_params.
recording_video_mime_params	(none)		Refer to recording_video_mime_params (not currently supported).
record_track	(none)		Refer to record_track.
terminate_digits	"#"	*	The digit or digits [0-9,*,#] used to terminate the play_record.
recording_uri	(none)		A filename "file://..." which must refer to an existing directory or "http://" ("https://"). See also media. **Note:** This is deprecated.
recording_audio_uri	(none)		The URL of the audio media to be recorded e.g. "file://...", "http(s)://...".

<table>
<tr><th>Parameter</th><th>Default</th><th>Optional</th><th>Description</th></tr>
<tr><td>recording_audio_type</td><td>(none)</td><td></td><td>The mime type of the audio media.
Values:
"audio/x-wav"
"audio/basic"
"audio/x-alaw-basic"
"audio/L8"
"audio/L16"
"audio/x-aud"
"audio/AMR"
"audio/AMR-WB"
"audio/3gpp"
"audio/mp4"
"audio/mkv"
"audio/webm"
"text/uri-list"</td></tr>
<tr><td>recording_video_uri</td><td>(none)</td><td></td><td>The URL of the video media to be recorded e.g., "file://...", "http(s)://..." (not currently supported).</td></tr>
<tr><td>recording_video_type</td><td>(none)</td><td></td><td>The mime type of the video media (not currently supported).
Values:
"video/x-vid"
"video/3gpp"
"video/mp4"
"video/mkv"
"video/webm"</td></tr>
<tr><td>max_time</td><td>"infinite"</td><td>*</td><td>The maximum length of time to record. Use "infinite" for no limit.</td></tr>
<tr><td>max_silence</td><td>"infinite"</td><td>*</td><td>The maximum length of silence to record after audio has been detected. Use "infinite" for no limit.</td></tr>
</table>

Parameter	Default	Optional	Description
noinput_timeout	"infinite"	*	The maximum time to wait for audio to be detected. Use "infinite" for no limit.
clamp_dtmf	"no" , "yes" if termination digit is used	*	Determines if dtmf digits are suppressed. Values: • "yes" • "no"
append	"no"	*	Determines if an audio-only recording should be appended to an existing file (file:// uri only). Values: • "yes" • "no"
public_key	(none)	*	RSA public key in pem/base64 format in each encrypted request.

Request Payload Example


```
<web_service version="1.0">
  <call>
      <call_action>
         <multi_record terminate_digits="#" max_time="10s"
                recording_audio_uri="file://verification/multi_recorded_file.wav"
                recording_audio_type="audio/x-wav">
                <record_track id="b9f8695b-e017-4dfa-bb81-002a8e57c169" media="audio"/>
                <record_track id="87b13b33-6558-4191-a1e0-db0052efaa63" media="audio"/>
         </multi_record>
      </call_action>
   </call>
</web_service>
```


Response Payload Example


```
<web_service version="1.0">
            <call_response identifier="b9f8695b-e017-4dfa-bb81-002a8e57c169" appid="app"
                           href="/default/calls/7b3652ed-594f-482d-a349-7625c7cf86ef"
                           connected="yes" signaling="yes" cpa="no" call_type="inbound"
                           media="audiovideo"
                           dtmf_mode="rfc2833"
                           destination_uri="sip:sip@10.20.129.61"
                           source_uri="sip:Username@10.20.129.115:5060"
```

```
                          async_dtmf="yes" async_tone="yes" cleardigits="no" encryption="none"
ice="no" info_ack_mode="automatic" audio="sendrecv" video="sendrecv"
                          gusid="b9a5ef23398749bfa6809fae974378f2">
                 <call_action>
                        <multi_record transaction_id="b3407431-801f-4a44-ba8b-fa8c9fddd653"
                                 terminate_digits="#"
                                 max_time="10s"
                                 max_silence="infinite"
                                 noinput_timeout="infinite"

recording_audio_uri="file://verification/multi_recorded_file.wav"
                                 recording_audio_type="audio/x-wav"
                                 <record_track id="b9f8695b-e017-4dfa-bb81-002a8e57c169"
media="audio" track_id="0"/>
                                <record_track id="87b13b33-6558-4191-a1e0-db0052efaa63"
media="audio" track_id="1"/>
                        </multi_record>
                 </call_action>
         </call_response>
</web_service>
```

Public Key Example

```
<web_service version="1.0">
 <call>
  <call_action>
<multi_record max_time="10s" public_key="-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxq1dDrwpaDhnkYCe7xZS
7qHYcXDo6bNNz54/gipWtL1nH2ArGZvAPMuLS5ADkbQdOSrHEd8tu1ziFpBt6Va8
mzLMFQ7o/q8VmBHFDXCLCWHDFZEVGfBTmryrrtMK1Cxj4r19SRZ3lneWd0HKIfyY
OWADzGDUsnEQXtX+o98xLQHh4sichnTvCLegCZrogZsVsi3uvUlUh1v1L0toje/X
uWuN6ZjloHmWIUS9MLmtQejhrJM9xrKnMbQF0MvMor5Iff3L0OcGmdYi2CFgksV0
ma5FRMYqoGL/LeszhZqwN7E6Zkh56AG5mMNuqB2XwdNv9lBKdROlZFFueHri7HT8
hwIDAQAB
-----END PUBLIC KEY-----"
recording_audio_type="audio/opus"
recording_audio_uri="file:///testing/mxml/recordings/mt_caller_1.webm" terminate_digits="#">
<record_track id="830b9fda-d89e-495d-b7a1-6a63402bcdcf" track_id="0"/>
<record_track id="87b13b33-6558-4191-a1e0-db0052efaa63" track_id="1"/>
      </multi_record>
  </call_action>
 </call>
</web_service>
```

update_multi_record

Update an active multitrack recording. It is used to remove a source from a track or add a source to a track. If adding a source, the current source needs to be stopped using id="none" and then calling update_multi_record again with the id of the new source.

Request Payload Attributes

Parameter	Default	Optional	Description
transaction_id	(none)		Media identifier, returned by multi_record.
id	(none)		Call or conference id. Source is removed from the recording. The recording continues with silence until another termination condition is reached or the record is stopped with stop action.
track_id	(none)		Record track identifier, returned by multi_record.
direction	recv	*	For calls only, this specifies the direction of the stream to be recorded. For conference, only "recv" direction is supported which is the output of the conference. Values: • "send" • "recv"

Request Payload Example

```
<web_service version="1.0">
  <call>
     <call_action>
        <update_multi_record id="b9f8695b-e017-4dfa-bb81-002a8e57c169" transaction_id="9d608231-
a164-4102-b9e6-3ba1f0671a53" direction="send" track_id="0"/>
     </call_action>
  </call>
</web_service>
```

Response Payload Example

```
<web_service version="1.0">
     <call_response identifier="b9f8695b-e017-4dfa-bb81-002a8e57c169" appid="app"
           href="/default/calls/b9f8695b-e017-4dfa-bb81-002a8e57c169"
           connected="yes" signaling="yes" cpa="no" call_type="inbound"
           media="audio"
           dtmf_mode="rfc2833"
           source_uri="sip:Username@10.20.129.113:5060"
           async_dtmf="no" async_tone="no" cleardigits="no" info_ack_mode="automatic"
early_media="yes">
           gusid="b9a5ef23398749bfa6809fae974378f2">
           <call_action>
```

```
            <update_multi_record id="b9f8695b-e017-4dfa-bb81-002a8e57c169" transaction_id="b3407431-801f-4a44-ba8b-fa8c9fddd653" direction="send" track_id="0"/>
        </call_action>
    </call_response>
</web_service>
```

playrecord

Request Payload Attributes

Parameter	Default	Optional	Description
play_source	(none)		Refer to play_source_attributes.
recording_audio_mime_params	(none)		Refer to recording_audio_mime_params.
recording_video_mime_params	(none)		Refer to recording_video_mime_params.
offset	"0"	*	Specifies the time offset from where the play should start. Note, the 'offset' is applied to the initial play only.
repeat	"0"	*	Number of times to repeat the play. "file://" URIs only.
delay	"1s"	*	Time delay between repeated plays.
terminate_digits	"#"	*	The digit or digits [0-9,*,#] used to terminate the play_record.
recording_uri	(none)		A filename "file://..." which must refer to an existing directory. **Note:** This is deprecated.
recording_audio_uri	(none)		The URL of the audio media to be recorded e.g., "file://...", "http(s)://...".

Parameter	Default	Optional	Description
recording_audio_type	(none)		The mime type of the audio media. Values: • "audio/x-wav" • "audio/basic" • "audio/x-alaw-basic" • "audio/L8" • "audio/L16" • "audio/x-aud" • "audio/AMR" • "audio/AMR-WB" • "audio/3gpp" • "audio/mp4" • "audio/mkv" • "audio/webm" • "text/uri-list"
recording_video_uri	(none)		The URL of the video media to be recorded e.g., "file://...", "http(s)://...".
recording_video_type	(none)		The mime type of the video media. Values: • "video/x-vid" • "video/3gpp" • "video/mp4" • "video/mkv" • "video/webm"
beep	"yes"	*	Play a tone before starting to record. Values: • "yes" • "no"
max_time	"infinite"	*	The maximum length of time to record. Use "infinite" for no limit.

Parameter	Default	Optional	Description
max_silence	"infinite"	*	The maximum length of silence to record after audio has been detected. Use "infinite" for no limit. This feature is supported for calls only.
noinput_timeout	"infinite"	*	The maximum time to wait for audio to be detected. Use "infinite" for no limit. This feature is supported for calls only.
barge	"yes"	*	Specifies whether dtmf digit input will barge the prompt and force transition to the record phase. Note that if the "barge" attribute is set to "no", the "cleardigits" attribute implicitly has the value "yes". Values: • "yes" • "no"
cleardigits	"no"	*	Specifies whether previous input should be considered or ignored for the purpose of barge-in. When it is set to "yes", any previously buffered digits are discarded. If it is set to "no", previously buffered digits will be considered. If "cleardigits" is set to "no" and "barge" is set to "yes", previously buffered digits will result in the recording phase starting immediately, and the prompt will not be played. Values: • "yes" • "no"

Parameter	Default	Optional	Description
no_cache		*	Cache-control for HTTP URIs. Values: • "yes" does not use caching. • "no" uses caching. The default is set by the configuration.
max_age		*	Cache-control for HTTP URIs, the maximum age, in seconds, of a cached file. The default is set by the configuration.
max_stale		*	Cache-control for HTTP URIs, the number of seconds that a cached file may exceed its expiration time by and still be a considered as fresh. The default is set by the configuration.
fetch_timeout	"300s"	*	HTTP URIs, the maximum time in seconds to retrieve or upload a file. This is the overall period of the transaction.
clamp_dtmf	"no", "yes" if termination digit is used	*	Determines if dtmf digits are suppressed. Values: • "yes" • "no"
public_key	(none)	*	RSA public key in pem/base64 format in each encrypted request.

Request Payload Example

```
<web_service version="1.0">
  <call>
    <call_action>
      <playrecord recording_audio_uri="file://recorded_file.wav" recording_audio_type="audio/x-wav"  max_time="10s" offset="0s" repeat="0" delay="1s" terminate_digits="#" beep="yes" barge="yes" cleardigits="yes" >
            <play_source audio_uri="file://verification/play_menu.wav" audio_type="audio/x-wav"/>
       </playrecord>
    </call_action>
  </call>
</web_service>
```

Response Payload Example


```
<web_service version="1.0">
      <call_response identifier="8fe4c7f4-067c-455d-afee-52f57e00314b" appid="app"
                     href="/default/calls/8fe4c7f4-067c-455d-afee-52f57e00314b"
                     connected="yes" signaling="yes" cpa="no" call_type="inbound"
                     media="audio"
                     dtmf_mode="rfc2833"
                     source_uri="sip:Username@10.20.129.113:5060"
                     async_dtmf="no" async_tone="no" cleardigits="yes" info_ack_mode="automatic"
early_media="yes">
                     gusid="b9a5ef23398749bfa6809fae974378f2">
                     <call_action>
                         <playrecord transaction_id="a88aac4a-dd73-4a3f-8268-792b90a5efb2"
                                     fetch_timeout="300s"
                                     terminate_digits="#"
                                     max_time="10s"
                                     beep="yes"
                                     barge="yes"
                                     cleardigits="yes"
                                     max_silence="infinite"
                                     noinput_timeout="infinite"
                                     recording_audio_uri="file://recorded_file.wav"
                                     recording_audio_type="audio/x-wav"
                                     offset="0s"
                                     delay="1s"
                                     repeat="0"
                                     terminate_digits="">
                                     <play_source audio_uri="file://verification/play_menu.wav"
                                                  audio_type="audio/x-wav"/>
                         </playrecord>
                     </call_action>
       </call_response>
</web_service>
```


Public Key Example


```
<web_service version="1.0">
 <call>
  <call_action>
<playrecord max_time="10s" public_key="-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxq1dDrwpaDhnkYCe7xZS
7qHYcXDo6bNNz54/gipWtL1nH2ArGZvAPMuLS5ADkbQdOSrHEd8tu1ziFpBt6Va8
mzLMFQ7o/q8VmBHFDXCLCWHDFZEVGfBTmryrrtMK1Cxj4r19SRZ3lneWd0HKIfyY
OWADzGDUsnEQXtX+o98xLQHh4sichnTvCLegCZrogZsVsi3uvUlUh1v1L0toje/X
uWuN6ZjloHmWIUS9MLmtQejhrJM9xrKnMbQF0MvMor5Iff3L0OcGmdYi2CFgksV0
ma5FRMYqoGL/LeszhZqwN7E6Zkh56AG5mMNuqB2XwdNv9lBKdROlZFFueHri7HT8
hwIDAQAB
-----END PUBLIC KEY-----"
```

```
recording_audio_type="audio/opus"
recording_audio_uri="file:///testing/mxml/recordings/mt_caller_1.webm" terminate_digits="#">
      </playrecord>
  </call_action>
 </call>
</web_service>
```

playcollect

Request Payload Attributes

Parameter	Default	Optional	Description
play_source	(none)		Refer to play_source_attributes.
offset	"0"	*	Specifies the time offset from where the play should start. Note, the "offset" is applied to the initial play only.
repeat	"0"	*	Number of times to repeat the play. Use "infinite" to repeat indefinitely. "file://" URIs only.
delay	"1s"	*	Time delay between repeated plays.
term_digits	"#"	*	The digit or digits [0-9,*,#] used to terminate the playcollect.
max_digits	Unlimited	*	The maximum number of digits to collect.
timeout	"infinite"	*	The maximum length of time to wait for the first digit or a tone. This time begins when the prompt phase ends.
interdigit_timeout	The value specified by the *timeout* parameter	*	The maximum length of time to wait for subsequent digits. This timeout is reset after each digit is received.

Parameter	Default	Optional	Description
tone_detection	"no"	*	Enable tone detection. Values: • "yes" • "no"
barge	"yes"	*	Specifies whether dtmf digit input will barge the prompt and force transition to the collect phase. Note that if the "barge" attribute is set to "no", the "cleardigits" attribute implicitly has the value "yes". Values: • "yes" • "no"
cleardigits	"no"	*	Specifies whether previous input should be considered or ignored for the purpose of barge-in and digit matching. When it is set to "yes", any previously buffered digits are discarded. If it is set to "no", previously buffered digits will be considered. If "cleardigits" is set to "no" and "barge" is set to "yes", previously buffered digits will result in the collection phase starting immediately, and the prompt will not be played. Values: • "yes" • "no"

Parameter	Default	Optional	Description
no_cache		*	Cache-control for HTTP URIs. Values: • "yes" does not use caching. • "no" uses caching. The default is set by the configuration.
max_age		*	Cache-control for HTTP URIs, the maximum age, in seconds, of a cached file. The default is set by the configuration.
max_stale		*	Cache-control for HTTP URIs, the number of seconds that a cached file may exceed its expiration time by and still be a considered as fresh. The default is set by the configuration.
fetch_timeout	"300s"	*	HTTP URIs, the maximum time in seconds to retrieve the file. This is the overall period of the transaction.

Request Payload Example

```
<web_service version="1.0">
  <call>
     <call_action>
        <playcollect max_digits="4" timeout="10s" offset="0s" repeat="0" delay="1s"
terminate_digits="#"
                   tone_detection="yes" barge="yes" cleardigits="yes">
                   <play_source audio_uri="file://verification/play_menu.wav"
audio_type="audio/x-wav"/>
        </playcollect>
     </call_action>
  </call>
</web_service>
```

Response Payload Example


```
<web_service version="1.0">
            <call_response identifier="c87fcca1-b2d0-49c5-8b89-baaae71cf695" appid="app"
                           href="/default/calls/c87fcca1-b2d0-49c5-8b89-baaae71cf695"
                           connected="yes" signaling="yes" cpa="no" call_type="inbound"
                           media="audiovideo"
                           dtmf_mode="rfc2833"
                           destination_uri="sip:sip@10.20.129.100"
                           source_uri="sip:Username@10.20.129.113:5060"
                           async_dtmf="yes" async_tone="yes" cleardigits="yes"
info_ack_mode="automatic"
                           audio="sendrecv" video="sendrecv">
                           gusid="b9a5ef23398749bfa6809fae974378f2">
                           <call_action>
                                <playcollect transaction_id="25608a36-2fd1-43ca-b741-
3420dbc389cb"
                                             fetch_timeout="300s"
                                             terminate_digits="#"
                                             timeout="10s"
                                             interdigit_timeout=""
                                             tone_detection="yes"
                                             max_digits="4"
                                             barge="yes"
                                             offset="0s"
                                             delay="1s"
                                             repeat="0"
                                             cleardigits="yes">
                                             <play_source
audio_uri="file://verification/play_menu.wav"
                                                          audio_type="audio/x-wav"/>
                                </playcollect>
                           </call_action>
          </call_response>
</web_service>
```


overlay

Request Payload Attributes

Parameter	Default	Optional	Description
uri	(none)		The template parameters passed to the image builder e.g., "image:"id=template&a=b...".
duration	"infinite"	*	Length of time that the overlay is shown. Use "infinite" without the quotes to display the overlay until explicitly stopped.

Parameter	Default	Optional	Description
direction	"send"	*	Stream direction to which the overlay is applied. Values: • "send" • "recv"

Request Payload Example


```
<web_service version="1.0">
  <call>
   <call_action>
      <overlay uri="image:id=menu&header=Menu du Jour&items=1 Pie&items=2 Chips&items=3
Burger&items=4
                    Pizza&items=5 Jacket&items=6 Panini&items=7 Pasta&footer=Tuesday"
               duration="15s"/>
   </call_action>
</call></web_service>
```


Response Payload Example


```
<web_service version="1.0">
     <call_response identifier="8fe4c7f4-067c-455d-afee-52f57e00314b" appid="app"
           href="/default/calls/8fe4c7f4-067c-455d-afee-52f57e00314b"
           connected="yes" signaling="yes" cpa="no" call_type="inbound"
           media="audio"
           dtmf_mode="rfc2833"
           source_uri="sip:Username@10.20.129.113:5060"
           async_dtmf="no" async_tone="no" cleardigits="no" info_ack_mode="automatic"
early_media="yes"
           gusid="b9a5ef23398749bfa6809fae974378f2">
           <call_action>
               <overlay transaction_id="c4023bb2-4062-415b-ae79-2e708458cfdd"
                          uri="image:id=menu&header=Menu du Jour&items=1 Pie&items=2
Chips&items=3
                             Burger&items=4 Pizza&items=5 Jacket&items=6 Panini&items=7
                             Pasta&footer=Tuesday"
                          duration="15s"/>
          </call_action>
     </call_response>
</web_service>
```

send_dtmf

Request Payload Attributes

Parameter	Default	Optional	Description
digits			Digit(s) to send [1234567890*#ABCD].
duration	100ms	*	Length of time of each digit.
interval	100ms	*	Time between successive digits.
level	-10dB	*	Amplitude of the dtmf digit tones. Range: 0 to -40dB.

Request Payload Example


```
<web_service version="1.0">
  <call>
   <call_action>
      <send_dtmf digits="2345" />
   </call_action>
  </call>
</web_service>
```


Response Payload Example


```
<web_service version="1.0">
     <call_response identifier="8fe4c7f4-067c-455d-afee-52f57e00314b" appid="app"
           href="/default/calls/8fe4c7f4-067c-455d-afee-52f57e00314b"
           connected="yes" signaling="yes" cpa="no" call_type="inbound"
           media="audio"
           dtmf_mode="rfc2833"
           source_uri="sip:Username@10.20.129.113:5060"
           async_dtmf="no" async_tone="no" cleardigits="no" info_ack_mode="automatic"
early_media="yes">
           gusid="b9a5ef23398749bfa6809fae974378f2">
           <call_action>
               <send_dtmf transaction_id="c4023bb2-4062-415b-ae79-2e708458cfdd"
                          digits="2345" duration="100ms" interval="100ms" level="-10dB"/>
          </call_action>
     </call_response>
</web_service>
```


send_event

Send one or more telephony events in RTP packets. This supports RFC 4733.

Request Payload Attributes

Parameter	Default	Optional	Description
events			Event(s) to send [0123456789]. For values higher than 9, use "$xxx" where "xxx" is the three digit decimal code. For example, to send digits 0, 1, and 4 and event 16, the digits attribute would be set to digits="014$016".
duration	100ms	*	Length of time of each digit.
interval	100ms	*	Time between successive digits.
level	-10dB	*	Amplitude of the dtmf digit tones. Range: 0 to -40dB.

Request Payload Example

```
<web_service version="1.0">
  <call>
   <call_action>
      <send_event events="2345$016" />
   </call_action>
</call></web_service>
```

Response Payload Example

```
<web_service version="1.0">
     <call_response identifier="8fe4c7f4-067c-455d-afee-52f57e00314b" appid="app"
           href="/default/calls/8fe4c7f4-067c-455d-afee-52f57e00314b"
           connected="yes" signaling="yes" cpa="no" call_type="inbound"
           media="audio"
           dtmf_mode="rfc2833"
           source_uri="sip:Username@10.20.129.113:5060"
           async_dtmf="no" async_tone="no" cleardigits="no" info_ack_mode="automatic"
early_media="yes">
           gusid="b9a5ef23398749bfa6809fae974378f2">
           <call_action>
               <send_event transaction_id="c4023bb2-4062-415b-ae79-2e708458cfdd"
                            events="2345$016" duration="100ms" interval="100ms" level="-10dB"/>
          </call_action>
     </call_response>
</web_service>
```

stop

Request Payload Attributes

Parameter	Default	Optional	Description
transaction_id	(none)		Identifier returned by play, playcollect, playrecord, record, or send_dtmf call action.

Request Payload Example

```
<web_service version="1.0">
  <call>
    <call_action>
       <stop transaction_id="c4023bb2-4062-415b-ae79-2e708458cfdd"/>
    </call_action>
  </call>
</web_service>
```

Response Payload Example

```
<web_service version="1.0">
     <call_response identifier="8fe4c7f4-067c-455d-afee-52f57e00314b" appid="app"
           href="/default/calls/8fe4c7f4-067c-455d-afee-52f57e00314b"
           connected="yes" signaling="yes" cpa="no" call_type="inbound"
           media="audio"
           dtmf_mode="rfc2833"
           source_uri="sip:Username@10.20.129.113:5060"
           async_dtmf="no" async_tone="no" cleardigits="no" info_ack_mode="automatic"
early_media="yes">
           gusid="b9a5ef23398749bfa6809fae974378f2">
           <call_action>
              <stop transaction_id="c4023bb2-4062-415b-ae79-2e708458cfdd"/>
          </call_action>
     </call_response>
</web_service>
```

add_party/update_party

Request Payload Attributes

Parameter	Default	Optional	Description
conf_id	(none)		Identifier of the conference to join.
caption	(none)	*	Text for caption e.g., caller name.

Parameter	Default	Optional	Description
caption_duration	The value specified in create conference	*	The length of time that the caption is shown. Use "infinite" without the quotation marks to display the caption for the entire call.
region	"0"	*	The ID of the region used to display this participant's video stream. The value "0" means no preference. The current occupant of the region (if any) will be reset to *no* preference and replaced by this party.
audio	"recvonly"	*	Sets the conference audio participation. Values: • "inactive" - No audio. • "sendonly" - Only transmit audio. • "recvonly" - Only receive audio. • "sendrecv" - Full duplex audio.

Parameter	Default	Optional	Description
video	"recvonly"	*	Sets the conference video participation. Values: • "inactive" - No video. • "sendonly" - Only transmit video. • "recvonly" - Only receive video. • "sendrecv" - Full duplex video.
clamp_dtmf	The value specified in create conference	*	Determines if dtmf digits are suppressed. Values: • "yes" • "no"
auto_gain_control	The value specified in create conference	*	Determines if automatic gain control should be used. Values: • "yes" • "no"
echo_cancellation	The value specified create conference	*	Determines if echo cancellation should be used. Values: • "yes" • "no"

Parameter	Default	Optional	Description
mute	"no"	*	Mutes or unmutes the audio stream from this party. Values: • "yes" mute the stream. • "no" unmute the stream.
tx_mute	"no"	*	Mutes or unmutes the audio stream to this party. Values: • "yes" mute the stream. • "no" unmute the stream.
privilege	"no"	*	Enables or disables privilege talker. When set to "yes", party is always included in the conference summation output process, providing its speech level is greater than zero. Values: • "yes" • "no"

Parameter	Default	Optional	Description
mode	"normal"	*	Determines the mixing for the party. Values: • "normal" • "coach" sets party as a coach, the coach is heard by pupil only. **Note:** Only one coach per conference. • "pupil" sets this party as a pupil, the pupil hears everyone including the coach.
primary_video_source		*	SFU only. Specifies the primary video source sent to this party. This may be either the string "vas", which automatically selects the video source based on voice activity, or a specific conference party identified using its call_id. **Note:** For add_party if it is n, the default behavior is "vas". **Note:** For a list of SFU limitations and precautions, see the *Dialogic® PowerMedia® XMS Release 3.3 Release Notes*.

Request Payload Example

add_party

```
<web_service version="1.0">
  <call>
    <call_action>
      <add_party conf_id="830b9fda-d89e-495d-b7a1-6a63402bcdcf" caption="Username" region="0"
                 audio="sendrecv" video="sendrecv"/>
    </call_action>
  </call>
</web_service>
```

update_party

```
<web_service version="1.0">
  <call>
    <call_action>
      <update_party conf_id="830b9fda-d89e-495d-b7a1-6a63402bcdcf" caption="Frank"/>
    </call_action>
  </call>
</web_service>
```

Response Payload Example

add_party

```
<web_service version="1.0">
     <call_response identifier="8fe4c7f4-067c-455d-afee-52f57e00314b" appid="app"
           href="/default/calls/8fe4c7f4-067c-455d-afee-52f57e00314b"
           connected="yes" signaling="yes" cpa="no" call_type="inbound"
           media="audio"
           dtmf_mode="rfc2833"
           source_uri="sip:Username@10.20.129.113:5060"
           async_dtmf="no" async_tone="no" cleardigits="no" info_ack_mode="automatic"
early_media="yes">
           gusid="b9a5ef23398749bfa6809fae974378f2">
           <call_action>
               <add_party conf_id="830b9fda-d89e-495d-b7a1-6a63402bcdcf" caption="Username"
region="0"
                          audio="sendrecv" video="sendrecv"/>
          </call_action>
     </call_response>
</web_service>
```

update_party

```
<web_service version="1.0">
     <call_response identifier="8fe4c7f4-067c-455d-afee-52f57e00314b" appid="app"
           href="/default/calls/8fe4c7f4-067c-455d-afee-52f57e00314b"
           connected="yes" signaling="yes" cpa="no" call_type="inbound"
           media="audio"
           dtmf_mode="rfc2833"
           source_uri="sip:Username@10.20.129.113:5060"
           async_dtmf="no" async_tone="no" cleardigits="no" info_ack_mode="automatic"
early_media="yes">
           gusid="b9a5ef23398749bfa6809fae974378f2">
           <call_action>
```

```
                <update_party conf_id="830b9fda-d89e-495d-b7a1-6a63402bcdcf" caption="Frank"
region="0"
                               audio="sendrecv" video="sendrecv"/>
          </call_action>
     </call_response>
</web_service>
```

remove_party

Request Payload Attributes

Parameter	Default	Optional	Description
conf_id	(none)		The identifier of the conference to unjoin.

Request Payload

```
<web_service version="1.0">
  <call>
    <call_action>
       <remove_party conf_id="830b9fda-d89e-495d-b7a1-6a63402bcdcf" />
    </call_action>
  </call>
</web_service>
```

Response Payload

```
<web_service version="1.0">
     <call_response identifier="8fe4c7f4-067c-455d-afee-52f57e00314b" appid="app"
           href="/default/calls/8fe4c7f4-067c-455d-afee-52f57e00314b"
           connected="yes" signaling="yes" cpa="no" call_type="inbound"
           media="audio"
           dtmf_mode="rfc2833"
           source_uri="sip:Username@10.20.129.113:5060"
           async_dtmf="no" async_tone="no" cleardigits="no" info_ack_mode="automatic"
early_media="yes">
           gusid="b9a5ef23398749bfa6809fae974378f2">
           <call_action>
              <remove_party conf_id="830b9fda-d89e-495d-b7a1-6a63402bcdcf"/>
          </call_action>
     </call_response>
</web_service>
```

join/unjoin

Request Payload Attributes

Parameter	Default	Optional	Description
call_id	(none)		The identifier of the other party call ID.

Parameter	Default	Optional	Description
audio	"sendrecv"	*	Direction of audio media relative to call_id. Values: • "inactive" • "sendonly" • "recvonly" • "sendrecv"
video	"sendrecv"	*	Direction of video media relative to call_id. Values: • "inactive" • "sendonly" • "recvonly" • "sendrecv"
audio_transcode	"yes"	*	Controls transcoding of audio between the calls (join only). Values: • "yes" • "no"
video_transcode	"yes"	*	Controls transcoding of video between the calls (join only). Values: • "yes" • "no"

Parameter	Default	Optional	Description
media_path_optimize	"no"	*	When both endpoints are using Dialogic WebRTC JavaScript API, media flow can be made p2p with the help of join API, with additional parameter media_path_optimize set to "yes". Unjoin would reinvite media back to XMS (join only). Values: • "yes" • "no"

Request Payload Example

join

```
<web_service version="1.0">
  <call>
    <call_action>
       <join call_id="830b9fda-d89e-495d-b7a1-6a63402bcdcf"/>
    </call_action>
  </call>
</web_service>
```

unjoin

```
<web_service version="1.0">
  <call>
    <call_action>
       <unjoin call_id="830b9fda-d89e-495d-b7a1-6a63402bcdcf"/>
    </call_action>
  </call>
</web_service>
```

Response Payload Example

join

```
<web_service version="1.0">
     <call_response identifier="8fe4c7f4-067c-455d-afee-52f57e00314b" appid="app"
           href="/default/calls/8fe4c7f4-067c-455d-afee-52f57e00314b"
           connected="yes" signaling="yes" cpa="no" call_type="inbound"
           media="audio"
           dtmf_mode="rfc2833"
           source_uri="sip:Username@10.20.129.113:5060"
           async_dtmf="no" async_tone="no" cleardigits="no" info_ack_mode="automatic"
early_media="yes">
```

```
            gusid="b9a5ef23398749bfa6809fae974378f2">
            &lt;call_action&gt;
                &lt;join call_id="830b9fda-d89e-495d-b7a1-6a63402bcdcf" /&gt;
           &lt;/call_action&gt;
     &lt;/call_response&gt;
&lt;/web_service&gt;
```

unjoin

```
&lt;web_service version="1.0"&gt;
     <call_response identifier="8fe4c7f4-067c-455d-afee-52f57e00314b" appid="app"
            href="/default/calls/8fe4c7f4-067c-455d-afee-52f57e00314b"
            connected="yes" signaling="yes" cpa="no" call_type="inbound"
            media="audio"
            dtmf_mode="rfc2833"
            source_uri="sip:Username@10.20.129.113:5060"
            async_dtmf="no" async_tone="no" cleardigits="no" info_ack_mode="automatic"
early_media="yes">
            gusid="b9a5ef23398749bfa6809fae974378f2">
            &lt;call_action&gt;
                &lt;unjoin call_id="830b9fda-d89e-495d-b7a1-6a63402bcdcf"/&gt;
           &lt;/call_action&gt;
     &lt;/call_response&gt;
&lt;/web_service&gt;
```

send_info/send_info_ack

Request Payload Attributes

Parameter	Default	Optional	Description
content_type	(none)		Mime type describing content (optional for send_info_ack).
content	(none)		Data content (optional for send_info_ack).
content_element	(none)	*	Refer to content_element_attributes. Can be repeated n times.
enable_info_ack	"no"	*	Should an info_ack be generated for this request (send_info only). Values: • "yes" • "no"

Request Payload Example

send_info

```
&lt;web_service version="1.0"&gt;
```

```
  <call>
    <call_action>
       <send_info content_type="text/plain" content="data"/>
    </call_action>
  </call>
</web_service>
```

send_info_ack

```
<web_service version="1.0">
  <call>
    <call_action>
       <send_info_ack/>
    </call_action>
  </call>
</web_service>
```

Response Payload Example

send_info

```
<web_service version="1.0">
     <call_response identifier="8fe4c7f4-067c-455d-afee-52f57e00314b" appid="app"
           href="/default/calls/8fe4c7f4-067c-455d-afee-52f57e00314b"
           connected="yes" signaling="yes" cpa="no" call_type="inbound"
           media="audio"
           dtmf_mode="rfc2833"
           source_uri="sip:Username@10.20.129.113:5060"
           async_dtmf="no" async_tone="no" cleardigits="no" info_ack_mode="automatic"
early_media="yes">
           gusid="b9a5ef23398749bfa6809fae974378f2">
           <call_action>
               <send_info content_type="text/plain" content="data"/>
          </call_action>
     </call_response>
</web_service>
```

send_info_ack

```
<web_service version="1.0">
     <call_response identifier="8fe4c7f4-067c-455d-afee-52f57e00314b" appid="app"
           href="/default/calls/8fe4c7f4-067c-455d-afee-52f57e00314b"
           connected="yes" signaling="yes" cpa="no" call_type="inbound"
           media="audio"
           dtmf_mode="rfc2833"
           source_uri="sip:Username@10.20.129.113:5060"
           async_dtmf="no" async_tone="no" cleardigits="no" info_ack_mode="automatic"
early_media="yes">
           gusid="b9a5ef23398749bfa6809fae974378f2">
           <call_action>
               <send_info_ack/>
          </call_action>
     </call_response>
</web_service>
```

send_prack/send_prack_ack/send_answer_ack

send_prack

Send a required SIP PRACK message after a RINGING or PROGRESS event. The prack_mode parameter must be set to "manual" by either creating an outbound call or dialing. The RINGING or PROGRESS event "prack_level" parameter must be "required".

send_prack_ack

Acknowledge a previous PRACK event. It requires that prack_ack_mode parameter to be set to "manual" by accepting a call. The RINGING event must contain the prack_level parameter "required".

send_answer_ack

Acknowledge a "connected" event. It requires that answer_ack_mode parameter to be set to "manual" by either creating an outbound call or dialing.

Request Payload Example

send_prack

```
<web_service version="1.0">
  <call>
    <call_action>
       <send_info content_type="text/plain" content_data="data"/>
    </call_action>
  </call>
</web_service>
```

send_prack_ack

```
<web_service version="1.0">
  <call>
    <call_action>
       <send_prack_ack/>
    </call_action>
  </call>
</web_service>
```

send_answer_ack

```
<web_service version="1.0">
  <call>
    <call_action>
       <send_answer_ack/>
    </call_action>
  </call>
</web_service>
```

Response Payload Example

send_prack

```
<web_service version="1.0">
     <call_response identifier="8fe4c7f4-067c-455d-afee-52f57e00314b" appid="app"
           href="/default/calls/8fe4c7f4-067c-455d-afee-52f57e00314b"
           connected="yes" signaling="yes" cpa="no" call_type="inbound"
```

```
        media="audio"
        dtmf_mode="rfc2833"
        source_uri="sip:Username@10.20.129.113:5060"
        async_dtmf="no" async_tone="no" cleardigits="no" info_ack_mode="automatic"
early_media="yes"
        prack_mode="manual">
        gusid="b9a5ef23398749bfa6809fae974378f2">
        <call_action>
            <send_prack/>
        </call_action>
    </call_response>
</web_service>
```

send_prack_ack

```
<web_service version="1.0">
    <call_response identifier="8fe4c7f4-067c-455d-afee-52f57e00314b" appid="app"
        href="/default/calls/8fe4c7f4-067c-455d-afee-52f57e00314b"
        connected="yes" signaling="yes" cpa="no" call_type="inbound"
        media="audio"
        dtmf_mode="rfc2833"
        source_uri="sip:Username@10.20.129.113:5060"
        async_dtmf="no" async_tone="no" cleardigits="no" info_ack_mode="automatic"
early_media="yes"
        prack_ack_mode="manual">
        gusid="b9a5ef23398749bfa6809fae974378f2">
        <call_action>
            <send_prack_ack/>
        </call_action>
    </call_response>
</web_service>
```

send_answer_ack

```
<web_service version="1.0">
    <call_response identifier="8fe4c7f4-067c-455d-afee-52f57e00314b" appid="app"
        href="/default/calls/8fe4c7f4-067c-455d-afee-52f57e00314b"
        connected="yes" signaling="yes" cpa="no" call_type="inbound"
        media="audio"
        dtmf_mode="rfc2833"
        source_uri="sip:Username@10.20.129.113:5060"
        async_dtmf="no" async_tone="no" cleardigits="no" info_ack_mode="automatic"
early_media="yes"
        answer_ack_mode="manual">
        gusid="b9a5ef23398749bfa6809fae974378f2">
        <call_action>
            <send_answer_ack/>
        </call_action>
    </call_response>
</web_service>
```

send_message

Send application defined data to the remote party. The data may be related to an existing call or standalone.

Request Payload Attributes

Parameter	Default	Optional	Description
msg_payload	(none)	* (required if mode != "signalling")	Refer to msg_payload. Can be repeated *n* times.
content_element	(none)	*	Refer to content_element. Can be repeated *n* times.
mode	"signalling"	*	Message transfer mode. Values: • "signalling" rtc or sip (future). • "msrp" MRSP instant message (rfc4975). Existing call only. • "rfc5547" MSRP file transfer.
uri	(none)	*	Destination address. ("signalling": out-of-call only or "rfc5547").
called_uri	uri	*	Logical destination address. ("signalling": out-of-call only or "rfc5547").
caller_uri	(none)	*	Caller address. ("signalling": out-of-call only or "rfc5547").
content_type	(none)	* (required if mode = "signalling")	Mime type describing content.
content	(none)	* (required if mode = "signalling")	Content data.

Parameter	Default	Optional	Description
msg_multipart_type	(none)	* (required if there is more than 1 msg_payload)	Multipart mime type for the msg_payload[] pseudo array (rfc4975 only).
report	(none)	*	Specifies the report type in the END_SEND_MESSAGE event. Values: • "success" • "failure" • "both"
headers	(none)	*	Raw SIP headers, delimited by the <CR><LF> end-of-line characters ("rfc5547" only). **Note:** This is deprecated since PowerMedia XMS Release 3.0. Use sip_headers.
sip_headers	(none)	*	Refer to sip_headers_attributes.

Request Payload Example


```
<web_service version="1.0">
  <call>
    <call_action>
       <send_message mode="msrp">
           <msg_payload content_id="id1" content_type="text/plain"
content_disposition="attachment">
                  <msg_payload_content content="data"/>
            </msg_payload>
       </send_message>
    </call_action>
  </call>
</web_service>
```


Response Payload Example


```
<web_service version="1.0">
     <call_response identifier="8fe4c7f4-067c-455d-afee-52f57e00314b" appid="app"
           href="/default/calls/8fe4c7f4-067c-455d-afee-52f57e00314b"
           signaling="no" cpa="no" call_type="3pcc"
```

```
            media="message">
            gusid="b9a5ef23398749bfa6809fae974378f2">
            <call_action>
               <send_message mode="msrp" transaction_id="e3510dc1-61c0-489c-b067-5f5c9a9b5b39"
                              report="failure" mode="msrp">
                    <msg_payload content_id="id1" content_type="text/plain"
content_disposition="attachment">
                           <msg_payload_content content="data"/>
                     </msg_payload>
               </send_message>
           </call_action>
      </call_response>
</web_service>
```

dial

Dial an outbound call.

Request Payload Attributes

Parameter	Default	Optional	Description
uri	(none)		Destination address. For SIP, this is the Request-Uri.
called_uri	uri	*	Logical destination address. For SIP, this is the To header.
caller_uri	(none)	*	Caller address. For SIP, this is the From header.
caller_display_name	(none)	*	Caller's display name. For SIP, this is prepended to the From header.
cpa	"no"	*	Use call progress detection. Values: • "yes" • "no"
cpa_profile	(none)	*	Use configuration profile for call progress analysis. See the *Dialogic® PowerMedia™ XMS Installation and Configuration Guide* for more information.

Parameter	Default	Optional	Description
codec_profile	(none)	*	Use codec profile for codec prioritization. See the *Dialogic® PowerMedia™ XMS Installation and Configuration Guide* for more information.
timeout	"30s"	*	Maximum time to wait for the call to be answered by the called party.
content_type	(none)	*	Mime type describing content.
content	(none)	*	Data content.
answer_ack_mode	"automatic"	*	Specifies how CONNECTED events are acknowledged. Values: • "automatic" • "manual"
prack_mode	"automatic"	*	Specifies how PRACK messages are sent. Values: • "automatic" • "manual"
prack_level	"supported"	*	Specifies to the remote if PRACK messages are supported or required. Values: • "supported" • "required"

Parameter	Default	Optional	Description
headers	(none)	*	Raw SIP headers, delimited by the <CR><LF> end-of-line characters (outbound call only). **Note:** This is deprecated since PowerMedia XMS Release 3.0. Use sip_headers.
sip_headers	(none)	*	Refer to sip_headers_attributes.

Request Payload Example


```
<web_service version="1.0">
  <call>
    <call_action>
      <dial uri="sip:b2bua@146.152.122.145:5070;lr">
        <sip_headers>
          <param name="Route"
value="<sip:b2bua@146.152.122.145:5070;lr>,<sip:b2bua@146.152.122.145:5070;lr>"/>
          <param name="Record-Route" value="<sip:b2bua@146.152.122.145:5070;lr>"/>
        </sip_headers>
      </dial>
    </call_action>
  </call>
</web_service>
```


Response Payload Example


```
<web_service version="1.0">
            <call_response identifier="ef5215fd-7a14-4c60-9e5c-f05fc1a57f26" appid="app"
                           href="/default/calls/ef5215fd-7a14-4c60-9e5c-f05fc1a57f26"
                           connected="no" signaling="yes" cpa="no" call_type="outbound"
                           media="audiovideo"
                           dtmf_mode="rfc2833"
                           destination_uri="sip:b2bua@146.152.122.145:5070;lr"
                           async_dtmf="no" async_tone="no" cleardigits="no" encryption="none"
ice="no" info_ack_mode="automatic"
                           hangup_ack_mode="automatic" early_media="no" answer_ack_mode="manual"
prack_mode="manual"
                           prack_level="required">
                           gusid="b9a5ef23398749bfa6809fae974378f2">
                     <sip_headers
raw_sip_headers="Route:<sip:b2bua@146.152.122.145:5070;lr>,<sip:b2bua@146.152.122.145:5070;lr>
Record-Route:<sip:b2bua@146.152.122.145:5070;lr>">
                            <param name="Route"
```

```
value="<sip:b2bua@146.152.122.145:5070;lr>,<sip:b2bua@146.152.122.145:5070;lr>"/>
                            <param name="Record-Route"
                                   value="<sip:b2bua@146.152.122.145:5070;lr>"/>
                     </sip_headers>
                     <call_action>
                          <dial uri = "sip:b2bua@146.152.122.145:5070;lr"
answer_ack_mode="manual"
                                prack_mode="manual"
                                prack_level="required"
                                cpa="no">
                          </dial>
                     </call_action>
            </call_response>
</web_service>
```

transfer

Transfers a call.

- Unattended. The call resource must in a connected state.
- Attended. The call resource must in a connected state and the call identified by call_id must be in a connected state.

Request Payload Attributes

Parameter	Default	Optional	Description
call_id	(none)	*	The call identifier of the consultation call (attended transfer only).
uri	(none)	*	The URI of the transfer target (unattended transfer only).

Request Payload Example

attended

```
<web_service version="1.0">
  <call>
   <call_action>
      <transfer call_id="7de4c7f4-067c-455d-afee-52f57e00314b" />
   </call_action>
</call></web_service>
```

unattended transfer

```
<web_service version="1.0">
  <call>
   <call_action>
      <transfer uri="sip:8001@192.168.195.52" />
   </call_action>
</call></web_service>
```

Response Payload Example

attended transfer


```
<web_service version="1.0">
     <call_response identifier="8fe4c7f4-067c-455d-afee-52f57e00314b" appid="app"
           href="/default/calls/8fe4c7f4-067c-455d-afee-52f57e00314b"
           connected="yes" signaling="yes" cpa="no" call_type="inbound"
           media="audio"
           dtmf_mode="rfc2833"
           source_uri="sip:Username@10.20.129.113:5060"
           async_dtmf="no" async_tone="no" cleardigits="no" info_ack_mode="automatic"
early_media="yes">
           gusid="b9a5ef23398749bfa6809fae974378f2">
           <call_action>
               <transfer call_id="7de4c7f4-067c-455d-afee-52f57e00314b" />
          </call_action>
     </call_response>
</web_service>
```


unattended transfer


```
<web_service version="1.0">
     <call_response identifier="8fe4c7f4-067c-455d-afee-52f57e00314b" appid="app"
           href="/default/calls/8fe4c7f4-067c-455d-afee-52f57e00314b"
           connected="yes" signaling="yes" cpa="no" call_type="inbound"
           media="audio"
           dtmf_mode="rfc2833"
           source_uri="sip:Username@10.20.129.113:5060"
           async_dtmf="no" async_tone="no" cleardigits="no" info_ack_mode="automatic"
early_media="yes">
           gusid="b9a5ef23398749bfa6809fae974378f2">
           <call_action>
               <transfer uri="sip:8001@192.168.195.52" />
          </call_action>
     </call_response>
</web_service>
```


redirect

Redirects an incoming call to another URI.

Request Payload Attributes

Parameter	Default	Optional	Description
uri	(none)		The URI of the redirection target.

Request Payload Example

```
<web_service version="1.0">
  <call>
    <call_action>
       <redirect uri="sip:8001@192.168.195.52"/>
    </call_action>
  </call>
</web_service>
```

Response Payload Example

```
<web_service version="1.0">
     <call_response identifier="8fe4c7f4-067c-455d-afee-52f57e00314b" appid="app"
           href="/default/calls/8fe4c7f4-067c-455d-afee-52f57e00314b"
           connected="yes" signaling="yes" cpa="no" call_type="inbound"
           media="audio"
           dtmf_mode="rfc2833"
           source_uri="sip:Username@10.20.129.113:5060"
           async_dtmf="no" async_tone="no" cleardigits="no" info_ack_mode="automatic"
early_media="yes">
           gusid="b9a5ef23398749bfa6809fae974378f2">
           <call_action>
              <redirect uri="sip:8001@192.168.195.52"/>
          </call_action>
     </call_response>
</web_service>
```

hangup

Ends a call.

- If the call is *joined* to another call, the other party will be automatically unjoined.
- If the call is in a conference, it will be removed from the conference automatically.

Request Payload Attributes

Parameter	Default	Optional	Description
reason	(none)	*	Why the call is being cleared e.g., "404 Not Found". This is only supported for calls that have not been answered.
content_type	(none)	*	Mime type describing content.
content	(none)	*	Data content.

Request Payload Example

```
<web_service version="1.0">
  <call>
    <call_action>
       <hangup content_type="text/plain" content="data"/>
    </call_action>
  </call>
</web_service>
```

Response Payload Example

```
<web_service version="1.0">
     <call_response identifier="8fe4c7f4-067c-455d-afee-52f57e00314b" appid="app"
           href="/default/calls/8fe4c7f4-067c-455d-afee-52f57e00314b"
           connected="yes" signaling="yes" cpa="no" call_type="inbound"
           media="audio"
           dtmf_mode="rfc2833"
           source_uri="sip:Username@10.20.129.113:5060"
           async_dtmf="no" async_tone="no" cleardigits="no" info_ack_mode="automatic"
early_media="yes">
           gusid="b9a5ef23398749bfa6809fae974378f2">
           <call_action>
              <hangup content_type="text/plain" content="data"/>
          </call_action>
     </call_response>
</web_service>
```

send_hangup_ack

Acknowledges the previous hangup event. The use of this API requires that hangup_ack_mode is set to *manual* by either create or answer or accept call.

Request Payload Attributes

Parameter	Default	Optional	Description
content_type	(none)	*	Mime type describing content.
content	(none)	*	Data content.

Request Payload Example

```
<web_service version="1.0">
  <call>
    <call_action>
       <send_hangup_ack content_type="text/plain" content="data"/>
    </call_action>
  </call>
</web_service>
```

Response Payload Example

```
<web_service version="1.0">
     <call_response identifier="8fe4c7f4-067c-455d-afee-52f57e00314b" appid="app"
           href="/default/calls/8fe4c7f4-067c-455d-afee-52f57e00314b"
           connected="yes" signaling="yes" cpa="no" call_type="inbound"
           media="audio"
           dtmf_mode="rfc2833"
           source_uri="sip:Username@10.20.129.113:5060"
           async_dtmf="no" async_tone="no" cleardigits="no" info_ack_mode="automatic"
early_media="yes">
           gusid="b9a5ef23398749bfa6809fae974378f2">
           <call_action>
              <send_hangup_ack content_type="text/plain" content="data"/>
          </call_action>
     </call_response>
</web_service>
```

get_call_info

Queries the call information.

Request Payload Example

```
<web_service version="1.0">
  <call>
    <call_action>
       <get_call_info/>
    </call_action>
  </call>
</web_service>
```

Response Payload Attributes

Parameter	Description
local_sdp	Local session description protocol.
remote_sdp	Remote session description protocol.
media	The media types supported by the call. Values: • "audio" • "audiovideo" • "video"

Parameter	Description
audio	Direction of audio media. Values: • "inactive" • "sendonly" • "recvonly" • "sendrecv"
video	Direction of video media. Values: • "inactive" • "sendonly" • "recvonly" • "sendrecv"
uri	The requested URI.
caller_uri	The callers URI ("To:" header for inbound calls, "From:" header for outbound calls).
called_uri	The called URI ("From:" header for inbound calls, "To:" header for outbound calls).
sip_headers	Refer to sip_headers_attributes.
application_id	ID of the controlling application.
local_seqno	Local SIP sequence number.
remote_seqno	Name of the controlling application.

Response Payload Example


```
<web_service version="1.0">

<call_response identifier="8be5d67d-07c7-4d18-a6ce-d44634a9f048" appid="app"
href="/default/calls/8be5d67d-07c7-4d18-a6ce-d44634a9f048"

connected="yes" signaling="yes" cpa="no" call_type="inbound"

media="audiovideo"

dtmf_mode="rfc2833"

destination_uri="sip:sip@10.20.129.61"

source_uri="sip:Username@10.20.129.115:5060"

async_dtmf="yes" async_tone="yes" cleardigits="no" encryption="none" ice="no"
info_ack_mode="automatic" audio="sendrecv" video="sendrecv">

gusid="b9a5ef23398749bfa6809fae974378f2">

<call_action>

<get_call_info local_sdp="v=0

o=xmserver 1377282511 1377282512 IN IP4 10.20.129.61

s=xmserver
```

```
c=IN IP4 10.20.129.61
t=0 0
m=audio 49152 RTP/AVP 0 101
a=rtpmap:0 pcmu/8000
a=sendrecv
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
m=video 57344 RTP/AVP 34
b=AS:128
a=rtpmap:34 h263/90000
a=fmtp:34 QCIF=2
a=sendrecv
"
         remote_sdp="v=0
o=Username 1377205528 1377282503 IN IP4 10.20.129.115
s=Kapanga [1377205528]
c=IN IP4 10.20.129.115
t=0 0
m=audio 5106 RTP/AVP 0 8 101
a=rtpmap:0 pcmu/8000
a=sendrecv
a=rtcp:5107
a=maxptime:20
a=ptime:20
a=rtpmap:8 pcma/8000
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15,36
m=video 5108 RTP/AVP 34 105
a=rtpmap:34 h263/90000
a=fmtp:34 QCIF=2
a=sendrecv
a=rtcp:5109
a=rtpmap:105 h263-2000/90000
a=fmtp:105 profile=0; level=10
"
         media="audiovideo"
         audio="sendrecv"
         video="sendrecv"
         uri="sip:sip@10.20.129.61"
         caller_uri=""Your Full
Name"<sip:Username@10.20.129.115:5060>;tag=21ED4D91C0C161B6761C647CD39FBB14"
         called_uri="<sip:sip@10.20.129.61>;tag=f717d348-3d81140a-13c4-50022-12d04-20c5bb20-
12d04"
         application_id="app" >
         <sip_headers raw_sip_headers="Call-
ID:A58594002950C3322BC918833221434A1267@10.20.129.115
CSeq: 1 INVITE
```

```
To: <sip:sip@10.20.129.61>
From: "Your Full Name"<sip:Username@10.20.129.115:5060>;tag=21ED4D91C0C161B6761C647CD39FBB14
Contact: <sip:Username@10.20.129.115:5060;transport=udp>
Content-Type:application/sdp
Content-Length:454
User-Agent: Kapanga Softphone Desktop Windows
1.00/2182d+1329238479_00FFB0ABE80A_A088B4778D9C_2C4138058FBB_CC52AFCC7D8F_080027006444 (not
registered)
Session-Expires: 1800;refresher=uac
">
         <param name="ack" value="GET_CALL_INFO"/>
         <param name="headers.CSeq" value="1 INVITE"/>
         <param name="headers.Call-ID"
value="A58594002950C3322BC918833221434A1267@10.20.129.115"/>
         <param name="headers.Contact" value="<sip:Username@10.20.129.115:5060;transport=udp>"/>
         <param name="headers.Content-Length" value="454"/>
         <param name="headers.Content-Type" value="application/sdp"/>
         <param name="headers.From" value=""Your Full
Name"<sip:Username@10.20.129.115:5060>;tag=21ED4D91C0C161B6761C647CD39FBB14"/>
         <param name="headers.Session-Expires" value="1800;refresher=uac"/>
         <param name="headers.To" value="<sip:sip@10.20.129.61>"/>
         <param name="headers.User-Agent" value="Kapanga Softphone Desktop Windows
1.00/2182d+1329238479_00FFB0ABE80A_A088B4778D9C_2C4138058FBB_CC52AFCC7D8F_080027006444 (not
registered)"/>
         <param name="status" value="0 No Error"/>
         <param name="transaction_id" value="9"/>
         <param name="type" value="ACK"/>
         </sip_headers>
</get_call_info>
</call_action>
</call_response>
</web_service>
```


add_ice_candidate

Add a *trickle-ICE* candidate to the remote SDP (3PCC).

Request Payload Example


```
<web_service version="1.0">
  <call>
    <call_action>
       <add_ice_candidate mline="[media section index]" mid="[media section Id]"
candidate="[candidate attribute]" />
    </call_action>
  </call>
</web_service>
```

Response Payload Attributes

Parameter	Default	Optional	Description
mline			The m-line index used to identify candidate's media section.
mid			The mid attribute used to identify candidate's media section.
candidate			The candidate attribute.

Response Payload Example


```
<web_service version="1.0">
<call_response identifier="8be5d67d-07c7-4d18-a6ce-d44634a9f048" appid="app"
               href="http://10.20.129.61:81/default/calls/8be5d67d-07c7-4d18-a6ce-d44634a9f048">
<call_action>
<add_ice_candidate mline="[media session index]" mid="[media session Id]" candidate="[candidate
attribute]" />
</call_action>
</call_response>
</web_service>
```


get_last_action

To get the last performed action.

Request Payload Example


```
<web_service version="1.0">
  <call>
    <call_action>
       <get_last_action/>
    </call_action>
  </call>
</web_service>
```


Response Payload Example


```
<web_service version="1.0">
<call_response identifier="76618368-ac8a-4288-8bad-8a3ce3d4b1b5" appid="app"
href="/default/calls/76618368-ac8a-4288-8bad-8a3ce3d4b1b5"
 connected="yes" signaling="yes" cpa="no" call_type="inbound"
 media="audiovideo"
 dtmf_mode="rfc2833"
 destination_uri="sip:sip@10.20.129.70"
 source_uri="sip:Username@10.20.129.79:5060"
 async_dtmf="yes" async_tone="yes" cleardigits="no" encryption="none" ice="no"
info_ack_mode="automatic" hangup_ack_mode="automatic" early_media="no" audio="sendrecv"
video="sendrecv">
 gusid="b9a5ef23398749bfa6809fae974378f2">
```

```
&lt;call_action&gt;
&lt;get_last_action&gt;
<play transaction_id="351d7173-f774-4c20-a279-8640359fc9f0"
         max_time="10s"
         fetch_timeout="300s"
         offset="0s"
         delay="1s"
         repeat="0"
         skip_interval="1s"
         terminate_digits="#">
         &lt;play_source location="file://verification/play_menu"/&gt;
         <dvr_setting forward_key="1"
                  backward_key="2"
                  pause_key="3"
                  resume_key="4"
                  restart_key="5"/>
&lt;/play&gt;&lt;/get_last_action&gt;
&lt;/call_action&gt;
&lt;/call_response&gt;
&lt;/web_service&gt;
```

get_last_event

To get the last event.

Request Payload Example

```
&lt;web_service version="1.0"&gt;
  &lt;call&gt;
    &lt;call_action&gt;
       &lt;get_last_event/&gt;
    &lt;/call_action&gt;
  &lt;/call&gt;
&lt;/web_service&gt;
```

Response Payload Example

```
&lt;web_service version="1.0"&gt;
&lt;event type="end_play" resource_id="76618368-ac8a-4288-8bad-8a3ce3d4b1b5" resource_type="call"&gt;
&lt;event_data name="duration" value="6180" /&gt;
&lt;event_data name="reason" value="end" /&gt;
&lt;event_data name="status" value="0 No Error" /&gt;
&lt;event_data name="transaction_id" value="351d7173-f774-4c20-a279-8640359fc9f0" /&gt;
&lt;/event&gt;
&lt;/web_service&gt;
```

cpa_operation

Start or stop call progress analysis (cpa); a cpa event might be generated.

Request Payload Attributes

Parameter	Default	Optional	Description
action	(none)		Start or stop call progress analysis. Values: • "start" • "stop"
cpa_profile	(system default)	*	Use configuration profile for call progress analysis. See the *Dialogic® PowerMedia™ XMS Installation and Configuration Guide* for more information.

Request Payload Example


```
<web_service version="1.0">
  <call>
    <call_action>
       <cpa_operation action="start"/>
    </call_action>
  </call>
</web_service>
```


Response Payload Example


```
<web_service version="1.0">
     <call_response identifier="8fe4c7f4-067c-455d-afee-52f57e00314b" appid="app"
           href="/default/calls/8fe4c7f4-067c-455d-afee-52f57e00314b"
           connected="yes" signaling="yes" cpa="no" call_type="inbound"
           media="audio"
           dtmf_mode="rfc2833"
           source_uri="sip:Username@10.20.129.113:5060"
           async_dtmf="no" async_tone="no" cleardigits="no" info_ack_mode="automatic"
early_media="yes"
           gusid="b9a5ef23398749bfa6809fae974378f2">
           <call_action>
             <cpa_operation action="start"/>
          </call_action>
     </call_response>
</web_service>
```


HTTP DELETE

Deletes a call resource.

```
DELETE /calls/[call_id]?appid=[app_id]
```

Conference Resource

The Conference Resource encapsulates a single instance of a conference resource on PowerMedia XMS. It contains all call resources currently included in the active conference.

Conference-related properties and actions associated with the conference are defined in this section. These include play, update_play, record, and stop.

Note: When using playrecord and record, playrecord executes the play list first. Once all of the specified plays are complete, the record functionality is executed.

For conference sub-resources, see the Conference Sub-Resource section.

The following tables show the HTTP methods that can be used with a conference.

Note: The payloads shown are examples only as there are many possible variations.

conferences

Resource URI

`/conferences?appid=[app_id]`

HTTP GET

Retrieves all available conference resources.

`GET /conferences?appid=[app_id]`

Response Payload Example


```
<web_service version="1.0">
   <conferences_response size="1">
              <conference_response appid="app" auto_gain_control="yes" beep="yes" caption="yes"
                                   caption_duration="30s" clamp_dtmf="yes"
echo_cancellation="yes"
                                   href="/default/conferences/830b9fda-d89e-495d-b7a1-
6a63402bcdcf"
                                   identifier="830b9fda-d89e-495d-b7a1-6a63402bcdcf" layout="2"
layout_size="automatic"
                                   max_parties="2" reserve="2" type="audiovideo">
              </conference_response>
  </conferences_response>
</web_service>
```

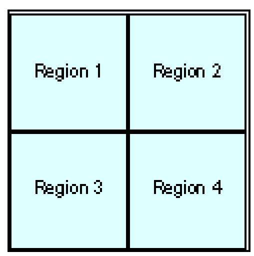
HTTP POST

Creates a conference resource.

`POST /conferences?appid=[app_id]`

Request Payload Attributes

Parameter	Default	Optional	Description
mixing_mode	"mcu"	*	Specifies the video conference mode. Values: • "mcu" • "sfu"
type	"audio"	*	Sets the media supported by conference. Values: • "audio" • "audiovideo"
max_parties	"9"	*	Maximum number of parties in a conference. Range: 2 to N.
reserve	"0"	*	Number of party resources to reserve for this conference. Any requests beyond this value are honored on a best-effort basis.
layout	"0"	*	The number of regions displayed in the conference video output using a standard layout. Valid values are "0", "1", "2","4", "6" and "9". Setting to "0" means that the number of regions displayed is determined by the number of visible parties. Intermediate values are rounded up to the next supported value e.g., 3 => 4, 7 => 9. Values greater than 9 will result in a 9 region layout. In addition to the original format, a new format using key/value pairs can be used.
layout_regions		*	Specifies a custom video layout. When defined, this takes precedence over the layout parameter. The layout is defined by a semicolon delimited list of region definitions. Each region is defined using this format: <ID> EQUALS <LEFT> COMMA <TOP> COMMA <RELATIVE_SIZE> COMMA <PRIORITY> [SEMICOLON if adding another region to the layout]. • ID uniquely identifies the region. The value "0" is reserved and may not be used as a region identifier.

Parameter	Default	Optional	Description
			• The LEFT parameter defines the position of the left region as an offset from the left of the root window. It is expressed as a percentage. • The TOP parameter defines the position of the top region as an offset from the top of the root window. It is expressed as a percentage. • RELATIVE_SIZE defines the size of a region relative to the size of the root window. It is expressed as either a fraction or decimal percentage. • The PRIORITY parameter is optional and is an integer values from 0 to 10, where 0 means disabled and 1 is the highest priority (default). Example of a 1-party conference: layout_regions="1=5,6,50,2" Example of a 4-party conference (with illustration): layout_regions="1=0,0,50,1;2=50,0,50,1;3=0,50,50,1;4=50,50,50,1"  The following parameters are supported: • **region** uniquely identifies the region. The value "0" is reserved and may not be used as a region identifier. • **left** is the position of the left of the region relative to the root window, expressed as either a fraction or as a decimal percentage. • **top** is the position of the top of the region relative to the root window, expressed as either a fraction or as a decimal percentage. • **width** is the width of the region relative to the size of the root window, expressed as either a fraction or as a decimal percentage. • **height** is the height of the region relative to the size of the root window, expressed as either a fraction or as a decimal percentage.

<table>
<tr><th>Parameter</th><th>Default</th><th>Optional</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>priority is optional. It is an integer value from 0 to 10, where 0 means disabled and 1 (default) is the highest priority.
background_color supports all the common HTML color names including the decimal "rgb(ddd,ddd,ddd)" and hexadecimal "#xxxxxx" RGB formats. Values are not case sensitive. The default is "black".
aspect_ratio_mode controls how a difference in aspect ratio between input video and region should be handled.
fill should not be used to compensate for aspect ratio differences. Fill the region with the input video. Image may appear distorted. (Default)
fit zooms out until image width or height fits the region. The image will appear centered in the region with letter or pillar boxes filled with the region's background color.
crop zooms in until image width or height fits the region. Image will appear centered in region with left/right or top/bottom cropped.
background_image is the URI of the image file to be used as the background image for the current region. JPEG and PNG are supported. The default is "(none)".</td></tr>
<tr><td>layout_size</td><td>"automatic"</td><td>*</td><td>Determines the size of the root window.
Values:
"automatic"
"qcif"
"cif"
"vga"
"720p"
The value "automatic" causes the window size to track the size of the largest party.</td></tr>
</table>

Parameter	Default	Optional	Description
orientation	"landscape"	*	Sets the root window orientation. Values: • "landscape" • "orientation" A root window with size "VGA" will be 640x480 pixels in landscape orientation and 480x640 pixels in portrait orientation.
aspect_ratio_mode	(none)	*	Sets the default aspect ratio mode for all the conference regions if not specified. Values: • "fill" • "fit" • "crop"
caption	"yes"	*	Determines if the caller's ID is overlaid on their image. Values: • "yes" • "no"
caption_duration	"20s"	*	The length of time that the caption is shown. Use "infinite" without the quotes to display the caption for the entire call.
beep	"yes"	*	Determines if a tone is played when a party joins/leaves a conference. Values: • "yes" • "no"
clamp_dtmf	"yes"	*	Determines if DTMF digits are suppressed. Values: • "yes" • "no"
auto_gain_control	"yes"	*	Determines if automatic gain control should be used. Values: • "yes" • "no"

Parameter	Default	Optional	Description
echo_cancellation	"yes"	*	Determines if echo cancellation should be used. Values: • "yes" • "no"
active_talker_region	(none)	*	Specifies the ID of the region used to display the active talker's video.
active_talker_interval	"0"	*	Specifies the minimum duration of time that must pass before changes to active talkers will be notified. The minimum value is 500ms. A value of "0" disables active talker notifications.
max_active_talkers	"10"	*	Sets the maximum number of active talkers to be included in the audio mix. Range: 2 to 10 (inclusive).
region_overlays	(none)	*	Specifies a set of text and/or image overlays for a conference layout. The overlays are defined by a semicolon delimited list of region definitions where each specifies a set of parameters that completely describes the overlay attributes for the conference layout regions or root window. Each region overlay is defined using this format: <REGION ID> EQUALS <ID> COMMA <PARAMETER> EQUALS <VALUE> COMMA <PARAMETER> EQUALS <VALUE>... • REGION ID uniquely identifies the conference region where the overlays should be applied. Note, "0" is reserved and used ONLY when applying a text or image overlay to the root window of a conference. • PARAMETER identifies the overlay parameter being set. • VALUE identifies the value of the overlay parameter given. Refer to Text and Image Overlays for details.
gusid		*	Specifies the Global Unique Session Identifier (gusid).

Request Payload Example

```
<web_service version="1.0">
<conference active_talker_interval="0ms" aspect_ratio_mode="fit" auto_gain_control="yes"
beep="yes" caption="no" caption_duration="infinite" clamp_dtmf="no" echo_cancellation="yes"
layout_regions="region=1,left=0,top=0,width=50%,height=100%,aspect_ratio_mode=fit;region=2,left=5
0%,top=0,width=50%,height=100%,aspect_ratio_mode=fit" layout_size="vga" max_parties="2"
mixing_mode="mcu" orientation="portrait" reserve="0" type="audiovideo"/>
</web_service>
```

Response Payload Example

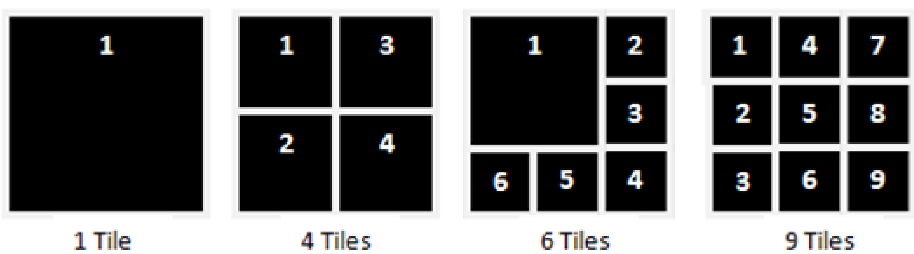

```
<web_service version="1.0">
<conference_response identifier="8b4148f3-e85b-4f7b-b0ac-9c07b59cddb0" appid="app"
href="/default/conferences/8b4148f3-e85b-4f7b-b0ac-9c07b59cddb0" type="audiovideo"
max_parties="2" reserve="0" layout="0" caption="no" caption_duration="infinite" beep="yes"
clamp_dtmf="no" auto_gain_control="yes" echo_cancellation="yes" layout_size="vga"
orientation="portrait" aspect_ratio_mode="fit"
layout_regions="region=1,left=0,top=0,width=50%,height=100%,aspect_ratio_mode=fit;region=2,left=5
0%,top=0,width=50%,height=100%,aspect_ratio_mode=fit" active_talker_interval="0ms"
mixing_mode="mcu">
</conference_response>
</web_service>
```


Conference Concepts

This section contains a higher-level look at various aspects of PowerMedia XMS conference behavior.

Screen Layout

The standard video conference tiling layout for 1, 4, 6 and 9 tile conferences and the order in which conferees appear in the tiles is shown below:



Conference Sub-Resource

For details on conference resources, see the Conference Resource section.

conference

Resource URI

`/conferences/[conference_id]?appid=[app_id]`

HTTP GET

Retrieves an available conference resource.

`GET /conferences/[conference_id]?appid=[app_id]`

Response Payload Example


```
<web_service version="1.0">
              <conference_response appid="app" auto_gain_control="yes" beep="yes" caption="yes"
caption_duration="30s"
                                   clamp_dtmf="yes" echo_cancellation="yes"
                                   href="/default/conferences/830b9fda-d89e-495d-b7a1-
6a63402bcdcf"
                                   identifier="830b9fda-d89e-495d-b7a1-6a63402bcdcf" layout="2"
layout_size="automatic"
                                   max_parties="2" reserve="2" type="audiovideo">
                                   <conf_participant audio="sendrecv" auto_gain_control="yes"
```

```
call_id="c87fcca1-b2d0-49c5-8b89-
baaae71cf695"
caption="Username" clamp_dtmf="yes"
echo_cancellation="yes"
mode="normal" mute="no" privilege="no"
region="0"
tx_mute="no" video="sendrecv"/>
        </conference_response>
</web_service>
```

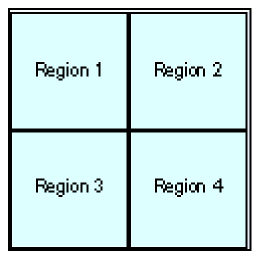

HTTP PUT

Updates a conference resource.

`PUT /conferences/[conference_id]?appid=[app_id]`

Modify Conference Attributes

Parameter	Default	Optional	Description
layout	(none)	*	The number of tiles displayed in the conference output. Valid values are "0", "1", "2", "4", "6" and "9". Setting to "0" means that the number of tiles displayed is determined by the number of active presenters.
layout_regions	(none)	*	Specifies a custom video layout. When defined, this takes precedence over the layout parameter. The layout is defined by a semicolon delimited list of region definitions. Each region is defined using this format: <ID> EQUALS <LEFT> COMMA <TOP> COMMA <RELATIVE_SIZE> COMMA <PRIORITY> [SEMICOLON if adding another region to the layout]. • ID uniquely identifies the region. The value "0" is reserved and may not be used as a region identifier. • The LEFT parameter defines the position of the left region as an offset from the left of the root window. It is expressed as a percentage. • The TOP parameter defines the position of the top region as an offset from the top of the root window. It is expressed as a percentage. • RELATIVE_SIZE defines the size of a region relative to the size of the root window. It is expressed as

Parameter	Default	Optional	Description
			either a fraction or decimal percentage. • The PRIORITY parameter is optional and is an integer values from 0 to 10, where 0 means disabled and 1 is the highest priority (default). Example of a 1-party conference: layout_regions="1=5,6,50,2" Example of a 4-party conference (with illustration): layout_regions="1=0,0,50,1;2=50,0,50,1;3=0,50,50,1;4=50,50,50,1" 
layout_size	(none)	*	Determines the size of the root window. Values: • "automatic" • "qcif" • "cif" • "vga" • "720p" The value "automatic" causes the window size to track the size of the largest party.
orientation	(none)	*	Sets the root window orientation. Values: • "landscape" • "orientation" A root window with size "VGA" will be 640x480 pixels in landscape orientation and 480x640 pixels in portrait orientation.
active_talker_region	(none)	*	Specifies the ID of the region used to display the active talker's video. Use "0" to disable.

Parameter	Default	Optional	Description
active_talker_interval	(none)	*	Specifies the minimum duration of time that must pass before changes to active talkers will be notified. The minimum value is 500ms. A value of "0" disables active talker notifications.
max_active_talkers	(none)	*	Sets the maximum number of active talkers to be included in the audio mix. Values range from 2 to 10 inclusive. Note, it is an error to set the value lower than the current number of privileged parties.
region_overlays	(none)	*	Specifies a set of text and/or image overlays for a conference layout. The overlays are defined by a semicolon delimited list of region definitions where each specifies a set of parameters that completely describes the overlay attributes for the conference layout regions or root window. Each region overlay is defined using this format: <REGION ID> EQUALS <ID> COMMA <PARAMETER> EQUALS <VALUE> COMMA <PARAMETER> EQUALS <VALUE>... • REGION ID uniquely identifies the conference region where the overlays should be applied. Note, "0" is reserved and used ONLY when applying a text or image overlay to the root window of a conference. • PARAMETER identifies the overlay parameter being set. • VALUE identifies the value of the overlay parameter given. Refer to Text and Image Overlays for details.
primary_video_source		*	SFU only. Specifies the primary video source sent to all parties. This may be either the string "vas", which automatically selects the video source based on voice activity or a specific conference party identified using its call_id. **Note:** For a list of SFU limitations and precautions, see the *Dialogic® PowerMedia® XMS Release 3.3 Release Notes*.

Request Payload Example


```
<web_service version="1.0">
<conference active_talker_interval="0ms" aspect_ratio_mode="fit" auto_gain_control="yes"
beep="yes" caption="no" caption_duration="infinite" clamp_dtmf="no" echo_cancellation="yes"
layout_regions="region=1,left=0,top=0,width=50%,height=100%,aspect_ratio_mode=fit;region=2,left=5
0%,top=0,width=50%,height=100%,aspect_ratio_mode=fit" layout_size="vga" max_parties="2"
mixing_mode="mcu" orientation="portrait" reserve="0" type="audiovideo"/>
</web_service>
```


Response Payload Example


```
<web_service version="1.0">
<conference_response identifier="8b4148f3-e85b-4f7b-b0ac-9c07b59cddb0" appid="app"
href="/default/conferences/8b4148f3-e85b-4f7b-b0ac-9c07b59cddb0" type="audiovideo"
max_parties="2" reserve="0" layout="0" caption="no" caption_duration="infinite" beep="yes"
clamp_dtmf="no" auto_gain_control="yes" echo_cancellation="yes" layout_size="vga"
orientation="portrait" aspect_ratio_mode="fit"
layout_regions="region=1,left=0,top=0,width=50%,height=100%,aspect_ratio_mode=fit;region=2,left=5
0%,top=0,width=50%,height=100%,aspect_ratio_mode=fit" active_talker_interval="0ms"
mixing_mode="mcu">
</conference_response>
</web_service>
```


Perform Conference Action

- play
- update_play
- record
- multi_record
- update_multi_record
- playcollect
- playrecord
- send_dtmf
- send_event
- stop
- get_last_action
- get_last_event

play

Request Payload Attributes

Parameter	Default	Optional	Description
play_source	(none)		Refer to play_source_attributes.
offset	"0"	*	Specifies the time offset from where the play should start. Note, the "offset" is applied to the initial play only.

Parameter	Default	Optional	Description
repeat	"0"	*	Number of times to repeat the play. Use "infinite" to repeat indefinitely. "file://" URIs only.
delay	"1s"	*	Time delay between repeated plays.
max_time	"infinite"	*	Limit the playback time to this value.
region	"0"	*	The ID of the region that will display the video media. The value "0" causes the video to be shown full-screen with the current layout being restored automatically when the play back completes.
no_cache		*	Cache-control for HTTP URIs. Values: • "yes" does not use caching. • "no" uses caching. The default is set by the configuration.
max_age		*	Cache-control for HTTP URIs, the maximum age, in seconds, of a cached file. The default is set by the configuration.
max_stale		*	Cache-control for HTTP URIs, the number of seconds that a cached file may exceed its expiration time by and still be a considered as fresh. The default is set by the configuration.

Parameter	Default	Optional	Description
fetch_timeout	"300s"	*	HTTP URIs, the maximum time in seconds to retrieve the file. This is the overall period of the transaction.

Request Payload Example


```
<web_service version="1.0">
  <conference>
    <conf_action>
        <play max_time="20s" >
              <play_source audio_uri="file://verification/play_menu.wav" audio_type="audio/x-wav"
/>
        </play>
    </conf_action>
</conference>
</web_service>
```


Response Payload Example


```
<web_service version="1.0">
        <conference_response identifier="830b9fda-d89e-495d-b7a1-6a63402bcdcf" appid="app"
                             href="/default/conferences/830b9fda-d89e-495d-b7a1-6a63402bcdcf"
                             type="audiovideo" max_parties="2" reserve="2"
                             layout="2" caption="yes" caption_duration="30s" beep="yes"
                             clamp_dtmf="yes" auto_gain_control="yes" echo_cancellation="yes"
layout_size="automatic">
                             <conf_participant call_id="c87fcca1-b2d0-49c5-8b89-baaae71cf695"
audio="sendrecv"
                                               video="sendrecv" caption="Username" region="0"
clamp_dtmf="yes"
                                               auto_gain_control="yes" echo_cancellation="yes"
mute="no" tx_mute="no"
                                               privilege="no" mode="normal"/>
                             <conf_action>
                                 <play transaction_id="acb08f84-afb4-430b-92f2-22083b7638aa"
                                       max_time="20s"
                                       fetch_timeout="300s"
                                       offset="0s"
                                       delay="1s"
                                       repeat="0"
                                       region="0">
                                       <play_source audio_uri="file://verification/play_menu.wav"
                                                    audio_type="audio/x-wav"/>
                                 </play>
                             </conf_action>
         </conference_response>
```

```
</web_service>
```

update_play

Request Payload Attributes

Parameter	Default	Optional	Description
transaction_id	(none)		Media identifier, returned by play.
dvr_action	(none)		Values: • "backward" - skip backwards. • "forward" - skip forward. • "pause" - pause playback. • "restart" - jump back to the start. • "resume" - resume paused playback.
region	(none)	*	Conference only. The ID of the region that will display the video media.

Request Payload Example


```
<web_service version="1.0">
  <conference>
    <conf_action>
        <update_play transaction_id="acb08f84-afb4-430b-92f2-22083b7638aa" dvr_action="pause"/>
    </conf_action>
  </conference>
</web_service>
```


Response Payload Example


```
<web_service version="1.0">
        <conference_response identifier="830b9fda-d89e-495d-b7a1-6a63402bcdcf" appid="app"
                             href="/default/conferences/830b9fda-d89e-495d-b7a1-6a63402bcdcf"
                             type="audiovideo" max_parties="2" reserve="2"
                             layout="2" caption="yes" caption_duration="30s" beep="yes"
                             clamp_dtmf="yes" auto_gain_control="yes" echo_cancellation="yes"
layout_size="automatic">
                             <conf_participant call_id="c87fcca1-b2d0-49c5-8b89-baaae71cf695"
audio="sendrecv"
                                               video="sendrecv" caption="Username" region="0"
clamp_dtmf="yes"
```

```
                                              auto_gain_control="yes" echo_cancellation="yes"
mute="no" tx_mute="no"
                                              privilege="no" mode="normal"/>
                              <conf_action>
                                  <update_play transaction_id="acb08f84-afb4-430b-92f2-
22083b7638aa"
                                             dvr_action="pause"/>
                              </conf_action>
         </conference_response>
</web_service>
```

record

Request Payload Attributes

Parameter	Default	Optional	Description
recording_audio_mime_params	(none)		Refer to recording_audio_mime_params.
recording_video_mime_params	(none)		Refer to recording_video_mime_params.
terminate_digits	"#"	*	The digit or digits [0-9,*,#] used to terminate the play_record.
recording_uri	(none)		A filename "file://..." that must refer to an existing directory or "http://" ("https://"). **Note:** This is deprecated.
recording_audio_uri	(none)		The URL of the audio media to be recorded e.g., "file://...", "http(s)://...".

Parameter	Default	Optional	Description
recording_audio_type	(none)		The mime type of the audio media. Values: • "audio/x-wav" • "audio/basic" • "audio/x-alaw-basic" • "audio/L8" • "audio/L16" • "audio/x-aud" • "audio/AMR" • "audio/AMR-WB" • "audio/3gpp" • "audio/mp4" • "audio/mkv" • "audio/webm" • "text/uri-list"
recording_video_uri	(none)		The URL of the video media to be recorded e.g., "file://...", "http(s)://...".
recording_video_type	video/x-vid		The mime type of the video media. Values: • "video/x-vid" • "video/3gpp" • "video/mp4" • "video/mkv" • "video/webm"
max_time	"infinite"	*	The maximum length of time to record. Use "infinite" for no limit.
public_key	(none)	*	RSA public key in pem/base64 format in each encrypted request.

Request Payload Example

```
<web_service version="1.0">
  <conference>
      <conf_action>
```

```
              <record recording_audio_uri="file://recorded_file.wav"
recording_audio_type="audio/x-wav"
                            max_time="10s" terminate_digits="#" />
      </conf_action>
  </conference>
</web_service>
```


Response Payload Example


```
<web_service version="1.0">
        <conference_response identifier="830b9fda-d89e-495d-b7a1-6a63402bcdcf" appid="app"
                             href="/default/conferences/830b9fda-d89e-495d-b7a1-6a63402bcdcf"
                             type="audiovideo" max_parties="2" reserve="2"
                             layout="2" caption="yes" caption_duration="30s" beep="yes"
                             clamp_dtmf="yes" auto_gain_control="yes" echo_cancellation="yes"
layout_size="automatic">
                             <conf_participant call_id="c87fcca1-b2d0-49c5-8b89-baaae71cf695"
audio="sendrecv"
                                               video="sendrecv" caption="Username" region="0"
clamp_dtmf="yes"
                                               auto_gain_control="yes" echo_cancellation="yes"
mute="no" tx_mute="no"
                                               privilege="no" mode="normal"/>
                             <conf_action>
                                 <record transaction_id="a560cbc1-5674-44e8-bb21-2b34130169c4"
                                         terminate_digits="#"
                                         max_time="10s"
                                    recording_audio_uri="file://recorded_file.wav"
                                         recording_audio_type="audio/x-wav"/>
                             </conf_action>
         </conference_response>
</web_service>
```


Public Key Example


```
<web_service version="1.0">
 <conference>
  <conf_action>
<record max_time="10s" public_key="-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxq1dDrwpaDhnkYCe7xZS
7qHYcXDo6bNNz54/gipWtL1nH2ArGZvAPMuLS5ADkbQdOSrHEd8tu1ziFpBt6Va8
mzLMFQ7o/q8VmBHFDXCLCWHDFZEVGfBTmryrrtMK1Cxj4r19SRZ3lneWd0HKIfyY
OWADzGDUsnEQXtX+o98xLQHh4sichnTvCLegCZrogZsVsi3uvUlUh1v1L0toje/X
uWuN6ZjloHmWIUS9MLmtQejhrJM9xrKnMbQF0MvMor5Iff3L0OcGmdYi2CFgksV0
ma5FRMYqoGL/LeszhZqwN7E6Zkh56AG5mMNuqB2XwdNv9lBKdROlZFFueHri7HT8
hwIDAQAB
-----END PUBLIC KEY-----"
recording_audio_type="audio/opus"
recording_audio_uri="file:///testing/mxml/recordings/mt_caller_1.webm" terminate_digits="#">
      </record>
  </conf_action>
```

```
 </conference>
</web_service>
```

multi_record

Record media from multiple sources to a multitrack container. The recording is terminated when the last source is destroyed or after an optional timeout. The recording may be cancelled using the stop API. Both tracks need to be recorded initially.

Request Payload Attributes

Parameter	Default	Optional	Description
recording_audio_mime_params	(none)		Refer to recording_audio_mime_params.
recording_video_mime_params	(none)		Refer to recording_video_mime_params.
record_track	(none)		Refer to record_track.
terminate_digits	"#"	*	The digit or digits [0-9,*,#] used to terminate the play_record.
recording_uri	(none)		A filename "file://..." which must refer to an existing directory or "http://" ("https://"). See also media. **Note:** This is deprecated.
recording_audio_uri	(none)		The URL of the audio media to be recorded e.g. "file://...", "http(s)://...".

Parameter	Default	Optional	Description
recording_audio_type	(none)		The mime-type of the audio media. Values: • "audio/x-wav" • "audio/basic" • "audio/x-alaw-basic" • "audio/L8" • "audio/L16" • "audio/x-aud" • "audio/AMR" • "audio/AMR-WB" • "audio/3gpp" • "audio/mp4" • "audio/mkv" • "audio/webm" • "text/uri-list"
recording_video_uri	(none)		The URL of the video media to be recorded e.g. "file://...", "http(s)://...".
recording_video_type	video/x-vid		The mime-type of the video media. Values: • "video/x-vid" • "video/3gpp" • "video/mp4" • "video/mkv" • "video/webm"
max_time	"infinite"	*	The maximum length of time to record. Use "infinite" for no limit.
public_key	(none)	*	RSA public key in pem/base64 format in each encrypted request.

Request Payload Example

```
<web_service version="1.0">
  <conference>
      <conf_action>
              <multi_record terminate_digits="#" max_time="10s"
                          recording_audio_uri="file://verification/multi_recorded_file.wav"
```

```
                              recording_audio_type="audio/x-wav">
                              <recording_audio_mime_params codec="mulaw" rate="8000"/>
                              <record_track id="830b9fda-d89e-495d-b7a1-6a63402bcdcf"/>
                              <record_track id="87b13b33-6558-4191-a1e0-db0052efaa63"/>
                </multi_record>
       </conf_action>
   </conference>
</web_service>
```

Response Payload Example

```
<web_service version="1.0">
        <conference_response identifier="830b9fda-d89e-495d-b7a1-6a63402bcdcf" appid="app"
                             href="/default/conferences/830b9fda-d89e-495d-b7a1-6a63402bcdcf"
                               type="audiovideo" max_parties="2" reserve="2"
                               layout="2" caption="yes" caption_duration="30s" beep="yes"
                               clamp_dtmf="yes" auto_gain_control="yes" echo_cancellation="yes"
layout_size="auto">
                               <conf_participant call_id="c87fcca1-b2d0-49c5-8b89-baaae71cf695"
audio="sendrecv"
                                                video="sendrecv" caption="Username" region="0"
         clamp_dtmf="yes"
                                                auto_gain_control="yes" echo_cancellation="yes"
         mute="no" tx_mute="no"
                                                privilege="no" mode="normal"/>
                             <conf_action>
                                 <multi_record transaction_id="b3407431-801f-4a44-ba8b-
fa8c9fddd653"
                                                terminate_digits="#" max_time="10s"

recording_audio_uri="file://verification/multi_recorded_file.wav"
                                                recording_audio_type="audio/x-wav">
                                                <recording_audio_mime_params codec="mulaw"
rate="8000"/>
                                                <record_track id="830b9fda-d89e-495d-b7a1-
6a63402bcdcf" track_id="0"/>
                                                <record_track id="87b13b33-6558-4191-a1e0-
db0052efaa63" track_id="1"/>
                                 </multi_record>
                             </conf_action>
         </conference_response>
</web_service>
```

Public Key Example

```
<web_service version="1.0">
 <conference>
  <conf_action>
<multi_record max_time="10s" public_key="-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxq1dDrwpaDhnkYCe7xZS
7qHYcXDo6bNNz54/gipWtL1nH2ArGZvAPMuLS5ADkbQdOSrHEd8tu1ziFpBt6Va8
mzLMFQ7o/q8VmBHFDXCLCWHDFZEVGfBTmryrrtMK1Cxj4r19SRZ3lneWd0HKIfyY
```

```
OWADzGDUsnEQXtX+o98xLQHh4sichnTvCLegCZrogZsVsi3uvUlUh1v1L0toje/X
uWuN6ZjloHmWIUS9MLmtQejhrJM9xrKnMbQF0MvMor5Iff3L0OcGmdYi2CFgksV0
ma5FRMYqoGL/LeszhZqwN7E6Zkh56AG5mMNuqB2XwdNv9lBKdROlZFFueHri7HT8
hwIDAQAB
-----END PUBLIC KEY-----"
recording_audio_type="audio/opus"
recording_audio_uri="file:///testing/mxml/recordings/mt_caller_1.webm" terminate_digits="#">
<record_track id="830b9fda-d89e-495d-b7a1-6a63402bcdcf" track_id="0"/>
<record_track id="87b13b33-6558-4191-a1e0-db0052efaa63" track_id="1"/>
      </multi_record>
  </conf_action>
 </conference>
</web_service>
```


update_multi_record

Update an active multitrack recording. It is used to remove a source from a track or add a source to a track. If adding a source, the current source needs to be stopped using id="none" and then calling update_multi_record again with the id of the new source.

Request Payload Attributes

Parameter	Default	Optional	Description
transaction_id	(none)		Media identifier, returned by multi_record.
id	(none)		Call or conference id. Source is removed from the recording. The recording continues with silence until another termination condition is reached or the record is stopped with stop action.
track_id	(none)		Record track identifier, returned by multi_record.
direction	recv	*	For calls only, this specifies the direction of the stream to be recorded. For conference, only "recv" direction is supported which is the output of the conference. Values: • "send" • "recv"

Request Payload Example


```
<web_service version="1.0">
  <conference>
```

```
     <conf_action>
        <update_multi_record transaction_id="b3407431-801f-4a44-ba8b-fa8c9fddd653"
direction="send" track_id="1"/>
     </conf_action>
  </conference>
</web_service>
```

Response Payload Example

```
<web_service version="1.0">
     <conference_response identifier="830b9fda-d89e-495d-b7a1-6a63402bcdcf" appid="app"
                             href="/default/conferences/830b9fda-d89e-495d-b7a1-6a63402bcdcf"
                              type="audiovideo" max_parties="2" reserve="2"
                              layout="2" caption="yes" caption_duration="30s" beep="yes"
                              clamp_dtmf="yes" auto_gain_control="yes" echo_cancellation="yes"
layout_size="auto">
                              <conf_participant call_id="c87fcca1-b2d0-49c5-8b89-baaae71cf695"
audio="sendrecv"
                                              video="sendrecv" caption="Username" region="0"
         clamp_dtmf="yes"
                                              auto_gain_control="yes" echo_cancellation="yes"
         mute="no" tx_mute="no"
                                              privilege="no" mode="normal"/>
          <conf_action>
              <update_multi_record transaction_id="b3407431-801f-4a44-ba8b-fa8c9fddd653"
direction="send" track_id="1"/>
          </conf_action>
     </conference_response>
</web_service>
```

playcollect

Request Payload Attributes

Parameter	Default	Optional	Description
play_source	(none)		Refer to play_source_attributes.
offset	"0"	*	Specifies the time offset from where the play should start. Note, the "offset" is applied to the initial play only.
repeat	"0"	*	Number of times to repeat the play. Use "infinite" to repeat indefinitely. "file://" URIs only.
delay	"1s"	*	Time delay between repeated plays.

Parameter	Default	Optional	Description
term_digits	"#"	*	The digit or digits [0-9,*,#] used to terminate the play_collect.
max_digits	Unlimited	*	The maximum number of digits to collect.
timeout	"infinite"	*	The maximum length of time to wait for the first digit or a tone. This time begins when the prompt phase ends.
interdigit_timeout	The value specified by the *timeout* parameter	*	The maximum length of time to wait for subsequent digits. This timeout is reset after each digit is received.
tone_detection	"no"	*	Enable tone detection. Values: • "yes" • "no"
barge	"yes"	*	Specifies whether dtmf digit input will barge the prompt and force transition to the collect phase. Note that if the "barge" attribute is set to "no", the "cleardigits" attribute implicitly has the value "yes". Values: • "yes" • "no"

Parameter	Default	Optional	Description
cleardigits	"no"	*	Specifies whether previous input should be considered or ignored for the purpose of barge-in and digit matching. When it is set to "yes", any previously buffered digits are discarded. If it is set to "no", previously buffered digits will be considered. If "cleardigits" is set to "no" and "barge" is set to "yes", previously buffered digits will result in the collection phase starting immediately, and the prompt will not be played. Values: • "yes" • "no"
no_cache		*	Cache-control for HTTP URIs. Values: • "yes" do not use caching. • "no" use caching. The default is set by the configuration.
max_age		*	Cache-control for HTTP URIs, the maximum age, in seconds, of a cached file. The default is set by the configuration.

Parameter	Default	Optional	Description
max_stale		*	Cache-control for HTTP URIs, the number of seconds that a cached file may exceed its expiration time by and still be a considered as fresh. The default is set by the configuration.
fetch_timeout	"300s"	*	HTTP URIs, the maximum time in seconds to retrieve the file. This is the overall period of the transaction.

Request Payload Example


```
<web_service version="1.0">
  <conference>
     <conf_action>
        <playcollect max_digits="4" timeout="10s" offset="0s" repeat="0" delay="1s"
terminate_digits="#"
                   tone_detection="yes" barge="yes" cleardigits="yes">
                   <play_source audio_uri="file://verification/play_menu.wav"
audio_type="audio/x-wav"/>
        </playcollect>
     </conf_action>
  </conference>
</web_service>
```


Response Payload Example


```
<web_service version="1.0">
           <conference_response identifier="830b9fda-d89e-495d-b7a1-6a63402bcdcf" appid="app"
                             href="/default/conferences/830b9fda-d89e-495d-b7a1-6a63402bcdcf"
                              type="audiovideo" max_parties="2" reserve="2"
                              layout="2" caption="yes" caption_duration="30s" beep="yes"
                              clamp_dtmf="yes" auto_gain_control="yes" echo_cancellation="yes"
layout_size="auto">
                              <conf_participant call_id="c87fcca1-b2d0-49c5-8b89-baaae71cf695"
audio="sendrecv"
                                              video="sendrecv" caption="Username" region="0"
         clamp_dtmf="yes"
                                              auto_gain_control="yes" echo_cancellation="yes"
         mute="no" tx_mute="no"
                                              privilege="no" mode="normal"/>
                           <conf_action>
                                <playcollect transaction_id="25608a36-2fd1-43ca-b741-
3420dbc389cb"
                                              fetch_timeout="300s"
```

```
                                                    terminate_digits="#"
                                                    timeout="10s"
                                                    interdigit_timeout=""
                                                    tone_detection="yes"
                                                    max_digits="4"
                                                    barge="yes"
                                                    offset="0s"
                                                    delay="1s"
                                                    repeat="0"
                                                    cleardigits="yes">
                                                    <play_source
audio_uri="file://verification/play_menu.wav"
                                                                  audio_type="audio/x-wav"/>
                                        </playcollect>
                                   </conf_action>
          </conference_response>
</web_service>
```

playrecord

Request Payload Attributes

Parameter	Default	Optional	Description
play_source	(none)		Refer to play_source_attributes.
recording_audio_mime_params	(none)		Refer to recording_audio_mime_params.
recording_video_mime_params	(none)		Refer to recording_video_mime_params.
offset	"0"	*	Specifies the time offset from where the play should start. Note that the offset is applied to the initial play only.
repeat	"0"	*	Number of times to repeat the play. "file://" URIs only.
delay	"1s"	*	Time delay between repeated plays.
terminate_digits	"#"	*	The digit or digits [0-9,*,#] used to terminate the play_record.
recording_uri	(none)		Filename "file://..." This must refer to an existing directory. **Note:** This is deprecated.

Parameter	Default	Optional	Description
recording_audio_uri	(none)		The URL of the audio media to be recorded e.g. "file://...", "http(s)://...".
recording_audio_type	(none)		The mime type of the audio media. Values: • "audio/x-wav" • "audio/basic" • "audio/x-alaw-basic" • "audio/L8" • "audio/L16" • "audio/x-aud" • "audio/AMR" • "audio/AMR-WB" • "audio/3gpp" • "audio/mp4" • "audio/mkv" • "audio/webm" • "text/uri-list"
recording_video_uri	(none)		The URL of the video media to be recorded e.g. "file://...", "http(s)://...".
recording_video_type	(none)		The mime-type of the video media. Values: • "video/x-vid" • "video/3gpp" • "video/mp4" • "video/mkv" • "video/webm"
beep	"yes"	*	Play a tone before starting to record. Values: • "yes" • "no"

<table>
<tr><th>Parameter</th><th>Default</th><th>Optional</th><th>Description</th></tr>
<tr><td>max_time</td><td>"infinite"</td><td>*</td><td>The maximum length of time to record. Use "infinite" for no limit.</td></tr>
<tr><td>max_silence</td><td>"infinite"</td><td>*</td><td>The maximum length of silence to record after audio has been detected. Use "infinite" for no limit. This feature is supported for calls only.</td></tr>
<tr><td>noinput_timeout</td><td>"infinite"</td><td>*</td><td>The maximum time to wait for audio to be detected. Use "infinite" for no limit. This feature is supported for calls only.</td></tr>
<tr><td>barge</td><td>"yes"</td><td>*</td><td>Specifies whether dtmf digit input will barge the prompt and force transition to the record phase. Note that if the "barge" attribute is set to "no", the "cleardigits" attribute implicitly has the value "yes".
Values:
"yes"
"no"</td></tr>
<tr><td>cleardigits</td><td>"no"</td><td>*</td><td>Specifies whether previous input should be considered or ignored for the purpose of barge-in. When it is set to "yes", any previously buffered digits are discarded. If it is set to "no", previously buffered digits will be considered. If "cleardigits" is set to "no" and "barge" is set to "yes", previously buffered digits will result in the recording phase starting immediately, and the prompt will not be played.
Values:
"yes"
"no"</td></tr>
</table>

Parameter	Default	Optional	Description
no_cache		*	Cache-control for HTTP URIs. Values: • "yes" do not use caching. • "no" use caching. The default is set by the configuration.
max_age		*	Cache-control for HTTP URIs, the maximum age, in seconds, of a cached file. The default is set by the configuration.
max_stale		*	Cache-control for HTTP URIs, the number of seconds that a cached file may exceed its expiration time by and still be a considered as fresh. The default is set by the configuration.
fetch_timeout	"300s"	*	HTTP URIs. The maximum time in seconds to retrieve or upload a file. This is the overall period of the transaction.
public_key	(none)	*	RSA public key in pem/base64 format in each encrypted request.

Request Payload Example

```
<web_service version="1.0">
  <conference>
    <conference_action>
      <playrecord recording_audio_uri="file://recorded_file.wav" recording_audio_type="audio/x-wav"
                  max_time="10s" offset="0s" repeat="0" delay="1s" terminate_digits="#"
                  beep="yes" barge="yes" cleardigits="yes" >
            <play_source audio_uri="file://verification/play_menu.wav" audio_type="audio/x-wav"/>
       </playrecord>
    </conference_action>
 </conference>
</web_service>
```

Response Payload Example

```
<web_service version="1.0">
      <conference_response identifier="830b9fda-d89e-495d-b7a1-6a63402bcdcf" appid="app"
```

```
                             href="/default/conferences/830b9fda-d89e-495d-b7a1-6a63402bcdcf"
                              type="audiovideo" max_parties="2" reserve="2"
                              layout="2" caption="yes" caption_duration="30s" beep="yes"
                              clamp_dtmf="yes" auto_gain_control="yes" echo_cancellation="yes"
layout_size="auto">
                              <conf_participant call_id="c87fcca1-b2d0-49c5-8b89-baaae71cf695"
audio="sendrecv"
                                              video="sendrecv" caption="Username" region="0"
         clamp_dtmf="yes"
                                              auto_gain_control="yes" echo_cancellation="yes"
         mute="no" tx_mute="no"
                                              privilege="no" mode="normal"/>
                     <conference_action>
                         <playrecord transaction_id="a88aac4a-dd73-4a3f-8268-792b90a5efb2"
                                      fetch_timeout="300s"
                                      terminate_digits="#"
                                      max_time="10s"
                                      beep="yes"
                                      barge="yes"
                                      cleardigits="yes"
                                      max_silence="infinite"
                                      noinput_timeout="infinite"
                                      recording_audio_uri="file://recorded_file.wav"
                                      recording_audio_type="audio/x-wav"
                                      offset="0s"
                                      delay="1s"
                                      repeat="0"
                                      terminate_digits="">
                                      <play_source audio_uri="file://verification/play_menu.wav"
                                                     audio_type="audio/x-wav"/>
                         </playrecord>
                     </conf_action>
      </conference_response>
</web_service>
```

Public Key Example

```
<web_service version="1.0">
 <conference>
  <conf_action>
<playrecord max_time="10s" public_key="-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxq1dDrwpaDhnkYCe7xZS
7qHYcXDo6bNNz54/gipWtL1nH2ArGZvAPMuLS5ADkbQdOSrHEd8tu1ziFpBt6Va8
mzLMFQ7o/q8VmBHFDXCLCWHDFZEVGfBTmryrrtMK1Cxj4r19SRZ3lneWd0HKIfyY
OWADzGDUsnEQXtX+o98xLQHh4sichnTvCLegCZrogZsVsi3uvUlUh1v1L0toje/X
uWuN6ZjloHmWIUS9MLmtQejhrJM9xrKnMbQF0MvMor5Iff3L0OcGmdYi2CFgksV0
ma5FRMYqoGL/LeszhZqwN7E6Zkh56AG5mMNuqB2XwdNv9lBKdROlZFFueHri7HT8
hwIDAQAB
-----END PUBLIC KEY-----"
```

```
recording_audio_type="audio/opus"
recording_audio_uri="file:///testing/mxml/recordings/mt_caller_1.webm" terminate_digits="#">
      </playrecord>
  </conf_action>
 </conference>
</web_service>
```

send_dtmf

Request Payload Attributes

Parameter	Default	Optional	Description
digits			Digit(s) to send [1234567890*#ABCD].
duration	100ms	*	Length of time of each digit.
interval	100ms	*	Time between successive digits.
level	-10dB	*	Amplitude of DTMF digit tones. Range: 0 to -40dB.

Request Payload Example

```
<web_service version="1.0">
  <conference>
   <conf_action>
      <send_dtmf digits="2345" />
   </conf_action>
  </conference>
</web_service>
```

Respone Payload Example

```
<web_service version="1.0">
    <conference_response identifier="830b9fda-d89e-495d-b7a1-6a63402bcdcf" appid="app"
                            href="/default/conferences/830b9fda-d89e-495d-b7a1-6a63402bcdcf"
                              type="audiovideo" max_parties="2" reserve="2"
                              layout="2" caption="yes" caption_duration="30s" beep="yes"
                              clamp_dtmf="yes" auto_gain_control="yes" echo_cancellation="yes"
layout_size="auto">
                              <conf_participant call_id="c87fcca1-b2d0-49c5-8b89-baaae71cf695"
audio="sendrecv"
                                               video="sendrecv" caption="Username" region="0"
         clamp_dtmf="yes"
                                               auto_gain_control="yes" echo_cancellation="yes"
         mute="no" tx_mute="no"
                                               privilege="no" mode="normal"/>
          <conf_action>
              <send_dtmf transaction_id="c4023bb2-4062-415b-ae79-2e708458cfdd"
                          digits="2345" duration="100ms" interval="100ms" level="-10dB"/>
          </conf_action>
```

```
    </conference_response>
</web_service>
```

send_event

Send one or more telephony events in RTP packets.

Request Payload Attributes

Parameter	Default	Optional	Description
digits			Event(s) to send [0123456789]. For values higher than 9, use "$xxx" where "xxx" is the three digit decimal code. For example, to send digits 0, 1, and 4 and event 16, the digits attribute would be set to digits="014$016".
duration	100ms	*	Length of time of each digit.
interval	100ms	*	Time between successive digits.
level	-10dB	*	Amplitude of the dtmf digit tones. Range: 0 to -40dB.

Request Payload Example

```
<web_service version="1.0">
  <conference>
   <conf_action>
      <send_event digits="2345$016" />
   </conf_action>
</conference>
</web_service>
```

Response Payload Example

```
<web_service version="1.0">
    <conference_response identifier="830b9fda-d89e-495d-b7a1-6a63402bcdcf" appid="app"
                             href="/default/conferences/830b9fda-d89e-495d-b7a1-6a63402bcdcf"
                              type="audiovideo" max_parties="2" reserve="2"
                              layout="2" caption="yes" caption_duration="30s" beep="yes"
                              clamp_dtmf="yes" auto_gain_control="yes" echo_cancellation="yes"
layout_size="auto">
                              <conf_participant call_id="c87fcca1-b2d0-49c5-8b89-baaae71cf695"
audio="sendrecv"
                                              video="sendrecv" caption="Username" region="0"
         clamp_dtmf="yes"
                                              auto_gain_control="yes" echo_cancellation="yes"
         mute="no" tx_mute="no"
                                              privilege="no" mode="normal"/>
```

```
        <conf_action>
            <send_event transaction_id="c4023bb2-4062-415b-ae79-2e708458cfdd"
                        digits="2345$016" duration="100ms" interval="100ms" level="-10dB"/>
        </conf_action>
    </conference_response>
</web_service>
```

stop

Request Payload Attributes

Parameter	Default	Optional	Description
transaction_id	(none)		Identifier returned by play, playcollect, playrecord, send_event, send_dtmf, record, or multi_record conference action.

Request Payload Example

```
<web_service version="1.0">
  <conference>
    <conf_action>
       <stop transaction_id="c4023bb2-4062-415b-ae79-2e708458cfdd"/>
    </conf_action>
  </conference>
</web_service>
```

Response Payload Example

```
<web_service version="1.0">
        <conference_response identifier="830b9fda-d89e-495d-b7a1-6a63402bcdcf" appid="app"
                             href="/default/conferences/830b9fda-d89e-495d-b7a1-6a63402bcdcf"
                             type="audiovideo" max_parties="2" reserve="2"
                             layout="2" caption="yes" caption_duration="30s" beep="yes"
                             clamp_dtmf="yes" auto_gain_control="yes" echo_cancellation="yes"
layout_size="automatic">
                             <conf_participant call_id="c87fcca1-b2d0-49c5-8b89-baaae71cf695"
audio="sendrecv"
                                               video="sendrecv" caption="Username" region="0"
clamp_dtmf="yes"
                                               auto_gain_control="yes" echo_cancellation="yes"
mute="no" tx_mute="no"
                                               privilege="no" mode="normal"/>
                             <conf_action>
                                  <stop transaction_id="c4023bb2-4062-415b-ae79-2e708458cfdd"/>

                             </conf_action>
         </conference_response>
</web_service>
```

get_last_action

To get the last performed action.

Request Payload Example

```
<web_service version="1.0">
  <conference>
    <conf_action>
       <get_last_action/>
    </conf_action>
  </conference>
</web_service>
```

Response Payload Example

```
<web_service version="1.0">
        <conference_response identifier="830b9fda-d89e-495d-b7a1-6a63402bcdcf" appid="app"
                             href="/default/conferences/830b9fda-d89e-495d-b7a1-6a63402bcdcf"
                              type="audiovideo" max_parties="2" reserve="2"
                              layout="2" caption="yes" caption_duration="30s" beep="yes"
                              clamp_dtmf="yes" auto_gain_control="yes" echo_cancellation="yes"
layout_size="automatic">
                              <conf_participant call_id="c87fcca1-b2d0-49c5-8b89-baaae71cf695"
audio="sendrecv"
                                              video="sendrecv" caption="Username" region="0"
         clamp_dtmf="yes"
                                              auto_gain_control="yes" echo_cancellation="yes"
         mute="no" tx_mute="no"
                                              privilege="no" mode="normal"/>
                             <conf_action>
                                 <get_last_action>
                                        <stop transaction_id="c4023bb2-4062-415b-ae79-
2e708458cfdd"/>
                                 </get_last_action>
                             </conf_action>
         </conference_response>
</web_service>
```

get_last_event

To get the last event.

Request Payload Example

```
<web_service version="1.0">
  <conference>
    <conf_action>
       <get_last_event/>
    </conf_action>
  </conference>
</web_service>
```

Response Payload Example


```
<web_service version="1.0">
     <event type="end_play" resourceid="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5"
resource_type="conference">
            <event_data name="reason" value="complete" />
            <event_data name="duration" value="30000ms" />
            <event_data name="transaction_id" value="0974c8b5-8a3c-4a8e-ae82-8f7c8bd0efdd" />
     </event>
</web_service>
```


HTTP DELETE

Deletes a conference resource.

```
DELETE /conferences/[conference_id]?appid=[app_id]
```

Event Handler Resource

HTTP event streaming is implemented in the PowerMedia XMS RESTful server as an eventhandler resource. When the client wishes to receive asynchronous events, it uses the web service to create an eventhandler and to subscribe to specific event types.

For example, when the client performs an HTTP **GET** on a newly created eventhandler, the PowerMedia XMS RESTful server responds with a `200 OK`; however, the TCP connection remains open until the client destroys the eventhandler. Event data related to resources and subscribed event types are sent to the client until it deletes the eventhandler. Event data related to resources and subscribed event types are sent to the client until it deletes the eventhandler.

An enabled PowerMedia XMS application ID must be included in the URL for the original HTTP POST where the eventhandler is created. This is utilized to assure that clients only have access to resources they create.

For eventhandler sub-resources, see the Event Handler Sub-Resource section.

The following tables show the HTTP methods that can be used with the eventhandler.

Note: The payloads shown are examples only as there are many possible variations.

RESTful Event Streaming Data Format Change

As of PowerMedia XMS Release 2.3, the RESTful event format has been updated to be compliant with HTTP chunked data formatting (RFC 7230, Section 4.1). The extra carriage return / line feed (CRLF) in previous PowerMedia XMS versions has been removed from the beginning of each chunk. Each chunk returned begins with the size of the XML payload in hex format.

Example:

```
44
<web_service version="1.0">
<event type="keepalive"/></web_service>
```

Note: Existing RESTful applications that make use of event handlers will require updating.

eventhandlers

Resource URI

```
/eventhandlers?appid=[app_id]
```

eventsubscribe_attributes

Parameter	Default	Optional	Description
action	add	*	This will add/remove an event subscription. Values: • "add" • "remove"
type	any	*	Type of events to monitor. Values: • "end_play" • "end_record" • "end_playcollect" • "end_playrecord" • "end_overlay" • "end_dtmf" • "keepalive" • "incoming" • "ringing" • "connected" • "hangup" • "info" • "dtmf" • "tone" • "any" • "end_speak" • "start_of_input" • "end_recognize" • "answered" • "active_talker" • "alarm" • "stream" • "message" • "conf_overlay_expired" • "end_send_message" • "speech_marker" • "start_play" (deprecated and replaced with

Parameter	Default	Optional	Description
			media_started) • "media_started" (sent when any media operation is started) • "end_fax" • "end_event" • "progress" • "prack_ack" • "info_ack" • "status" • "end_transfer" • "cpa" (any: any event type)
resource_id	any	*	Monitor events for a specific resource. (any: any resource id)
resource_type	any	*	Monitor events for a specific resource type. Values: • "call" • "conference" • "mrcp" • "any" (any: any resource type)

HTTP GET

Retrieves all available eventhandler resources.

`GET /eventhandlers?appid=[app_id]`

Response Payload Example

```
<web_service version="1.0">
       <eventhandlers_response size="1">
              <eventhandler_response appid="app"
                                    href="/default/eventhandlers/2d013b6d-c943-46d9-a7d5-
                                          8509cfb177ce"
                                    identifier="2d013b6d-c943-46d9-a7d5-8509cfb177ce">
                                    <eventsubscribe resource_id="any" resource_type="any"
type="any"/>
              </eventhandler_response>
       </eventhandlers_response>
```

```
</web_service>
```

HTTP POST

Creates an eventhandler resource.

```
POST /eventhandlers?appid=[app_id]
```

Request Payload Attributes

Parameter	Default	Optional	Description
eventsubscribe		*	Refer to eventsubscribe_attributes.

Request Payload Example

```
<web_service version="1.0">
    <eventhandler>
        <eventsubscribe type="any" resource_id="any" resource_type="any"/>
    </eventhandler>
</web_service>
```

Response Payload Example

```
<web_service version="1.0">
      <eventhandler_response identifier="90ba2f00-e460-4351-9203-1a2fcf25a739" appid="app"
                             href="/default/eventhandlers/90ba2f00-e460-4351-9203-1a2fcf25a739">
                             <eventsubscribe type="any" resource_id="any" resource_type="any"/>
      </eventhandler_response>
</web_service>
```

Event Handler Sub-Resource

For details on eventhandler resources, see the Event Handler Resource section.

eventhandler

Resource URI

```
/eventhandlers/[eventhandler_id]?appid=[app_id]
```

HTTP GET

Get the events.

```
GET /eventhandlers/[eventhandler_id]?appid=[app_id]
```

event_data_attributes

Parameter	Default	Optional	Description
name	(none)		Event data name.
value	(none)		Event data value. Refer to the Events section for details on what is provided in each event.

Response Payload Attributes

Parameter	Default	Optional	Description
type	(none)		Type of events to monitor. Values: • "end_play" • "end_record" • "end_playcollect" • "end_playrecord" • "end_overlay" • "end_dtmf" • "keepalive" • "incoming" • "ringing" • "connected" • "hangup" • "info" • "dtmf" • "tone" • "any" • "end_speak" • "start_of_input" • "end_recognize" • "answered" • "active_talker" • "alarm" • "stream" • "message" • "conf_overlay_expired" • "end_send_message" • "speech_marker" • "start_play" (deprecated and replaced with media_started) • "media_started" (sent when any media operation is started) • "end_fax" • "end_event"

Parameter	Default	Optional	Description
			• "progress" • "prack_ack" • "info_ack" • "status" • "end_transfer" • "cpa"
resource_id	(none)	*	Monitor events for a specific resource.
resource_type	(none)	*	Monitor events for a specific resource type. Values: • "call" • "conference" • "mrcp" • "any"
event_data	(none)	*	Values (if applicable) for all data that describe each event. Refer to event_data_attributes or the Events section for details on what is provided in each event. This can be repeated 0 ~ n.

Response Payload Example

T1:

```
200 OK
```

T2:


```
256
<web_service version="1.0">
    <event type="hangup" resource_id="c87fcca1-b2d0-49c5-8b89-baaae71cf695" resource_type="call">
           <event_data name="reason" value="5800 IPEC_SIPReasonStatusBYE" />
    </event>
</web_service>
```


T3:


```
345
<web_service version="1.0">
    <event type="incoming" resource_id="603bf73e-5e74-4c72-865a-6e498a5e2ad5" 
resource_type="call">
           <event_data name="caller_uri" value="sip:Username@10.20.129.113:5060" />
           <event_data name="uri" value="sip:sip@10.20.129.100" />
    </event>
</web_service>
```

T(N):

```
0
```

- in T2 time: the first line is the size of payload; in above example size is 256.
- in T(N) time: size is 0; it means no more events and the connection about to close.

HTTP PUT

Adds or removes an event subscription.

`PUT /eventhandlers/[eventhandler_id]?appid=[app_id]`

Request Payload Attributes

Parameter	Default	Optional	Description
eventsubscribe	(none)		Refer to eventsubscribe_attributes.

Request Payload Example

```
<web_service version="1.0">
  <eventhandler>
        <eventsubscribe action="add" type="incoming" resource_id="any" resource_type="any"/>
  </eventhandler>
</web_service>
```

Response Payload Example

```
<web_service version="1.0">
   <eventhandler_response identifier="2d013b6d-c943-46d9-a7d5-8509cfb177ce" appid="app"
                          href="/default/eventhandlers/2d013b6d-c943-46d9-a7d5-8509cfb177ce">
                          <eventsubscribe type="any" resource_id="any" resource_type="any"/>
                          <eventsubscribe type="incoming" resource_id="any" resource_type="any"/>
   </eventhandler_response>
</web_service>
```

HTTP DELETE

Deletes an eventhandler resource.

`DELETE /eventhandlers/[eventhandler_id]?appid=[app_id]`

Events

This section describes the event data that is associated with event types. Events are asynchronously returned to the application from the eventhandler.

This section describes the events that are provided by PowerMedia XMS to a RESTful application on the open TCP connection maintained between PowerMedia XMS and the RESTful application. Events are asynchronously sent to the application via the eventhandler.

keepalive

Once the application starts to monitor the events, the REST Web Service will send a "keepalive" event periodically.

Event Payload Example


```
<web_service version="1.0">
             <event type="keepalive" />
</web_service>
```


Media Events

start_play

Note: This is deprecated and replaced with media_started.

Notification event for conference play and call play. This is generated each time a play starts, including repeats. Playlists will generate an event for each element.

- **transaction_id**
- **audio_uri**
- **video_uri**

Event Payload Example


```
<web_service version="1.0">
     <event type="start_play" resourceid="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5"
resource_type="call">
            <event_data name="audio_uri" value="file://tmp/my.wav" />
            <event_data name="video_uri" value="file://tmp/my.vid" />
            <event_data name="transaction_id" value="0974c8b5-8a3c-4a8e-ae82-8f7c8bd0efdd" />
     </event>
</web_service>
```


media_started

Notification event for conference play, call play, conference record, call record, mrcp_speak, mrcp_recognize, call_overlay, fax, and send_message. This is generated each time a play starts, including repeats. Playlists will generate an event for each element.

- **transaction_id**
- **action**
  - "PLAY"
  - "RECORD"
  - "SPEAK"
  - "RECOGNIZE"
  - "OVERLAY"
  - "MESSAGE"
  - "FAX"
- **audio_uri** - play only (useful for playlist).
- **video_uri** - play only (useful for playlist).
- **gusid** - Global Unique Session Identifier.

Event Payload Example


```
<web_service version="1.0">
```

```
    <event type="start_play" resourceid="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5"
resource_type="call">
        <event_data name="audio_uri" value="file://tmp/my.wav" />
        <event_data name="video_uri" value="file://tmp/my.vid" />
        <event_data name="transaction_id" value="0974c8b5-8a3c-4a8e-ae82-8f7c8bd0efdd" />
        <event_data name="gusid" value="6962b3ec71f44deda6fe998aa2860e39" />
    </event>
</web_service>
```


end_play

Completion event for play (conference play and call play).

- **transaction_id**
- **reason**
  - "end"
  - "term-digit"
  - "stopped"
  - "max-time"
  - "error"
  - "hangup"
- **duration** - in milliseconds.
- **status** - extended error information.
- **gusid** - Global Unique Session Identifier.

Event Payload Example


```
<web_service version="1.0">
    <event type="end_play" resourceid="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5"
resource_type="call">
        <event_data name="reason" value="complete" />
        <event_data name="duration" vaule="30000ms" />
        <event_data name="transaction_id" value="0974c8b5-8a3c-4a8e-ae82-8f7c8bd0efdd" />
        <event_data name="gusid" value="6962b3ec71f44deda6fe998aa2860e39" />
    </event>
</web_service>
```


end_playcollect

Completion event for playcollect (conference play and call play).

- **transaction_id**
- **reason**
  - "term-digit"
  - "max-time"
  - "max-silence"
  - "stopped"
  - "error"

- "no-input"
- "hangup"

- **digits** - digits collected.
- **tone** - tone identifier (if reason is "tone").
- **duration** - in milliseconds.
- **status** - extended error information.
- **gusid** - Global Unique Session Identifier.

Event Payload Example


```
<web_service version="1.0">
      <event type="end_playcollect" resourceid="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5"
resource_type="call">
             <event_data name="reason" value="term-digit" />
             <event_data name="digits" value="1234" />
             <event_data name="transaction_id" value="0974c8b5-8a3c-4a8e-ae82-8f7c8bd0efdd" />
             <event_data name="gusid" value="6962b3ec71f44deda6fe998aa2860e39" />
      </event>
</web_service>
```


end_record

Completion event for record (conference play and call play).

- **transaction_id**
- **reason**
    - "term-digit"
    - "max-time"
    - "max-silence"
    - "stopped"
    - "error"
    - "no-input"
    - "hangup"
- **duration** - in milliseconds.
- **audio_location** - from a response to HTTP PUT, this is the location header value.
- **video_location** - from a response to HTTP PUT, this is the location header value.
- **status** - extended error information.
- **gusid** - Global Unique Session Identifier.
- **encrypted_key** - The AES key that was used to encrypt the recording. This value has been encrypted with the application's public key and base64 encoded.
- **encrypted_iv** - The AES initialization vector that was used to encrypt the recording. This value has been encrypted with the application's public key and base64 encoded.

Event Payload Example


```
<web_service version="1.0">
```

```
        <event type="end_record" resource_id="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5"
resource_type="call">
                <event_data name="reason" value="max-time" />
                <event_data name="transaction_id" value="0974c8b5-8a3c-4a8e-ae82-8f7c8bd0efdd" />
                <event_data name="gusid" value="6962b3ec71f44deda6fe998aa2860e39" />
        </event>
</web_service>
```

end_playrecord

Completion event for playrecord (conference play and call play).

- **transaction_id**
- **reason**
  - "term-digit"
  - "max-time"
  - "max-silence"
  - "stopped"
  - "error"
  - "no-input"
  - "hangup"
- **duration** - in milliseconds.
- **audio_location** - from a response to HTTP PUT, this is the location header value.
- **video_location** - from a response to HTTP PUT, this is the location header value.
- **status** - extended error information.
- **gusid** - Global Unique Session Identifier.
- **encrypted_key** - The AES key that was used to encrypt the recording. This value has been encrypted with the application's public key and base64 encoded.
- **encrypted_iv** - The AES initialization vector that was used to encrypt the recording. This value has been encrypted with the application's public key and base64 encoded.

Event Payload Example

```
<web_service version="1.0">
        <event type="end_playrecord" resource_id="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5"
resource_type="call">
                <event_data name="reason" value="max-time" />
                <event_data name="transaction_id" value="0974c8b5-8a3c-4a8e-ae82-8f7c8bd0efdd" />
                <event_data name="gusid" value="6962b3ec71f44deda6fe998aa2860e39" />
        </event>
</web_service>
```

end_overlay

Completion event for overlay (call overlay).

- **transaction_id**
- **reason**

- "stopped"
- "max-time"

- **duration** - in milliseconds.
- **gusid** - Global Unique Session Identifier.

Event Payload Example


```
<web_service version="1.0">
            <event type="end_overlay" resource_id="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5" resource_type="call">
                   <event_data name="reason" value=" stop " />
                   <event_data name="transaction_id" value="0974c8b5-8a3c-4a8e-ae82-8f7c8bd0efdd" />
                   <event_data name="gusid" value="6962b3ec71f44deda6fe998aa2860e39" />
            </event>
</web_service>
```


end_send_message

Completion event for send_message (send_message).

- **transaction_id**
- **reason**
  - "success"
  - "failure"
- **result**
- **duration** - in milliseconds.
- **gusid** - Global Unique Session Identifier.

Event Payload Example


```
<web_service version="1.0">
            <event type="end_send_message" resource_id="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5" resource_type="call">
                   <event_data name="reason" value="success" />
                   <event_data name="transaction_id" value="0974c8b5-8a3c-4a8e-ae82-8f7c8bd0efdd" />
                   <event_data name="result" value="success" />
                   <event_data name="duration" value="1" />
                   <event_data name="gusid" value="6962b3ec71f44deda6fe998aa2860e39" />
            </event>
</web_service>
```


end_dtmf

Completion event for call send_dtmf and conference send_dtmf.

- **transaction_id**
- **call_id**
- **reason**
  - "end"

- "stopped"
- **gusid** - Global Unique Session Identifier.

Event Payload Example


```
<web_service version="1.0">
            <event type="end_send_dtmf" resource_id="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5"
resource_type="call">
                   <event_data name="reason" value="end" />
                   <event_data name="transaction_id" value="0974c8b5-8a3c-4a8e-ae82-8f7c8bd0efdd"
/>
                   <event_data name="gusid" value="6962b3ec71f44deda6fe998aa2860e39" />
            </event>
</web_service>
```


end_event

Completion event for call send_event and conference send_event.

- **transaction_id**
- **call_id**
- **reason**
    - "end"
    - "stopped"
- **gusid** - Global Unique Session Identifier.

Event Payload Example


```
<web_service version="1.0">
            <event type="end_event" resource_id="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5"
resource_type="call">
                   <event_data name="reason" value="end" />
                   <event_data name="transaction_id" value="0974c8b5-8a3c-4a8e-ae82-8f7c8bd0efdd"
/>
                   <event_data name="gusid" value="6962b3ec71f44deda6fe998aa2860e39" />
            </event>
</web_service>
```


fax_info

Progress event for fax.

- **transaction_id**
- **reason**
    - "started
    - "polling"
    - "negotiation"
    - "page"
    - "document"
- **status** - extended error information.
    - <informational string>

- "success" (for "document" reason)
- "partial" (for "document" reason)

- **uri**
- **direction**
- **resolution**
- **page_size**
- **encoding**
- **bad_lines**
- **ecm**
- **remote_id**
- **bitrate**
- **gusid** - Global Unique Session Identifier.

Event Payload Example


```
<web_service version="1.0">
            <event type="fax_info" resource_id="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5"
resource_type="call">
                   <event_data name="reason" value="started" />
                   <event_data name="status" value="success" />
                   <event_data name="uri" value="file://tmp/my.pdf" />
                   <event_data name="direction" value="send" />
                   <event_data name="transaction_id" value="0974c8b5-8a3c-4a8e-ae82-8f7c8bd0efdd"
/>
                   <event_data name="gusid" value="6962b3ec71f44deda6fe998aa2860e39" />
            </event>
</web_service>
```


end_fax

Completion event for fax.

- **transaction_id**
- **reason**
    - "end"
    - "stopped"
    - "max-time"
    - "error"
    - "hangup"
- **duration** - in milliseconds.
- **status** - extended error information.
- **gusid** - Global Unique Session Identifier.

Event Payload Example


```
<web_service version="1.0">
```

```
            <event type="end_fax" resource_id="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5"
resource_type="call">
                   <event_data name="reason" value="end" />
                   <event_data name="status" value="success" />
                   <event_data name="duration" value="2000" />
                   <event_data name="transaction_id" value="0974c8b5-8a3c-4a8e-ae82-8f7c8bd0efdd"
/>
                   <event_data name="gusid" value="6962b3ec71f44deda6fe998aa2860e39" />
            </event>
</web_service>
```


Call Events

incoming

A new inbound call.

- **call_id**
- **uri**
- **caller_uri**
- **name** - application name, from routing rule.
- **headers** - raw SIP headers, delimited by the <CR><LF> end-of-line characters.
- **headers.<NAME>** - individual SIP header.
- **content_type** - mime type of content.
- **content** - optional content.
- **gusid** - Global Unique Session Identifier.

Event Payload Example


```
<web_service version="1.0">
             <event type="incoming" resource_id="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5"
resource_type="call">
                    <event_data name="call_id" value=" 5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5" />
                    <event_data name="uri" value="sip:sip@10.20.129.8" />
                    <event_data name="caller_uri" value="sip:frank@10.20.129.20" />
                    <event_data name="gusid" value="6962b3ec71f44deda6fe998aa2860e39" />
             </event>
</web_service>
```


ringing

The remote party of an outbound call is ringing.

- **call_id**
- **prack_level -** prack level from remote: "required" or "none".
- **gusid** - Global Unique Session Identifier.

Event Payload Example


```
<web_service version="1.0">
<event type="ringing" resource_id="ce926447-8fff-4d4d-bfd9-0ca57c461bfb" resource_type="call">
<event_data name="call_id" value="ce926447-8fff-4d4d-bfd9-0ca57c461bfb" />
```

```
&lt;event_data name="prack_level" value="required" /&gt;
&lt;event_data name="type" value="RINGING" /&gt;
&lt;event_data name="gusid" value="6962b3ec71f44deda6fe998aa2860e39" /&gt;
&lt;/event&gt;
&lt;/web_service&gt;
```

connected

The remote party of an outbound call has answered.

- **call_id**
- **reason**
    - "unknown"
    - "answer-machine"
    - "voice"
    - "ced" (fax detection)
    - custom tone name
- **media**
    - "audio"
    - "audiovideo"
- **caller_uri** ('From' header)
- **called_uri** ('To' header)
- **gusid** - Global Unique Session Identifier.

Event Payload Example

```
&lt;web_service version="1.0"&gt;
            <event type="connected" resource_id="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5"
resource_type="call">
                    &lt;event_data name="reason" value="voice" /&gt;
                    &lt;event_data name="media" value="audio" /&gt;
                    &lt;event_data name="media" value="audio" /&gt;
            &lt;/event&gt;
&lt;/web_service&gt;
```

message

A party has sent the message. Generic message. May be related to an existing call or standalone.

- **call_id**
- **uri** (outside of existing call only).
- **caller_uri** (outside of existing call only).
- **called_uri** (outside of existing call only).
- **name** (application name, from the routing rule, outside of existing call only).
- **content_type** - mime type of **content** (alias for **content[0].content_type**).
- **content** - data (alias for **content[0].content**).
- **content_count** - number of elements in the contents[] array.

- **content[n].content_type** - mime type of **content[n].content**.
- **content[n].content** - data.
- **content[n].content_id** - content identifier.
- **multipart_type** - mime type for multi part content.
- **msg_payload_count** - number of elements in the msg_payload[] array.
- **msg_payload[n].content_type** - mime type of **msg_payload[n].content**.
- **msg_payload[n].content** - message data.
- **msg_payload[n].content_id** - content identifier.
- **msg_payload[n].content_disposition** - disposition of **msg_payload[n].uri**.
- **msg_payload[n].uri** - file uri.
- **gusid** - Global Unique Session Identifier.

Event Payload Example


```
<web_service version="1.0">
            <event type="message" resource_id="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5"
resource_type="call">
                   <event_data name="call_id" value="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5" />
                   <event_data name="msg_payload_count" value="1" />
                   <event_data name="msg_payload[0].content_id" value="id1" />
                   <event_data name="msg_payload[0].content_type" value="text/plain" />
                   <event_data name="msg_payload[0].content" value="data" />
                   <event_data name="msg_payload[0].content_disposition" value="attachment" />
                   <event_data name="gusid" value="6962b3ec71f44deda6fe998aa2860e39" />
            </event>
</web_service>
```


hangup

An outbound call request has failed or the remote party has ended the call.

All call resources are released automatically, and if the call is joined to another call, the other party will be automatically unjoined. No further actions may be performed on the call.

- **call_id**
- **reason**
  - "busy-tone"
  - "operator-intercept"
  - "no-answer"
- **content_type**
- **content**
- **gusid** - Global Unique Session Identifier.

Event Payload Example


```
<web_service version="1.0">
            <event type="hangup" resource_id="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5"
resource_type="call">
```

```
                <event_data name="reason" value=”no-answer" />
                <event_data name="gusid" value="6962b3ec71f44deda6fe998aa2860e39" />
        </event>
</web_service>
```

info

Unsolicited user information (for example, SIP INFO).

- **call_id**
- **content_type** - mime type of content (alias for **content[0].content_type**).
- **content** - data (alias for **content[0].content**).
- **content_id** - content identifier (alias of **content[0].content_id**).
- **content_count** - number of elements in the contents[] array.
- **content[n].content_type** - mime type of **content[n].content**.
- **content[n].content** - data.
- **content[n].content_id** - content identifier.
- **multipart_type** - mime type for multi part content.
- **gusid** - Global Unique Session Identifier.

Event Payload Example

```
<web_service version="1.0">
        <event type="info" resource_id="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5"
resource_type="call">
                <event_data name="call_id" value="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5" />
                <event_data name="content-type" value="sdp" />
                <event_data name="content" value="[sdp]" />
                <event_data name="gusid" value="6962b3ec71f44deda6fe998aa2860e39" />
        </event>
</web_service>
```

info_ack

Remote acknowledgement for send_info.

- **call_id**
- **reason**
  - "**success**"
  - "**error**"
- **status** - string format: "XMXERR-code Code Informative-Only-Text"
- **gusid** - Global Unique Session Identifier.

XMXERR-code	Code	Informative-Only-Text
"11" (XMSERR_SERVER_ERROR)	"481"	"Call/Transaction Does Not Exist"
"11" (XMSERR_SERVER_ERROR)	"0"	"Internal Error"

- **content_type** - mime type of content (alias for **content[0].content_type**).
- **content** - data (alias for **content[0].content**).
- **content_id** - content identifier (alias of **content[0].content_id**).
- **content_count** - number of elements in the contents[] array.
- **content[n].content_type** - mime type of **content[n].content**.
- **content[n].content** - data.
- **content[n].content_id** - content identifier.
- **multipart_type** - mime type for multi part content.

Event Payload Example


```
<web_service version="1.0">
            <event type="info_ack" resource_id="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5"
resource_type="call">
                   <event_data name="call_id" value="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5" />
                   <event_data name="reason" value="success" />
                   <event_data name="gusid" value="6962b3ec71f44deda6fe998aa2860e39" />
            </event>
</web_service>
```


dtmf

Unsolicited DTMF digits.

- **call_id** or **conference_id**
- **digits**

Event Payload Example


```
<web_service version="1.0">
            <event type="dtmf" resource_id="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5"
resource_type="call">
                   <event_data name="call_id" value=" 5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5" />
                   <event_data name="digits" value="4" />
            </event>
</web_service>
```


tone

Unsolicited tone detection events.

- **call_id** or **conference_id**
- **tone**
- **gusid** - Global Unique Session Identifier.

Event Payload Example


```
<web_service version="1.0">
            <event type="tone" resource_id="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5"
resource_type="call">
                   <event_data name="call_id" value="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5" />
                   <event_data name="tone" value="[name]" />
```

```
                    <event_data name="gusid" value="6962b3ec71f44deda6fe998aa2860e39" />
             </event>
</web_service>
```


updated

Indicates that the call's media has been updated (for example, SIP reINVITE).

- **call_id**
- **media**
  - "unknown"
  - "audio"
  - "video"
  - "audiovideo"
- **audio**
  - "inactive"
  - "sendonly"
  - "recvonly"
  - "sendrecv"
- **video**
  - "inactive"
  - "sendonly"
  - "recvonly"
  - "sendrecv"
- **gusid** - Global Unique Session Identifier.

Event Payload Example


```
<web_service version="1.0">
             <event type="tone" resource_id="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5"
resource_type="call">
                    <event_data name="call_id" value="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5" />
                    <event_data name="media" value="audiovideo" />
                    <event_data name="audio" value="sendrecv" />
                    <event_data name="video" value="sendrecv" />
                    <event_data name="gusid" value="6962b3ec71f44deda6fe998aa2860e39" />
             </event>
</web_service>
```


alarm

Generic alarm notification.

- **call_id**
- **alarm**
  - "rtp-timeout"
  - "rtcp-timeout"

- **state**
  - "on"
  - "off"
- **gusid** - Global Unique Session Identifier.

Event Payload Example


```
<web_service version="1.0">
            <event type="alarm" resource_id="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5"
resource_type="call">
                   <event_data name="call_id" value="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5" />
                   <event_data name="alarm" value="rtp-timeout" />
                   <event_data name="state" value="off" />
                   <event_data name="gusid" value="6962b3ec71f44deda6fe998aa2860e39" />
            </event>
</web_service>
```


stream

RTP streaming status notification.

- **call_id**
- **state**
  - "started"
  - "stopped"
- **gusid** - Global Unique Session Identifier.

Event Payload Example


```
<web_service version="1.0">
            <event type="alarm" resource_id="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5"
resource_type="call">
                   <event_data name="call_id" value="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5" />
                   <event_data name="state" value="started" />
                   <event_data name="gusid" value="6962b3ec71f44deda6fe998aa2860e39" />
            </event>
</web_service>
```


prack

Provisional Acknowledgement received on an accepted *inbound* call.

- **call_id**
- **gusid** - Global Unique Session Identifier.

Event Payload Example


```
<web_service version="1.0">
            <event type="prack" resource_id="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5"
resource_type="call">
                   <event_data name="call_id" value="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5" />
                   <event_data name="gusid" value="6962b3ec71f44deda6fe998aa2860e39" />
            </event>
```

```
</web_service>
```

accepted

This event indicates that an asynchronous xms_accept request has completed successfully. It is also signaled while handling an asynchronous xms_answer on an offering call, to indicate the completion of the accept phase.

- **call_id**
- **caller_uri** ("From" header)
- **called_uri** ("To" header)
- **gusid** - Global Unique Session Identifier.

Event Payload Example


```
<web_service version="1.0">
     <event type="accepted" resource_id="a2c91ebd-074f-4319-8d5d-5f431add31c0"
resource_type="call">
       <event_data name="app_data" value="" />
       <event_data name="called_uri" value="<sip:sip@10.20.129.112>;tag=f7064c00-7081140a-13c4-
50022-304f-308fc5d9-304f" />
       <event_data name="caller_uri" value=""Your Full
Name"<sip:Username@10.20.129.83:5060>;tag=E08E00D6B4090B7C65DE6AEA4880EA8B" />
                    <event_data name="gusid" value="6962b3ec71f44deda6fe998aa2860e39" />
     </event>
</web_service>
```


answered

Indicates that an asynchronous answer completed.

- **call_id**
- **media**
  - (none)
  - "audio"
  - "video"
  - "audiovideo"
- **audio**
  - "inactive"
  - "sendonly"
  - "recvonly"
  - "sendrecv"
- **video**
  - "inactive"
  - "sendonly"
  - "recvonly"
  - "sendrecv"
- **caller_uri** ('From' header)

- **called_uri** ('To' header)
- **gusid** - Global Unique Session Identifier.

Event Payload Example


```
<web_service version="1.0">
     <event type="answered" resource_id="a2c91ebd-074f-4319-8d5d-5f431add31c0"
resource_type="call">
           <event_data name="audio" value="sendrecv" />
           <event_data name="called_uri" value="<sip:sip@10.20.129.112>;tag=f7064c00-7081140a-
13c4-50022-304f-308fc5d9-304f" />
           <event_data name="caller_uri" value=""Your Full
Name"<sip:Username@10.20.129.83:5060>;tag=E08E00D6B4090B7C65DE6AEA4880EA8B" />
           <event_data name="media" value="audiovideo" />
           <event_data name="message" value="" />
           <event_data name="video" value="sendrecv" />
           <event_data name="gusid" value="6962b3ec71f44deda6fe998aa2860e39" />
     </event>
</web_service>
```


prack_ack

Remote acknowledgment of a sent PRACK in an outbound call.

- **call_id**
- **gusid** - Global Unique Session Identifier.

Event Payload Example


```
<web_service version="1.0">
<event type="prack_ack" resource_id="ce926447-8fff-4d4d-bfd9-0ca57c461bfb" resource_type="call">
<event_data name="type" value="PRACK_ACK" />
</event>
</web_service>
```


progress

Session progress (SIP 183) received from the remote party of an outbound call.

- **call_id**
- **prack_level** (prack level from remote: required or none)
- **gusid** - Global Unique Session Identifier.

Event Payload Example


```
<web_service version="1.0">
<event type="progress" resource_id="ce926447-8fff-4d4d-bfd9-0ca57c461bfb" resource_type="call">
<event_data name="call_id" value="ce926447-8fff-4d4d-bfd9-0ca57c461bfb" />
<event_data name="prack_level" value="required" />
<event_data name="type" value="RINGING" />
<event_data name="gusid" value="6962b3ec71f44deda6fe998aa2860e39" />
</event>
</web_service>
```

end_transfer

Completion event for transfer.

- **call_id**
- **reason**
- **gusid** - Global Unique Session Identifier.

Event Payload Example


```
<web_service version="1.0">
<event type="end_transfer" resource_id="ce926447-8fff-4d4d-bfd9-0ca57c461bfb"
resource_type="call">
<event_data name="call_id" value="ce926447-8fff-4d4d-bfd9-0ca57c461bfb" />
<event_data name="reason" value="required" />
<event_data name="gusid" value="6962b3ec71f44deda6fe998aa2860e39" />
</event>
</web_service>
```


status

Status of the call.

- **call_id**
- **reason**
    - "success"
    - "error"
- **status** - string format: "XMXERR-code Code Informative-Only-Text"
- **gusid** - Global Unique Session Identifier.

XMSERR-code	Code	Informative-Only-Text
"11" (XMSERR_SERVER_ERROR)	"481"	"Call/Transaction Does Not Exist"
"11" (XMSERR_SERVER_ERROR)	"0"	"Internal Error"

Event Payload Example


```
<web_service version="1.0">
<event type="status" resource_id="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5" resource_type="call">
<event_data name="call_id" value=" 5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5" />
<event_data name="reason" value="success" />
<event_data name="gusid" value="6962b3ec71f44deda6fe998aa2860e39" />
</event>
</web_service>
```


cpa

The call progress analysis.

- **call_id**
- **reason** - CPA completion reason. See the *Dialogic® PowerMedia™ XMS Installation and Configuration Guide* for more information.

- **gusid** - Global Unique Session Identifier.

Event Payload Example


```
<web_service version="1.0">
<event type="cap" resource_id="ce926447-8fff-4d4d-bfd9-0ca57c461bfb" resource_type="call">
<event_data name="call_id" value="ce926447-8fff-4d4d-bfd9-0ca57c461bfb" />
<event_data name="reason" value="voice" />
<event_data name="gusid" value="6962b3ec71f44deda6fe998aa2860e39" />
</event>
</web_service>
```


MRCP Events

end_speak

Completion event for mrcp_speak.

- **mrcp_id**
- **id**
- **transaction_id**
- **reason** - raw result from MRCP server.
- **duration** - in milliseconds.
- **gusid** - Global Unique Session Identifier.

Event Payload Example


```
<web_service version="1.0">
             <event type="tone" resource_id="97d105ae-30b9-45e2-acdf-82e67d9b34aa" 
resource_type="mrcp">
                    <event_data name="id" value="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5" />
                    <event_data name="mrcp_id" value="97d105ae-30b9-45e2-acdf-82e67d9b34aa" />
                    <event_data name="transaction_id" value="0974c8b5-8a3c-4a8e-ae82-
8f7c8bd0efdd" />
                    <event_data name="reason" value="anything" />
                    <event_data name="duration" value="20s" />
                    <event_data name="gusid" value="6962b3ec71f44deda6fe998aa2860e39" />
             </event>
</web_service>
```


speech_marker

- **mrcp_id**
- **id**
- **transaction_id**
- **name**
- **timestamp**
- **gusid** - Global Unique Session Identifier.

Event Payload Example


```
<web_service version="1.0">
```

```
            <event type="speech_marker" resource_id="97d105ae-30b9-45e2-acdf-82e67d9b34aa"
resource_type="mrcp">
                    &lt;event_data name="id" value="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5" /&gt;
                    &lt;event_data name="mrcp_id" value="97d105ae-30b9-45e2-acdf-82e67d9b34aa" /&gt;
                    <event_data name="transaction_id" value="0974c8b5-8a3c-4a8e-ae82-
8f7c8bd0efdd" />
                    &lt;event_data name="name" value="xyz" /&gt;
                    &lt;event_data name="timestamp" value="2014-2-2 1900:00" /&gt;
                    &lt;event_data name="gusid" value="6962b3ec71f44deda6fe998aa2860e39" /&gt;
            &lt;/event&gt;
&lt;/web_service&gt;
```

end_recognize

Completion event for mrcp_recognize.

- **mrcp_id**
- **id**
- **transaction_id**
- **reason** - raw result from MRCP server.
- **duration** - in milliseconds.
- **content**
- **content_type**
- **waveform_uri**
- **gusid** - Global Unique Session Identifier.

Event Payload Example

```
&lt;web_service version="1.0"&gt;
            <event type="tone" resource_id="97d105ae-30b9-45e2-acdf-82e67d9b34aa"
resource_type="mrcp">
                    &lt;event_data name="id" value="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5" /&gt;
                    &lt;event_data name="mrcp_id" value="97d105ae-30b9-45e2-acdf-82e67d9b34aa" /&gt;
                    <event_data name="transaction_id" value="0974c8b5-8a3c-4a8e-ae82-
8f7c8bd0efdd" />
                    &lt;event_data name="reason" value="anything" /&gt;
                    &lt;event_data name="duration" value="20s" /&gt;
                    &lt;event_data name="content" value="data" /&gt;
                    &lt;event_data name="content_type" value="plain/text" /&gt;
                    &lt;event_data name="gusid" value="6962b3ec71f44deda6fe998aa2860e39" /&gt;
            &lt;/event&gt;
&lt;/web_service&gt;
```

start_of_input

Input detected event for mrcp_recognize.

- **mrcp_id**
- **id**
- **transaction_id**

- **gusid** - Global Unique Session Identifier.

Event Payload Example


```
<web_service version="1.0">
            <event type="tone" resource_id="97d105ae-30b9-45e2-acdf-82e67d9b34aa" 
resource_type="mrcp">
                   <event_data name="id" value="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5" />
                   <event_data name="mrcp_id" value="97d105ae-30b9-45e2-acdf-82e67d9b34aa" />
                   <event_data name="transaction_id" value="0974c8b5-8a3c-4a8e-ae82-
8f7c8bd0efdd" />
                   <event_data name="gusid" value="6962b3ec71f44deda6fe998aa2860e39" />
            </event>
</web_service>
```


Conference Events

active_talker

- **conf_id**
- **talkers** - space delimited list of call IDs.
- **gusid** - Global Unique Session Identifier.

Event Payload Example


```
<web_service version="1.0">
            <event type="active_talker" resource_id="97d105ae-30b9-45e2-acdf-82e67d9b34aa" 
resource_type="conference">
                   <event_data name="conf_id" value="97d105ae-30b9-45e2-acdf-82e67d9b34aa" />
                   <event_data name="talkers" value="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5 
6798c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5" />
                   <event_data name="gusid" value="6962b3ec71f44deda6fe998aa2860e39" />
            </event>
</web_service>
```


conf_overlay_expired

- **conf_id** - identifies the conference.
- **region** - identifies the region that contained the overlay that expired.
- **overlay_id** - unique identifier for the text or image overlay that expired.
- **content_id** - unique identifier for the content that expired.

Event Payload Example


```
<web_service version="1.0">
            <event type="conf_overlay_expired" resource_id="97d105ae-30b9-45e2-acdf-
82e67d9b34aa" resource_type="conference">
                   <event_data name="conf_id" value="97d105ae-30b9-45e2-acdf-82e67d9b34aa" />
                   <event_data name="region" value=v2" />
                   <event_data name="overlay_id" value="2" />
                   <event_data name="content_id" value="2" />
            </event>
</web_service>
```

MRB Events

resource_ended

Notification event that the resource has been deleted by the PowerMedia MRB without being triggered by the application, such as when the calls/conferences are ended by the user in the Console or if the media server storing the resource has failed and the PowerMedia MRB could not find another media server to move to. Note that this event is not specifically subscribed to. Instead, the PowerMedia MRB will send the event if it has any subscription that covers that particular resource.

- **reason**

Event Payload Example


```
<web_service version="1.0">
    <event type="resource_ended" resource_id="1234" resource_type="call">
        <event_data name="reason" value="Ended via MRB admin UI"/>
    </event>
</web_service>
```


See the *Dialogic® PowerMedia™ Media Resource Broker (MRB) Technology Guide* for more information.

MRCP Resource

The Media Resource Control Protocol (MRCP), is used by PowerMedia XMS as an interface to Automatic Speech Recognition (ASR) and Text-to-Speech (TTS) systems. MRCP provides an easy way to build voice user interfaces, allowing a grammar to be built for speech input and providing a way to easily translate text into voice prompts without reading and recording them.

For MRCP sub-resources, see the MRCP Sub-Resource section.

The following tables show the HTTP methods that can be used with MRCP.

Note: The payloads shown are examples only as there are many possible variations.

mrcps

Resource URI

`/mrcps?appid=[app_id]`

HTTP GET

Retrieves all available MRCP resources.

`GET /mrcps?appid=[app_id]`

Response Payload Example


```
<web_service version="1.0">
   <mrcps_response size="1">
              <mrcps_response size="1">
                     <mrcp_response appid="app" asr="yes" tts="yes"
                                   href="/default/mrcps/8d200e91-8b56-465d-9e0b-6d9f05fb4ee3"
                                   identifier="8d200e91-8b56-465d-9e0b-6d9f05fb4ee3">
                     </mrcp_response>
              </mrcps_response>
```

```
  </mrcps_response>
</web_service>
```

HTTP POST

Creates a MRCP resource.

`POST /mrcps?appid=[app_id]`

Request Payload Attributes

Parameter	Default	Optional	Description
asr	"yes"	*	Specifies if a speech recognizer resource is required. Values: • "yes" • "no"
tts	"yes"	*	Specifies if a speech synthesizer resource is required. Values: • "yes" • "no"
gusid		*	Specifies the Global Unique Session Identifier (gusid).

Request Payload Example

```
<web_service version="1.0">
  <mrcp asr="yes" tts="yes"/>
</web_service>
```

Response Payload Example

```
<web_service version="1.0">
              <mrcp_response appid="app" asr="yes" tts="yes"
                             href="/default/mrcps/8d200e91-8b56-465d-9e0b-6d9f05fb4ee3"
                             identifier="8d200e91-8b56-465d-9e0b-6d9f05fb4ee3">
              </mrcp_response>
</web_service>
```

MRCP Sub-Resource

For details on MRCP resources, see the MRCP Resource section.

mrcp

Resource URI

`/mrcps/[mrcp_id]?appid=[app_id]`

HTTP GET

Retrieves an available MRCP resource.

`GET /mrcps/[mrcp_id]?appid=[app_id]`

Response Payload Example


```
<web_service version="1.0">
            <mrcp_response appid="app" asr="yes" tts="yes"
                          href="/default/mrcps/8d200e91-8b56-465d-9e0b-6d9f05fb4ee3"
                          identifier="8d200e91-8b56-465d-9e0b-6d9f05fb4ee3">
            </mrcp_response>
</web_service>
```


HTTP PUT

Updates a MRCP resource.

`PUT /mrcps/[mrcp_id]?appid=[app_id]`

Perform MRCP Action

- speak
- recognize
- set-asr-param/set-tts-param
- get-asr-param/get-tts-param
- define-grammar
- mrcp-update-action
- get_last_action
- get_last_event

param_attributes

Parameter	Default	Optional	Description
name	(none)		Parameter name.
value	(none)		Parameter value.

speak

Synthesizes speech (TTS).

Request Payload Attributes

Parameter	Default	Optional	Description
call_id	(none)		The call receiving the synthesized speech.
param	(none)	*	Generic parameter. Refer to param_attributes can be repeated 0 ~ n.

Parameter	Default	Optional	Description
barge	"yes"	*	Sets whether the synthesis can be barged. Values: • "yes" • "no"
locale	"en-US"	*	Language and country code. See RFC 3066.
content	(none)		Content to be synthesized.
content_type	(none)		Mime type of the content.

Request Payload Example


```
<web_service version="1.0">
  <mrcp>
     <mrcp_action>
          <speak call_id="c87fcca1-b2d0-49c5-8b89-baaae71cf695" barge="yes" content="Hello World"
content_type="text/plain"/>
     </mrcp_action>
  </mrcp>
</web_service>
```


Response Payload Example


```
<web_service version="1.0">
       <mrcp_response identifier="8d200e91-8b56-465d-9e0b-6d9f05fb4ee3" appid="app"
                      href="/default/mrcps/8d200e91-8b56-465d-9e0b-6d9f05fb4ee3"
                      asr="yes" tts="yes">
                      <mrcp_action>
                         <speak transaction_id="b9e5c859-e5ed-44fd-bf91-b636ba8b14f0"
                                call_id="c87fcca1-b2d0-49c5-8b89-baaae71cf695"
                                barge="yes"
                                locale="en-US"
                                content="Hello World"
                                content_type="text/plain">
                         </speak>
                      </mrcp_action>
       </mrcp_response>
</web_service>
```


recognize

Recognizes speech (ASR).

Request Payload Attributes

Parameter	Default	Optional	Description
call_id	(none)		The call sourcing the media to be recognized.
param	(none)	*	Generic parameter. Refer to param_attributes can be repeated 0 ~ n.
grammar	(none)	*	Grammar for recognizer.
grammar_type	(none)	*	Mime type of the grammar.
grammar_id	(none)	*	Application identifier for a previously defined grammar.
timeout	"infinite"	*	The maximum length of time to wait for input.

Request Payload Example

```
<web_service version="1.0">
 <mrcp>
    <mrcp_action>
         <recognize call_id="c87fcca1-b2d0-49c5-8b89-baaae71cf695"
                    grammar="grammar" grammar_type="type" grammar_id="text/plain"
timeout="infinite"/>
    </mrcp_action>
 </mrcp>
</web_service>
```

Response Payload Example

```
<web_service version="1.0">
       <mrcp_response identifier="8d200e91-8b56-465d-9e0b-6d9f05fb4ee3" appid="app"
                      href="/default/mrcps/8d200e91-8b56-465d-9e0b-6d9f05fb4ee3"
                      asr="yes" tts="yes">
                      <mrcp_action>
                         <recognize transaction_id="842722ef-75d0-407d-9b52-29c5270868bb"
                                    call_id="c87fcca1-b2d0-49c5-8b89-baaae71cf695"
                                    grammar="grammar"
                                    grammar_type="type"
                                    grammar_id="text/plain"
                                    timeout="infinite">
                         </recognize>
```

```
                    </mrcp_action>
       </mrcp_response>
</web_service>
```

set-asr-param/set-tts-param

Request Payload Attributes

Parameter	Default	Optional	Description
param	(none)	*	Generic parameter. Value has the format: name=value, name2=value2.

Request Payload Example

set-asr-param

```
<web_service version="1.0">
  <mrcp>
      <mrcp_action>
          <set-asr-param>
               <param name="name" value="value"/>
          </set-asr-param>
      </mrcp_action>
  </mrcp>
</web_service>
```

set-tts-param

```
<web_service version="1.0">
  <mrcp>
      <mrcp_action>
          <set-tts-param>
               <param name="name" value="value"/>
          </set-tts-param>
      </mrcp_action>
  </mrcp>
</web_service>
```

Response Payload Example

set-asr-param

```
<web_service version="1.0">
       <mrcp_response identifier="8d200e91-8b56-465d-9e0b-6d9f05fb4ee3" appid="app"
                      href="/default/mrcps/8d200e91-8b56-465d-9e0b-6d9f05fb4ee3"
                      asr="yes" tts="yes">
                      <mrcp_action>
                         <set-asr-param>
                             <param name="name" value="value"/>
                         </set-asr-param>
                      </mrcp_action>
       </mrcp_response>
</web_service>
```

set-tts-param

```
<web_service version="1.0">
       <mrcp_response identifier="8d200e91-8b56-465d-9e0b-6d9f05fb4ee3" appid="app"
                      href="/default/mrcps/8d200e91-8b56-465d-9e0b-6d9f05fb4ee3"
                      asr="yes" tts="yes">
                      <mrcp_action>
                         <set-tts-param>
                             <param name="name" value="value"/>
                         </set-tts-param>
                      </mrcp_action>
       </mrcp_response>
</web_service>
```

get-asr-param/get-tts-param

Request Payload Example

get-asr-param

```
<web_service version="1.0">
  <mrcp>
      <mrcp_action>
          <get-asr-param/>
      </mrcp_action>
  </mrcp>
</web_service>
```

get-tts-param

```
<web_service version="1.0">
  <mrcp>
      <mrcp_action>
          <get-tts-param/>
      </mrcp_action>
  </mrcp>
</web_service>
```

Response Payload Example

get-asr-param

```
<web_service version="1.0">
       <mrcp_response identifier="8d200e91-8b56-465d-9e0b-6d9f05fb4ee3" appid="app"
                      href="/default/mrcps/8d200e91-8b56-465d-9e0b-6d9f05fb4ee3"
                      asr="yes" tts="yes">
                      <mrcp_action>
                         <get-asr-param>
                             <param name="name" value="value"/>
                         </get-asr-param>
                      </mrcp_action>
       </mrcp_response>
</web_service>
```

get-tts-param

```
<web_service version="1.0">
       <mrcp_response identifier="8d200e91-8b56-465d-9e0b-6d9f05fb4ee3" appid="app"
                      href="/default/mrcps/8d200e91-8b56-465d-9e0b-6d9f05fb4ee3"
                      asr="yes" tts="yes">
                      <mrcp_action>
                         <get-tts-param>
                             <param name="name" value="value"/>
                         </get-tts-param>
                      </mrcp_action>
       </mrcp_response>
</web_service>
```

define-grammar

Request Payload Attributes

Parameter	Default	Optional	Description
grammar	(none)	*	Grammar for recognizer.
grammar_type	(none)	*	Mime type of the grammar.
grammar_id	(none)	*	Application identifier for a previously defined grammar.

Request Payload Example

```
<web_service version="1.0">
 <mrcp>
    <mrcp_action>
         <define-grammar  grammar="grammar" grammar_type="type" grammar_id="id"/>
    </mrcp_action>
 </mrcp>
</web_service>
```

Response Payload Example

```
<web_service version="1.0">
       <mrcp_response identifier="8d200e91-8b56-465d-9e0b-6d9f05fb4ee3" appid="app"
                      href="/default/mrcps/8d200e91-8b56-465d-9e0b-6d9f05fb4ee3"
                      asr="yes" tts="yes">
                      <mrcp_action>
                         <define-grammar  grammar="grammar" grammar_type="type" grammar_id="id"/>

                      </mrcp_action>
       </mrcp_response>
</web_service>
```

mrcp-update-action

Request Payload Attributes

Parameter	Default	Optional	Description
action	(none)		Values: • "stop-speak" • "stop-recognize" • "pause" (speak only) • "resume" (speak only) • "barge" (speak only) • "start-input-timers" (recognize only)

Request Payload Example

```
<web_service version="1.0">
 <mrcp>
    <mrcp_action>
         <mrcp-update-action action="pause"/>
    </mrcp_action>
 </mrcp>
</web_service>
```

Response Payload Example

```
<web_service version="1.0">
       <mrcp_response identifier="8d200e91-8b56-465d-9e0b-6d9f05fb4ee3" appid="app"
                      href="/default/mrcps/8d200e91-8b56-465d-9e0b-6d9f05fb4ee3"
                      asr="yes" tts="yes">
                      <mrcp_action>
                         <mrcp-update-action action="pause"/>
                      </mrcp_action>
       </mrcp_response>
</web_service>
```

get_last_action

To get the last performed action.

Request Payload Example

```
<web_service version="1.0">
  <mrcp>
    <mrcp_action>
       <get_last_action/>
    </mrcp_action>
  </mrcp>
</web_service>
```

Response Payload Example


```
<web_service version="1.0">
        <mrcp_response identifier="8d200e91-8b56-465d-9e0b-6d9f05fb4ee3" appid="app"
                      href="/default/mrcps/8d200e91-8b56-465d-9e0b-6d9f05fb4ee3"
                      asr="yes" tts="yes">
                      <mrcp_action>
                            <get_last_action>
                                 <mrcp-update-action action="pause"/>
                            </get_last_action>
                      </mrcp_action>
       </mrcp_response>
</web_service>
```


get_last_event

To get the last event.

Request Payload Example


```
<web_service version="1.0">
  <mrcp>
    <mrcp_action>
       <get_last_event/>
    </mrcp_action>
  </mrcp>
</web_service>
```


Response Payload Example


```
<web_service version="1.0">
       <event type="start_of_input" resource_id="97d105ae-30b9-45e2-acdf-82e67d9b34aa"
resource_type="mrcp">
               <event_data name="call_id" value="5974c8b5-8a3c-4a8e-ae82-8f7c8bd0efd5" />
               <event_data name="mrcp_id" value="97d105ae-30b9-45e2-acdf-82e67d9b34aa" />
               <event_data name="transaction_id" value="0974c8b5-8a3c-4a8e-ae82-8f7c8bd0efdd" />
       </event>
</web_service>
```


HTTP DELETE

Deletes a MRCP resource.

```
DELETE /mrcps/[mrcp_id]?appid=[app_id]
```

6. XML Schema Definition of Elements

This section contains the complete XML schema definition of elements (XSD).

Note: This schema definition may occasionally be updated. Always use the XSD (*xmsrest.xsd*) available with the current PowerMedia XMS version in the */etc/xms* directory.

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

        <xs:simpleType name="boolean_type">
                <xs:restriction base="xs:string">
                        <xs:enumeration value="yes" />
                        <xs:enumeration value="no" />
                </xs:restriction>
        </xs:simpleType>

        <xs:simpleType name="action_option">
                <xs:restriction base="xs:string">
                        <xs:enumeration value="add" />
                        <xs:enumeration value="remove" />
                </xs:restriction>
        </xs:simpleType>

        <xs:simpleType name="call_type_option">
                <xs:restriction base="xs:string">
                        <xs:enumeration value="inbound" />
                        <xs:enumeration value="outbound" />
                        <xs:enumeration value="3pcc" />
                </xs:restriction>
        </xs:simpleType>

        <xs:simpleType name="media_type">
                <xs:restriction base="xs:string">
                        <xs:enumeration value="audio" />
                        <xs:enumeration value="video" />
                        <xs:enumeration value="audiovideo" />
                        <xs:enumeration value="message" />
                        <xs:enumeration value="image" />
                        <xs:enumeration value="unknown" />
                </xs:restriction>
        </xs:simpleType>

        <xs:simpleType name="media_direction">
                <xs:restriction base="xs:string">
                        <xs:enumeration value="inactive" />
```

```
				<xs:enumeration value="sendonly" />
				<xs:enumeration value="recvonly" />
				<xs:enumeration value="sendrecv" />
			</xs:restriction>
		</xs:simpleType>

		<xs:simpleType name="audio_codec_option">
			<xs:restriction base="xs:string">
				<xs:enumeration value="L8" />
				<xs:enumeration value="L16" />
				<xs:enumeration value="mulaw" />
				<xs:enumeration value="alaw" />
				<xs:enumeration value="AMR" />
				<xs:enumeration value="AMR-WB" />
				<xs:enumeration value="OPUS" />
			</xs:restriction>
		</xs:simpleType>

		<xs:simpleType name="audio_rate_option">
			<xs:restriction base="xs:string">
				<xs:enumeration value="8000" />
				<xs:enumeration value="11025" />
				<xs:enumeration value="16000" />
			</xs:restriction>
		</xs:simpleType>

		<xs:simpleType name="video_codec_option">
			<xs:restriction base="xs:string">
				<xs:enumeration value="h264" />
				<xs:enumeration value="h263" />
				<xs:enumeration value="h263-1998" />
				<xs:enumeration value="mp4v-es" />
				<xs:enumeration value="jpeg" />
				<xs:enumeration value="vp8" />
				<xs:enumeration value="vp9" />
			</xs:restriction>
		</xs:simpleType>

		<xs:simpleType name="video_type_option">
			<xs:restriction base="xs:string">
				<xs:enumeration value="video/x-vid" />
				<xs:enumeration value="video/3gpp" />
				<xs:enumeration value="image/jpeg" />
				<xs:enumeration value="video/mp4" />
				<xs:enumeration value="video/mkv" />
			</xs:restriction>
```

```
	</xs:simpleType>

	<xs:simpleType name="audio_type_option">
		<xs:restriction base="xs:string">
			<xs:enumeration value="audio/x-wav" />
			<xs:enumeration value="audio/basic" />
			<xs:enumeration value="audio/x-alaw-basic" />
			<xs:enumeration value="audio/L8" />
			<xs:enumeration value="audio/L16" />
			<xs:enumeration value="audio/x-aud" />
			<xs:enumeration value="audio/AMR" />
			<xs:enumeration value="audio/AMR-WB" />
			<xs:enumeration value="audio/3gpp" />
			<xs:enumeration value="audio/mp4" />
			<xs:enumeration value="audio/mkv" />
			<xs:enumeration value="text/uri-list" />
		</xs:restriction>
	</xs:simpleType>

	<xs:simpleType name="recording_video_type_option">
		<xs:restriction base="xs:string">
			<xs:enumeration value="video/x-vid" />
			<xs:enumeration value="video/3gpp" />
			<xs:enumeration value="image/jpeg" />
			<xs:enumeration value="video/mp4" />
			<xs:enumeration value="video/mkv" />
		</xs:restriction>
	</xs:simpleType>



	<xs:simpleType name="conf_party_mode">
		<xs:restriction base="xs:string">
			<xs:enumeration value="normal" />
			<xs:enumeration value="coach" />
			<xs:enumeration value="pupil" />
		</xs:restriction>
	</xs:simpleType>

	<xs:simpleType name="layout_size_option">
		<xs:restriction base="xs:string">
			<xs:enumeration value="automatic" />
			<xs:enumeration value="qcif" />
			<xs:enumeration value="cif" />
			<xs:enumeration value="vga" />
			<xs:enumeration value="720p" />
```

```
            </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="rtp_encryption_option">
            <xs:restriction base="xs:string">
                    <xs:enumeration value="none" />
                    <xs:enumeration value="dtls" />
                    <xs:enumeration value="srtp" />
            </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="time_value">
            <xs:restriction base="xs:string">
                    <xs:pattern value="(\+)?([0-9]*\.)?[0-9]+(ms|s)|infinite"/>
            </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="digit_value">
            <xs:restriction base="xs:string">
                    <xs:pattern value="[0-9#*]+|"/>
            </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="rfc2833_digit_value">
            <xs:restriction base="xs:string">
                    <xs:pattern value="[0-9#$*a-dA-D]+|"/>
            </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="audio_codec_amr_mode_value">
            <xs:restriction base="xs:string">
                    <xs:pattern value="[0-8]"/>
            </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="dtmf_tone_range">
            <xs:restriction base="xs:string">
                    <xs:pattern value="(0|(\-)([0-9]|1[0-9]|2[0-9]|3[0-9]|40))(dB|db|DB|Db)"/>
            </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="volume_range">
            <xs:restriction base="xs:string">
                    <xs:pattern value="((\+|)([0-9]|1[0-9]|2[0-9]|3[0-1])|(\-)([0-9]|1[0-
9]|2[0-9]|3[0-2]))((dB|db|DB|Db)|(dB|db|DB|Db)(;)(relative|absolute))"/>
            </xs:restriction>
```

```
    </xs:simpleType>

    <xs:simpleType name="integer_value">
            <xs:restriction base="xs:string">
                    <xs:pattern value="[0-9]+|infinite"/>
            </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="dtmf_mode_option">
            <xs:restriction base="xs:string">
                    <xs:enumeration value="inband" />
                    <xs:enumeration value="outofband" />
                    <xs:enumeration value="rfc2833" />
            </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="ack_mode_option">
            <xs:restriction base="xs:string">
                    <xs:enumeration value="automatic" />
                    <xs:enumeration value="manual" />
            </xs:restriction>
    </xs:simpleType>



    <xs:simpleType name="event_type">
            <xs:restriction base="xs:string">
                    <xs:enumeration value="end_play" />
                    <xs:enumeration value="end_record" />
                    <xs:enumeration value="end_playcollect" />
                    <xs:enumeration value="end_playrecord" />
                    <xs:enumeration value="end_overlay" />
                    <xs:enumeration value="end_dtmf" />
                    <xs:enumeration value="keepalive" />
                    <xs:enumeration value="incoming" />
                    <xs:enumeration value="ringing" />
                    <xs:enumeration value="connected" />
                    <xs:enumeration value="hangup" />
         <xs:enumeration value="info" />
                    <xs:enumeration value="dtmf" />
                    <xs:enumeration value="tone" />
                    <xs:enumeration value="any" />
                    <xs:enumeration value="end_speak" />
                    <xs:enumeration value="start_of_input" />
                    <xs:enumeration value="end_recognize" />
                    <xs:enumeration value="answered" />
```

```
                <xs:enumeration value="accepted" />
                <xs:enumeration value="updated" />
                <xs:enumeration value="active_talker" />
                <xs:enumeration value="alarm" />
                <xs:enumeration value="prack" />
                <xs:enumeration value="conf_overlay_expired" />
                <xs:enumeration value="message" />
                <xs:enumeration value="end_send_message" />
                <xs:enumeration value="stream" />
                <xs:enumeration value="speech_marker" />
                <xs:enumeration value="start_play" />
                <xs:enumeration value="end_event" />
                <xs:enumeration value="progress" />
                <xs:enumeration value="prack_ack" />
                <xs:enumeration value="status" />
                <xs:enumeration value="info_ack" />
        </xs:restriction>
</xs:simpleType>

<xs:simpleType name="event_data_name">
        <xs:restriction base="xs:string">
                <xs:enumeration value="tone" />
                <xs:enumeration value="digits" />
                <xs:enumeration value="info" />
                <xs:enumeration value="reason" />
                <xs:enumeration value="duration" />
                <xs:enumeration value="uri" />
                <xs:enumeration value="caller_uri" />
                <xs:enumeration value="content_type" />
                <xs:enumeration value="content" />
                <xs:enumeration value="transaction_id" />
        </xs:restriction>
</xs:simpleType>

<xs:simpleType name="event_resource_type">
        <xs:restriction base="xs:string">
                <xs:enumeration value="call" />
                <xs:enumeration value="conference" />
                <xs:enumeration value="mrcp" />
                <xs:enumeration value="any" />
        </xs:restriction>
</xs:simpleType>

<xs:element name="dvr_setting">
        <xs:complexType>
                <xs:attribute name="forward_key" type="digit_value" default="1" />
```

```
                <xs:attribute name="backward_key" type="digit_value" default="2" />
                <xs:attribute name="pause_key" type="digit_value" default="3" />
                <xs:attribute name="resume_key" type="digit_value" default="4" />
                <xs:attribute name="restart_key" type="digit_value" default="5" />
        </xs:complexType>
</xs:element>

<xs:simpleType name="dvr_action_option">
        <xs:restriction base="xs:string">
                <xs:enumeration value="forward" />
                <xs:enumeration value="backward" />
                <xs:enumeration value="pause" />
                <xs:enumeration value="resume" />
                <xs:enumeration value="restart" />
        </xs:restriction>
</xs:simpleType>

<xs:simpleType name="mrcp_action_option">
        <xs:restriction base="xs:string">
                <xs:enumeration value="stop-speak" />
                <xs:enumeration value="stop-recognize" />
                <xs:enumeration value="pause" />
                <xs:enumeration value="resume" />
                <xs:enumeration value="barge" />
                <xs:enumeration value="start-input-timers" />
        </xs:restriction>
</xs:simpleType>


<xs:element name="recording_audio_mime_params">
        <xs:complexType>
                        <xs:attribute name="codec" type="audio_codec_option" />
                        <xs:attribute name="mode" type="audio_codec_amr_mode_value" />
                        <xs:attribute name="rate" type="audio_rate_option" />
        </xs:complexType>
</xs:element>

<xs:element name="recording_video_mime_params">
        <xs:complexType>
                        <xs:attribute name="codec" type="video_codec_option" />
                        <xs:attribute name="profile" type="digit_value" />
                        <xs:attribute name="level" type="xs:string" />
                        <xs:attribute name="framerate" type="digit_value" />
                        <xs:attribute name="maxbitrate" type="digit_value" />
                        <xs:attribute name="height" type="digit_value" />
                        <xs:attribute name="width" type="digit_value" />
```

```
		</xs:complexType>
	</xs:element>

	<xs:element name="param">
		<xs:complexType>
				<xs:attribute name="name" type="xs:string" use="required"/>
				<xs:attribute name="value" type="xs:string"/>
		</xs:complexType>
	</xs:element>

	<xs:element name="content_element">
		<xs:complexType>
				<xs:attribute name="id" type="xs:string" use="required"/>
				<xs:attribute name="type" type="xs:string" use="required"/>
				<xs:attribute name="content" type="xs:string" use="required"/>
		</xs:complexType>
	</xs:element>



	<xs:element name="msg_payload">
		<xs:complexType>
				<xs:choice>
					<xs:element name="msg_payload_content">
						<xs:complexType>
								<xs:attribute name="content"
type="xs:string" use="required"/>
						</xs:complexType>
					</xs:element>
					<xs:element name="msg_payload_uri" >
						<xs:complexType>
								<xs:attribute name="uri"
type="xs:string" use="required"/>
						</xs:complexType>
					</xs:element>
				</xs:choice>
				<xs:attribute name="content_id" type="xs:string" use="required"/>
				<xs:attribute name="content_type" type="xs:string" use="required"/>
				<xs:attribute name="content_disposition" default="attachment">
					<xs:simpleType>
						<xs:restriction base="xs:string">
							<xs:enumeration value="attachment"/>
							<xs:enumeration value="render"/>
						</xs:restriction>
					</xs:simpleType>
				</xs:attribute>
				<xs:attribute name="uri" type="xs:string"/>
```

```
        </xs:complexType>
</xs:element>

<xs:element name="conf_participant">
        <xs:complexType>
                        <xs:attribute name="call_id" type="xs:string" />
                        <xs:attribute name="audio" type="media_direction" />
                        <xs:attribute name="video" type="media_direction" />
                        <xs:attribute name="caption" type="xs:string" />
                        <xs:attribute name="region" type="xs:string" />
        </xs:complexType>
</xs:element>

<xs:element name="get_last_action">
        <xs:complexType>
                <xs:choice minOccurs="0" maxOccurs="1">
                        <xs:element ref="play" />
                        <xs:element ref="record" />
                        <xs:element ref="update_play" />
                        <xs:element ref="playcollect" />
                        <xs:element ref="playrecord" />
                        <xs:element ref="overlay" />
                        <xs:element ref="stop" />
                        <xs:element ref="join" />
                        <xs:element ref="unjoin" />
                        <xs:element ref="add_party" />
                        <xs:element ref="update_party" />
                        <xs:element ref="remove_party" />
                        <xs:element ref="send_dtmf" />
                        <xs:element ref="send_info" />
                        <xs:element ref="send_info_ack" />
                        <xs:element ref="transfer" />
                        <xs:element ref="redirect" />
                        <xs:element ref="hangup" />
                        <xs:element ref="get_call_info" />
                        <xs:element ref="add_ice_candidate" />
                        <xs:element ref="send_message" />
                        <xs:element ref="send_hangup_ack" />
                        <xs:element ref="get_last_event" />
                        <xs:element ref="speak" />
                        <xs:element ref="recognize" />
                        <xs:element ref="mrcp-update-action" />
                        <xs:element ref="set-asr-param" />
                        <xs:element ref="get-asr-param" />
                        <xs:element ref="set-tts-param" />
                        <xs:element ref="get-tts-param" />
```

```
                              <xs:element ref="define-grammar" />
                              <xs:element ref="send_event" />
                              <xs:element ref="dial" />
                              <xs:element ref="send_prack" />
                              <xs:element ref="send_prack_ack" />
                              <xs:element ref="send_answer_ack" />
                       </xs:choice>
                </xs:complexType>
        </xs:element>

        <xs:element name="get_last_event">
                <xs:complexType>
                              <xs:all minOccurs="0" maxOccurs="1">
                                     <xs:element ref="event" />
                              </xs:all>
                </xs:complexType>
        </xs:element>

        <xs:element name="add_party">
                <xs:complexType>
                              <xs:attribute name="conf_id" type="xs:string" use="required" />
                              <xs:attribute name="audio" type="media_direction"
default="recvonly" />
                              <xs:attribute name="video" type="media_direction"
default="recvonly" />
                              <xs:attribute name="caption" type="xs:string" />
                              <xs:attribute name="caption_duration" type="time_value"  />
                              <xs:attribute name="clamp_dtmf" type="boolean_type"  />
                              <xs:attribute name="auto_gain_control" type="boolean_type"  />
                              <xs:attribute name="echo_cancellation" type="boolean_type"  />
                              <xs:attribute name="mute" type="boolean_type" default="no" />
                              <xs:attribute name="tx_mute" type="boolean_type" default="no" />
                              <xs:attribute name="privilege" type="boolean_type" default="no" />
                              <xs:attribute name="mode" type="conf_party_mode" default="normal"
/>
                              <xs:attribute name="region" type="xs:string" default="0"/>
                </xs:complexType>
        </xs:element>

        <xs:element name="update_party">
                <xs:complexType>
                              <xs:attribute name="conf_id" type="xs:string" />
                              <xs:attribute name="audio" type="media_direction" />
                              <xs:attribute name="video" type="media_direction" />
                              <xs:attribute name="caption" type="xs:string" />
                              <xs:attribute name="caption_duration" type="time_value"  />
                              <xs:attribute name="clamp_dtmf" type="boolean_type" />
```

```
                              <xs:attribute name="auto_gain_control" type="boolean_type"/>
                              <xs:attribute name="echo_cancellation" type="boolean_type" />
                              <xs:attribute name="mute" type="boolean_type"  />
                              <xs:attribute name="tx_mute" type="boolean_type"  />
                              <xs:attribute name="privilege" type="boolean_type"  />
                              <xs:attribute name="mode" type="conf_party_mode"  />
                              <xs:attribute name="region" type="xs:string" />
               </xs:complexType>
        </xs:element>

        <xs:element name="remove_party">
               <xs:complexType>
                              <xs:attribute name="conf_id" type="xs:string" />
               </xs:complexType>
        </xs:element>

        <xs:element name="join">
               <xs:complexType>
                              <xs:attribute name="call_id" type="xs:string" use="required" />
                              <xs:attribute name="audio" type="media_direction"
default="sendrecv"/>
                              <xs:attribute name="video" type="media_direction"
default="sendrecv"/>
                              <xs:attribute name="audio_transcode" type="boolean_type"
default="yes"/>
                              <xs:attribute name="video_transcode" type="boolean_type"
default="yes"/>
                              <xs:attribute name="media_path_optimize" type="boolean_type"
default="no"/>
               </xs:complexType>
        </xs:element>

        <xs:element name="unjoin">
               <xs:complexType>
                              <xs:attribute name="call_id" type="xs:string" />
                              <xs:attribute name="audio" type="media_direction"
default="sendrecv"/>
                              <xs:attribute name="video" type="media_direction"
default="sendrecv"/>
               </xs:complexType>
        </xs:element>

        <xs:element name="transfer">
               <xs:complexType>
                              <xs:attribute name="call_id" type="xs:string" />
                              <xs:attribute name="uri" type="xs:string" />
               </xs:complexType>
        </xs:element>
```

```
<xs:element name="redirect">
        <xs:complexType>
                        <xs:attribute name="uri" type="xs:string" use="required"/>
        </xs:complexType>
</xs:element>

<xs:element name="hangup">
        <xs:complexType>
                        <xs:attribute name="content_type" type="xs:string" />
                        <xs:attribute name="content" type="xs:string" />
                        <xs:attribute name="reason" type="xs:string" />
        </xs:complexType>
</xs:element>

<xs:element name="dial">
        <xs:complexType>
                        <xs:all minOccurs="0">
                                <xs:element ref="sip_headers" />
                        </xs:all>
                        <xs:attribute name="uri" type="xs:string" use="required"/>
                        <xs:attribute name="called_uri" type="xs:string" />
                        <xs:attribute name="caller_uri" type="xs:string" />
                        <xs:attribute name="caller_display_name" type="xs:string" />
                        <xs:attribute name="cpa" type="boolean_type" />
                        <xs:attribute name="timeout" type="time_value"/>
                        <xs:attribute name="content_type" type="xs:string" />
                        <xs:attribute name="content" type="xs:string" />
                        <xs:attribute name="answer_ack_mode"  type="ack_mode_option" />
                        <xs:attribute name="prack_mode"  type="ack_mode_option" />
                        <xs:attribute name="prack_level">
                                <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                                <xs:enumeration value="supported"/>
                                                <xs:enumeration value="required"/>
                                        </xs:restriction>
                                </xs:simpleType>
                        </xs:attribute>
        </xs:complexType>
</xs:element>



<xs:element name="send_info">
        <xs:complexType>
                        <xs:sequence>
```

```
                                        <xs:element ref="content_element" minOccurs="0"
maxOccurs="unbounded"/>
                                </xs:sequence>
                                <xs:attribute name="content_type" type="xs:string" />
                                <xs:attribute name="content" type="xs:string"/>
                                <xs:attribute name="enable_info_ack" type="boolean_type"
default="no"/>
                </xs:complexType>
        </xs:element>

        <xs:element name="send_hangup_ack">
                <xs:complexType>
                                <xs:attribute name="content_type" type="xs:string" />
                                <xs:attribute name="content" type="xs:string" />
                </xs:complexType>
        </xs:element>

        <xs:element name="send_dtmf">
                <xs:complexType>
                                <xs:attribute name="digits" type="rfc2833_digit_value"
use="required"/>
                                <xs:attribute name="duration" type="time_value" default="100ms"/>
                                <xs:attribute name="interval" type="time_value" default="100ms"/>
                                <xs:attribute name="level" type="dtmf_tone_range" default="-10dB"/>
                                <xs:attribute name="transaction_id" type="xs:string" />
                </xs:complexType>
        </xs:element>

        <xs:element name="get_call_info">
                <xs:complexType>
                                <xs:all minOccurs="0">
                                        <xs:element ref="sip_headers" />
                                </xs:all>
                                <xs:attribute name="local_sdp" type="xs:string" />
                                <xs:attribute name="remote_sdp" type="xs:string" />
                                <xs:attribute name="media" type="media_type"/>
                                <xs:attribute name="audio" type="media_direction" />
                                <xs:attribute name="video" type="media_direction" />
                                <xs:attribute name="uri" type="xs:string" />
                                <xs:attribute name="caller_uri" type="xs:string" />
                                <xs:attribute name="called_uri" type="xs:string" />
                                <xs:attribute name="application_id" type="xs:string" />
                                <xs:attribute name="local_seqno" type="xs:string" />
                                <xs:attribute name="remote_seqno" type="xs:string" />
                </xs:complexType>
        </xs:element>
```

```
	<xs:element name="add_ice_candidate">
		<xs:complexType>
				<xs:attribute name="mline" type="xs:string" use="required"/>
				<xs:attribute name="mid" type="xs:string" use="required"/>
				<xs:attribute name="candidate" type="xs:string" use="required"/>
		</xs:complexType>
	</xs:element>

	<xs:element name="send_message">
		<xs:complexType>
				<xs:sequence>
					<xs:choice minOccurs="0" maxOccurs="unbounded">
							<xs:element ref="content_element" />
							<xs:element ref="msg_payload" />
							<xs:element ref="sip_headers" minOccurs="0"
maxOccurs="1" />
					</xs:choice>
				</xs:sequence>
				<xs:attribute name="caller_uri" type="xs:string" />
				<xs:attribute name="called_uri" type="xs:string" />
				<xs:attribute name="uri" type="xs:string" />
				<xs:attribute name="content_type" type="xs:string" />
				<xs:attribute name="content" type="xs:string" />
				<xs:attribute name="mode" default="signalling">
					<xs:simpleType>
						<xs:restriction base="xs:string">
							<xs:enumeration value="signalling"/>
							<xs:enumeration value="msrp"/>
							<xs:enumeration value="rfc5547"/>
						</xs:restriction>
					</xs:simpleType>
				</xs:attribute>
				<xs:attribute name="report" >
					<xs:simpleType>
						<xs:restriction base="xs:string">
							<xs:enumeration value="success"/>
							<xs:enumeration value="failure"/>
							<xs:enumeration value="both"/>
						</xs:restriction>
					</xs:simpleType>
				</xs:attribute>
				<xs:attribute name="msg_multipart_type" type="xs:string" />

				<xs:attribute name="transaction_id" type="xs:string" />
		</xs:complexType>
	</xs:element>
```

```
	<xs:element name="send_info_ack">
		<xs:complexType>
				<xs:sequence>
					<xs:element ref="content_element" minOccurs="0"
maxOccurs="unbounded"/>
				</xs:sequence>
				<xs:attribute name="content_type" type="xs:string" />
				<xs:attribute name="content" type="xs:string" />
		</xs:complexType>
	</xs:element>



	<xs:element name="send_event">
		<xs:complexType>
				<xs:attribute name="events" type="rfc2833_digit_value"
use="required"/>
				<xs:attribute name="duration" type="time_value" default="100ms"/>
				<xs:attribute name="interval" type="time_value" default="100ms"/>
				<xs:attribute name="level" type="dtmf_tone_range" default="-10dB"/>
				<xs:attribute name="transaction_id" type="xs:string" />
		</xs:complexType>
	</xs:element>

	<xs:element name="send_prack">
		<xs:complexType>
		</xs:complexType>
	</xs:element>
	<xs:element name="send_prack_ack">
		<xs:complexType>
		</xs:complexType>
	</xs:element>
	<xs:element name="send_answer_ack">
		<xs:complexType>
		</xs:complexType>
	</xs:element>

	<xs:element name="call_action">
		<xs:complexType>
			<xs:choice minOccurs="1" maxOccurs="1">
				<xs:element ref="play" />
				<xs:element ref="record" />
				<xs:element ref="update_play" />
				<xs:element ref="playcollect" />
				<xs:element ref="playrecord" />
				<xs:element ref="overlay" />
				<xs:element ref="stop" />
```

```
                        <xs:element ref="join" />
                        <xs:element ref="unjoin" />
                        <xs:element ref="add_party" />
                        <xs:element ref="update_party" />
                        <xs:element ref="remove_party" />
                        <xs:element ref="send_dtmf" />
                        <xs:element ref="send_info" />
                        <xs:element ref="send_info_ack" />
                        <xs:element ref="transfer" />
                        <xs:element ref="redirect" />
                        <xs:element ref="hangup" />
                        <xs:element ref="get_call_info" />
                        <xs:element ref="add_ice_candidate" />
                        <xs:element ref="send_message" />
                        <xs:element ref="send_hangup_ack" />
                        <xs:element ref="get_last_action" />
                        <xs:element ref="get_last_event" />
                        <xs:element ref="send_event" />
                        <xs:element ref="dial" />
                        <xs:element ref="send_prack" />
                        <xs:element ref="send_prack_ack" />
                        <xs:element ref="send_answer_ack" />
                </xs:choice>
        </xs:complexType>
</xs:element>

<xs:element name="conf_action">
        <xs:complexType>
                <xs:choice minOccurs="1" maxOccurs="1">
                        <xs:element ref="play" />
                        <xs:element ref="record" />
                        <xs:element ref="update_play" />
                        <xs:element ref="stop" />
                        <xs:element ref="get_last_action" />
                        <xs:element ref="get_last_event" />
                        <xs:element ref="send_dtmf" />
                        <xs:element ref="playcollect" />
                        <xs:element ref="playrecord" />
                        <xs:element ref="send_event" />
                </xs:choice>
        </xs:complexType>
</xs:element>

<xs:element name="speak">
        <xs:complexType>
                        <xs:sequence>
```

```
                                    <xs:element ref="param" minOccurs="0" maxOccurs="unbounded"
/>
                              </xs:sequence>
                              <xs:attribute name="call_id" type="xs:string" use="required" />
                              <xs:attribute name="barge" type="boolean_type" default="yes" />
                              <xs:attribute name="locale" type="xs:string" default="en-US" />
                              <xs:attribute name="content" type="xs:string" use="required" />
                              <xs:attribute name="content_type" type="xs:string" use="required"
/>
                              <xs:attribute name="transaction_id" type="xs:string" />
               </xs:complexType>
        </xs:element>

        <xs:element name="recognize">
               <xs:complexType>
                              <xs:sequence>
                                    <xs:element ref="param" minOccurs="0" maxOccurs="unbounded"
/>
                              </xs:sequence>
                              <xs:attribute name="call_id" type="xs:string" use="required" />
                              <xs:attribute name="grammar" type="xs:string" />
                              <xs:attribute name="grammar_type" type="xs:string" />
                              <xs:attribute name="grammar_id" type="xs:string" />
                              <xs:attribute name="timeout" type="time_value" default="infinite"
/>
                              <xs:attribute name="transaction_id" type="xs:string" />
               </xs:complexType>
        </xs:element>

        <xs:element name="set-asr-param">
               <xs:complexType>
                              <xs:sequence>
                                    <xs:element ref="param" minOccurs="1" maxOccurs="unbounded"
/>
                              </xs:sequence>
               </xs:complexType>
        </xs:element>

        <xs:element name="get-asr-param">
               <xs:complexType>
                              <xs:sequence>
                                    <xs:element ref="param" minOccurs="0" maxOccurs="unbounded"
/>
                              </xs:sequence>
               </xs:complexType>
        </xs:element>

        <xs:element name="set-tts-param">
```

```
                <xs:complexType>
                                <xs:sequence>
                                        <xs:element ref="param" minOccurs="1" maxOccurs="unbounded"
/>
                                </xs:sequence>
                </xs:complexType>
        </xs:element>

        <xs:element name="define-grammar">
                <xs:complexType>
                                <xs:attribute name="grammar" type="xs:string" use="required"/>
                                <xs:attribute name="grammar_type" type="xs:string" use="required"/>
                                <xs:attribute name="grammar_id" type="xs:string" use="required"/>
                </xs:complexType>
        </xs:element>

        <xs:element name="get-tts-param">
                <xs:complexType>
                                <xs:sequence>
                                        <xs:element ref="param" minOccurs="0" maxOccurs="unbounded"
/>
                                </xs:sequence>
                </xs:complexType>
        </xs:element>

        <xs:element name="mrcp-update-action">
                <xs:complexType>
                                <xs:attribute name="action" type="mrcp_action_option"
use="required"/>
                </xs:complexType>
        </xs:element>

        <xs:element name="mrcp_action">
                <xs:complexType>
                        <xs:choice minOccurs="1" maxOccurs="1">
                                <xs:element ref="speak" />
                                <xs:element ref="recognize" />
                                <xs:element ref="mrcp-update-action" />
                                <xs:element ref="set-asr-param" />
                                <xs:element ref="get-asr-param" />
                                <xs:element ref="set-tts-param" />
                                <xs:element ref="get-tts-param" />
                                <xs:element ref="define-grammar" />
                                <xs:element ref="get_last_action" />
                                <xs:element ref="get_last_event" />
                        </xs:choice>
                </xs:complexType>
```

```
	</xs:element>

	<xs:attributeGroup name="response_attrgroup">
		<xs:attribute name="href" type="xs:string" use="required" />
		<xs:attribute name="identifier" type="xs:string" use="required" />
		<xs:attribute name="appid" type="xs:string" use="required" />
	</xs:attributeGroup>

	<xs:element name="overlay">
		<xs:complexType>
			<xs:attribute name="uri" type="xs:string" use="required"/>
			<xs:attribute name="duration" type="time_value" default="infinite" />
			<xs:attribute name="direction"  default="send">
				<xs:simpleType>
					<xs:restriction base="xs:string">
						<xs:enumeration value="send"/>
						<xs:enumeration value="recv"/>
					</xs:restriction>
				</xs:simpleType>
			</xs:attribute>
			<xs:attribute name="transaction_id" type="xs:string" />
		</xs:complexType>
	</xs:element>

	<xs:element name="event_data">
		<xs:complexType>
			<xs:attribute name="name" type="xs:string" use="required" />
			<xs:attribute name="value" type="xs:string" use="required" />
		</xs:complexType>
	</xs:element>


	<xs:element name="sip_headers">
		<xs:complexType>
			<xs:sequence>
				<xs:element ref="param" minOccurs="0" maxOccurs="unbounded" />
			</xs:sequence>
			<xs:attribute name="raw_sip_headers" type="xs:string"  />
		</xs:complexType>
	</xs:element>

	<xs:element name="event">
		<xs:complexType>
			<xs:sequence>
				<xs:element ref="event_data" minOccurs="0" maxOccurs="unbounded" />
			</xs:sequence>
```

```
                    <xs:attribute name="type" type="event_type" use="required" />
                    <xs:attribute name="resource_type" type="xs:string" />
                    <xs:attribute name="resource_id" type="xs:string" />
            </xs:complexType>
    </xs:element>

    <xs:element name="stop">
            <xs:complexType>
                    <xs:attribute name="transaction_id" type="xs:string" use="required"/>
            </xs:complexType>
    </xs:element>

    <xs:element name="play_source">
            <xs:complexType>
                    <xs:attribute name="location" type="xs:string" />
                    <!-- Will be deprecated after xms 3.0  xs:attribute name="base_audio_uri"
type="xs:string" using xs:attribute name="audio_base_uri" type="xs:string" instead/-->
                    <xs:attribute name="base_audio_uri" type="xs:string" />
                    <xs:attribute name="audio_base_uri" type="xs:string" />
                    <xs:attribute name="audio_uri" type="xs:string" />
                    <xs:attribute name="audio_type" type="audio_type_option" />
                    <xs:attribute name="audio_rate" type="audio_rate_option" />
                    <!-- Will be deprecated after xms 3.0  xs:attribute name="base_video_uri"
type="xs:string" using xs:attribute name="video_base_uri" type="xs:string" instead/-->
                    <xs:attribute name="base_video_uri" type="xs:string" />
                    <xs:attribute name="video_base_uri" type="xs:string" />
                    <xs:attribute name="video_uri" type="xs:string" />
                    <xs:attribute name="video_type" type="video_type_option" />
            </xs:complexType>
    </xs:element>

    <xs:element name="play">
            <xs:complexType>
                    <xs:sequence>
                            <xs:element ref="play_source" minOccurs="1" maxOccurs="1" />
                            <xs:element ref="dvr_setting" minOccurs="0" maxOccurs="1" />
                    </xs:sequence>
                    <xs:attribute name="offset" type="time_value" default="0s" />
                    <xs:attribute name="repeat" type="integer_value" default="0" />
                    <xs:attribute name="delay" type="time_value" default="1s" />
                    <xs:attribute name="skip_interval" type="time_value" default="1s" />
                    <xs:attribute name="max_time" type="time_value" default="infinite" />
                    <xs:attribute name="terminate_digits" type="digit_value" default="#"/>
                    <xs:attribute name="region" type="xs:string" />
                    <xs:attribute name="transaction_id" type="xs:string" />
                    <xs:attribute name="no_cache" type="boolean_type" />
                    <xs:attribute name="max_age" type="time_value" />
```

```
                    <xs:attribute name="max_stale" type="time_value" />
                    <xs:attribute name="fetch_timeout" type="time_value" default="300s" />
            </xs:complexType>
        </xs:element>

        <xs:element name="update_play">
            <xs:complexType>
                    <xs:attribute name="dvr_action" type="dvr_action_option" />
                    <xs:attribute name="region" type="xs:string" />
                    <xs:attribute name="transaction_id" type="xs:string" use="required"/>
            </xs:complexType>
        </xs:element>

        <xs:element name="record">
            <xs:complexType>
                    <xs:all>
                            <xs:element ref="recording_audio_mime_params" minOccurs="0"
maxOccurs="1" />
                            <xs:element ref="recording_video_mime_params" minOccurs="0"
maxOccurs="1" />
                    </xs:all>
                    <xs:attribute name="terminate_digits" type="digit_value" default="#"/>
            <xs:attribute name="recording_uri" type="xs:string" />
            <xs:attribute name="recording_audio_uri" type="xs:string" />
                    <xs:attribute name="recording_audio_type" type="audio_type_option" />
                    <xs:attribute name="recording_video_uri" type="xs:string" />
                    <xs:attribute name="recording_video_type"
type="recording_video_type_option" default="video/x-vid"/>
                    <xs:attribute name="max_silence" type="time_value" default="infinite"  />
                    <xs:attribute name="max_time" type="time_value" default="infinite" />
                    <xs:attribute name="noinput_timeout" type="time_value" default="infinite"
/>
                    <xs:attribute name="transaction_id" type="xs:string" />
            </xs:complexType>
        </xs:element>

        <xs:element name="playrecord">
            <xs:complexType>
            <xs:all>
                            <xs:element ref="play_source" minOccurs="0" maxOccurs="1" />
                            <xs:element ref="recording_audio_mime_params" minOccurs="0"
maxOccurs="1" />
                            <xs:element ref="recording_video_mime_params" minOccurs="0"
maxOccurs="1" />
                    </xs:all>
                    <xs:attribute name="barge" type="boolean_type" default="yes" />
                    <xs:attribute name="cleardigits" type="boolean_type" default="no" />
                    <xs:attribute name="offset" type="time_value" default="0s" />
```

```
                        <xs:attribute name="repeat" type="integer_value" default="0" />
                        <xs:attribute name="delay" type="time_value" default="1s" />
                        <xs:attribute name="recording_uri" type="xs:string" />
                        <xs:attribute name="recording_audio_uri" type="xs:string" />
                        <xs:attribute name="recording_audio_type" type="audio_type_option" />
                        <xs:attribute name="recording_video_uri" type="xs:string" />
                        <xs:attribute name="recording_video_type"
type="recording_video_type_option" default="video/x-vid"/>
                        <xs:attribute name="beep" type="boolean_type" default="yes" />
                        <xs:attribute name="terminate_digits" type="digit_value" default="#"/>
                        <xs:attribute name="max_time" type="time_value" default="infinite" />
                        <xs:attribute name="max_silence" type="time_value" default="infinite"  />
                        <xs:attribute name="noinput_timeout" type="time_value" default="infinite"
/>
                        <xs:attribute name="transaction_id" type="xs:string" />
                        <xs:attribute name="no_cache" type="boolean_type" />
                        <xs:attribute name="max_age" type="time_value" />
                        <xs:attribute name="max_stale" type="time_value" />
                        <xs:attribute name="fetch_timeout" type="time_value" default="300s" />
                </xs:complexType>
        </xs:element>

        <xs:element name="playcollect">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element ref="play_source" minOccurs="0" maxOccurs="1" />
                        </xs:sequence>
                        <xs:attribute name="barge" type="boolean_type" default="yes" />
                        <xs:attribute name="cleardigits" type="boolean_type" default="no" />
                        <xs:attribute name="offset" type="time_value" default="0s" />
                        <xs:attribute name="repeat" type="integer_value" default="0" />
                        <xs:attribute name="delay" type="time_value" default="1s" />
                        <xs:attribute name="max_digits" type="xs:string" />
                        <xs:attribute name="timeout" type="time_value" />
                        <xs:attribute name="interdigit_timeout" type="time_value" />
                        <xs:attribute name="terminate_digits" type="digit_value" default="#"/>
                        <xs:attribute name="tone_detection" type="boolean_type" default="no" />
                        <xs:attribute name="transaction_id" type="xs:string" />
                        <xs:attribute name="no_cache" type="boolean_type" />
                        <xs:attribute name="max_age" type="time_value" />
                        <xs:attribute name="max_stale" type="time_value" />
                        <xs:attribute name="fetch_timeout" type="time_value" default="300s" />
                </xs:complexType>
        </xs:element>

        <xs:element name="error">
                <xs:complexType>
```

```
                        <xs:attribute name="code" type="xs:string" use="required" />
                        <xs:attribute name="description" type="xs:string" use="required" />
                </xs:complexType>
        </xs:element>

        <xs:element name="call">
                <xs:complexType>
                        <xs:sequence>
                                <xs:choice minOccurs="0" maxOccurs="unbounded">
                                        <xs:element ref="call_action" minOccurs="0" maxOccurs="1"
/>
                                        <xs:element ref="sip_headers" minOccurs="0" maxOccurs="1"
/>
                                </xs:choice>
                        </xs:sequence>

             <xs:attribute name="answer" type="boolean_type"/>
                        <xs:attribute name="signaling" type="boolean_type" default="yes" />
                        <xs:attribute name="media" type="media_type" />
                        <xs:attribute name="source_uri" type="xs:string" />
                        <xs:attribute name="destination_uri" type="xs:string" />
                        <xs:attribute name="called_uri" type="xs:string" />
                        <xs:attribute name="display_name" type="xs:string"/>
                        <xs:attribute name="sdp" type="xs:string"/>
                        <xs:attribute name="cpa" type="boolean_type" default="no"/>
                        <xs:attribute name="dtmf_mode" type="dtmf_mode_option" />
                        <xs:attribute name="async_dtmf" type="boolean_type" />
                        <xs:attribute name="async_tone" type="boolean_type" />
                        <xs:attribute name="rx_delta" type="volume_range" />
                        <xs:attribute name="tx_delta" type="volume_range" />
                        <xs:attribute name="cleardigits" type="boolean_type" />
                        <xs:attribute name="info_ack_mode" type="ack_mode_option" />
                        <xs:attribute name="hangup_ack_mode" type="ack_mode_option" />
                        <xs:attribute name="early_media" type="boolean_type" />
                        <xs:attribute name="audio" type="media_direction" />
                        <xs:attribute name="video" type="media_direction" />
                        <xs:attribute name="message" type="media_direction" />
                        <xs:attribute name="image" type="media_direction" />
                        <xs:attribute name="accept" type="boolean_type" />
                        <xs:attribute name="async_completion" type="boolean_type" />
                        <xs:attribute name="dial_timeout" type="time_value"/>
                        <xs:attribute name="encryption" type="rtp_encryption_option" />
                        <xs:attribute name="ice" type="boolean_type"/>
                        <xs:attribute name="setup" >
                                <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                                <xs:enumeration value="active"/>
```

```
                                        <xs:enumeration value="passive"/>
                                </xs:restriction>
                        </xs:simpleType>
                </xs:attribute>
                <xs:attribute name="content" type="xs:string" />
                <xs:attribute name="content_type" type="xs:string" />
                <xs:attribute name="local_rtp_address" type="xs:string" />
                <xs:attribute name="answer_ack_mode"  type="ack_mode_option" />
                <xs:attribute name="prack_mode"  type="ack_mode_option" />
                <xs:attribute name="prack_ack_mode"  type="ack_mode_option" />
                <xs:attribute name="prack_level">
                        <xs:simpleType>
                                <xs:restriction base="xs:string">
                                        <xs:enumeration value="none"/>
                                        <xs:enumeration value="required"/>
                                        <xs:enumeration value="supported"/>
                                </xs:restriction>
                        </xs:simpleType>
                </xs:attribute>
        </xs:complexType>
</xs:element>

<xs:element name="call_response">
        <xs:complexType>
                <xs:sequence>
                        <xs:choice minOccurs="0" maxOccurs="unbounded">
                                <xs:element ref="call_action" minOccurs="0" maxOccurs="1"
/>
                                <xs:element ref="sip_headers" minOccurs="0" maxOccurs="1"
/>
                        </xs:choice>
                </xs:sequence>
                <xs:attribute name="signaling" type="boolean_type" />
                <xs:attribute name="media" type="media_type" />
     <xs:attribute name="destination_uri" type="xs:string" />
     <xs:attribute name="display_name" type="xs:string"/>
     <xs:attribute name="source_uri" type="xs:string" />
     <xs:attribute name="called_uri" type="xs:string" />
                <xs:attribute name="call_type" type="call_type_option" />
                <xs:attribute name="connected" type="boolean_type" />
     <xs:attribute name="sdp" type="xs:string"/>
                <xs:attribute name="cpa" type="boolean_type" />
                <xs:attribute name="dtmf_mode" type="dtmf_mode_option" />
                <xs:attribute name="async_dtmf" type="boolean_type"/>
                <xs:attribute name="async_tone" type="boolean_type" />
                <xs:attribute name="rx_delta" type="volume_range" />
                <xs:attribute name="tx_delta" type="volume_range" />
```

```
                    <xs:attribute name="cleardigits" type="boolean_type" />
                    <xs:attribute name="info_ack_mode" type="ack_mode_option" />
                    <xs:attribute name="hangup_ack_mode" type="ack_mode_option" />
                    <xs:attribute name="early_media" type="boolean_type" />
                    <xs:attribute name="audio" type="media_direction" />
                    <xs:attribute name="video" type="media_direction" />
                    <xs:attribute name="message" type="media_direction" />
                    <xs:attribute name="image" type="media_direction" />
                    <xs:attribute name="async_completion" type="boolean_type" />
                    <xs:attribute name="encryption" type="rtp_encryption_option" />
                    <xs:attribute name="ice" type="boolean_type"/>
                    <xs:attribute name="setup" >
                        <xs:simpleType>
                            <xs:restriction base="xs:string">
                                <xs:enumeration value="active"/>
                                <xs:enumeration value="passive"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:attribute>
                    <xs:attribute name="content" type="xs:string" />
                    <xs:attribute name="content_type" type="xs:string" />
                    <xs:attributeGroup ref="response_attrgroup" />
                    <xs:attribute name="local_rtp_address" type="xs:string" />
                    <xs:attribute name="answer_ack_mode"  type="ack_mode_option" />
                    <xs:attribute name="prack_mode"  type="ack_mode_option" />
                    <xs:attribute name="prack_ack_mode"  type="ack_mode_option" />
                    <xs:attribute name="prack_level">
                        <xs:simpleType>
                            <xs:restriction base="xs:string">
                                <xs:enumeration value="none"/>
                                <xs:enumeration value="required"/>
                                <xs:enumeration value="supported"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:attribute>
                </xs:complexType>
            </xs:element>

            <xs:element name="calls_response">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element ref="call_response" minOccurs="0"
maxOccurs="unbounded"/>
                    </xs:sequence>
                    <xs:attribute name="size" type="xs:string" use="required" />
                </xs:complexType>
```

```
	</xs:element>

	<xs:element name="eventhandler">
		<xs:complexType>
			<xs:sequence>
				<xs:element ref="eventsubscribe" minOccurs="1"
maxOccurs="unbounded" />
			</xs:sequence>
		</xs:complexType>
	</xs:element>

	<xs:element name="eventhandler_response">
		<xs:complexType>
			<xs:sequence>
				<xs:element ref="eventsubscribe" minOccurs="1"
maxOccurs="unbounded" />
			</xs:sequence>
			<xs:attributeGroup ref="response_attrgroup" />
		</xs:complexType>
	</xs:element>

	<xs:element name="eventhandlers_response">
		<xs:complexType>
			<xs:sequence>
				<xs:element ref="eventhandler_response" minOccurs="0"
maxOccurs="unbounded" />
			</xs:sequence>
			<xs:attribute name="size" type="xs:string" use="required" />
		</xs:complexType>
	</xs:element>

	<xs:element name="eventsubscribe">
		<xs:complexType>
			<xs:attribute name="type" type="event_type" default="any" />
			<xs:attribute name="action" type="action_option" default="add" />
			<xs:attribute name="resource_id" type="xs:string" default="any" />
			<xs:attribute name="resource_type" type="event_resource_type"
default="any" />
		</xs:complexType>
	</xs:element>

	<xs:element name="conference">
		<xs:complexType>
			<xs:sequence>
				<xs:element ref="conf_action" minOccurs="0" maxOccurs="1" />
			</xs:sequence>
			<xs:attribute name="type" type="media_type" default="audio" />
```

```
			<xs:attribute name="max_parties" type="xs:string" default="9" />
			<xs:attribute name="reserve" type="xs:string" default="0" />
			<xs:attribute name="layout" type="xs:string" />
			<xs:attribute name="layout_regions" type="xs:string" />
			<xs:attribute name="layout_size" type="layout_size_option" />
			<xs:attribute name="caption" type="boolean_type" default="yes" />
			<xs:attribute name="caption_duration" type="time_value" default="20s" />
			<xs:attribute name="beep" type="boolean_type" default="yes" />
			<xs:attribute name="clamp_dtmf" type="boolean_type" default="yes" />
			<xs:attribute name="auto_gain_control" type="boolean_type" default="yes"
/>
			<xs:attribute name="echo_cancellation" type="boolean_type" default="yes"
/>
			<xs:attribute name="active_talker_region" type="xs:string" />
			<xs:attribute name="active_talker_interval">
				<xs:simpleType>
					<xs:restriction base="xs:string">
						<xs:pattern value="(\+)?([0-9]*\.)?[0-
9]+(ms|s)|infinite|0"/>
					</xs:restriction>
				</xs:simpleType>
			</xs:attribute>
			<xs:attribute name="max_active_talkers">
				<xs:simpleType>
					<xs:restriction base="xs:string">
						<xs:pattern value="[2-9]|10"/>
					</xs:restriction>
				</xs:simpleType>
			</xs:attribute>
			<xs:attribute name="region_overlays" type="xs:string" />
		</xs:complexType>
	</xs:element>

	<xs:element name="conference_response">
		<xs:complexType>
			<xs:sequence>
				<xs:element ref="conf_action" minOccurs="0" maxOccurs="1" />
				<xs:element ref="conf_participant" minOccurs="0"
maxOccurs="unbounded"/>
			</xs:sequence>
			<xs:attribute name="type" type="media_type" />
			<xs:attribute name="max_parties" type="xs:string"  />
			<xs:attribute name="reserve" type="xs:string" />
			<xs:attribute name="layout" type="xs:string"  />
			<xs:attribute name="layout_regions" type="xs:string" />
			<xs:attribute name="layout_size" type="layout_size_option" />
			<xs:attribute name="caption" type="boolean_type" />
```

```
                        <xs:attribute name="caption_duration" type="time_value"  />
                        <xs:attribute name="beep" type="xs:string" default="yes" />
                        <xs:attribute name="clamp_dtmf" type="boolean_type"/>
                        <xs:attribute name="auto_gain_control" type="boolean_type"/>
                        <xs:attribute name="echo_cancellation" type="boolean_type"/>
                        <xs:attribute name="active_talker_region" type="xs:string" />
                        <xs:attribute name="active_talker_interval">
                                <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                                <xs:pattern value="(\+)?([0-9]*\.)?[0-
9]+(ms|s)|infinite|0"/>
                                        </xs:restriction>
                                </xs:simpleType>
                        </xs:attribute>
                        <xs:attribute name="max_active_talkers">
                                <xs:simpleType>
                                        <xs:restriction base="xs:string">
                                                <xs:pattern value="[2-9]|10"/>
                                        </xs:restriction>
                                </xs:simpleType>
                        </xs:attribute>
                        <xs:attribute name="region_overlays" type="xs:string" />
                        <xs:attributeGroup ref="response_attrgroup" />
                </xs:complexType>
        </xs:element>

        <xs:element name="conferences_response">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element ref="conference_response" minOccurs="0"
maxOccurs="unbounded"/>
                        </xs:sequence>
                        <xs:attribute name="size" type="xs:string" use="required" />
                </xs:complexType>
        </xs:element>

        <xs:element name="mrcp">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element ref="mrcp_action" minOccurs="0" maxOccurs="1" />
                        </xs:sequence>
                        <xs:attribute name="asr" type="boolean_type" default="yes" />
                        <xs:attribute name="tts" type="boolean_type" default="yes" />
                </xs:complexType>
        </xs:element>

        <xs:element name="mrcp_response">
```

```
            <xs:complexType>
                <xs:sequence>
                    <xs:element ref="mrcp_action" minOccurs="0" maxOccurs="1" />
                </xs:sequence>
                <xs:attribute name="asr" type="boolean_type" />
                <xs:attribute name="tts" type="boolean_type" />
                <xs:attributeGroup ref="response_attrgroup" />
            </xs:complexType>
        </xs:element>

        <xs:element name="mrcps_response">
            <xs:complexType>
                <xs:sequence>
                    <xs:element ref="mrcp_response" minOccurs="0" maxOccurs="unbounded"
/>
                </xs:sequence>
                <xs:attribute name="size" type="xs:string" use="required" />
            </xs:complexType>
        </xs:element>

        <xs:element name="web_service">
            <xs:complexType>
                <xs:choice minOccurs="0" maxOccurs="1">
                    <xs:element ref="call" />
                    <xs:element ref="call_response" />
                    <xs:element ref="calls_response" />
                    <xs:element ref="conference" />
                    <xs:element ref="conference_response" />
                    <xs:element ref="conferences_response" />
                    <xs:element ref="eventhandler" />
                    <xs:element ref="eventhandler_response" />
                    <xs:element ref="eventhandlers_response" />
                    <xs:element ref="mrcp" />
                    <xs:element ref="mrcp_response" />
                    <xs:element ref="mrcps_response" />
                    <xs:element ref="event" />
                    <xs:element ref="error" />
                </xs:choice>
                <xs:attribute name="version" type="xs:NMTOKEN" fixed="1.0" use="required"
/>
            </xs:complexType>
        </xs:element>
</xs:schema>
```

To simplify PowerMedia XMS RESTful application programming in Java, see the following Tech Note: http://www.dialogic.com/~/media/products/media-server-software/download/xms-demos/XMS-XMLBeans_Technote_20130405.pdf.

7. Text and Image Overlays

PowerMedia XMS supports the ability to apply text and images as captions in video conferences. When an overlay is applied to an existing video stream, text and image parameters are required. Each parameter is used as an XML attribute to describe the properties of the image or text overlay to the media engine. All text and image overlay parameters are sent in a string in the region_overlays parameter.

The region_overlays parameter specifies a set of text or image overlays for a conference layout. The overlays are defined by a semicolon delimited list of region definitions where each specifies a set of parameters that completely describes the overlay attributes for the conference layout regions or root window. Each region overlay is defined using this format: <REGION ID> EQUALS <ID> COMMA <PARAMETER> EQUALS <VALUE> COMMA <PARAMETER> EQUALS <VALUE>...

- REGION ID uniquely identifies the conference region where the overlays should be applied. Note, "0" is reserved and used ONLY when applying a text or image overlay to the root window of a conference.
- PARAMETER identifies the overlay parameter being set.
- VALUE identifies the value of the overlay parameter given.

Note the following when entering parameters into the region_overlays string:

- All parameters take only a single value.
- If a parameter value contains an ampersand (&) or equals (=) character, the character must be passed as a percent escaped hex value (e.g., for an ampersand (&), use %26).
- All parameters must be encoded without spaces between them and passed as a single string to the region_overlays field of the <conference> element.

Refer to the following example.

```
region_overlays -->
region=1,overlay_id=id1,left=10,top=80,hsize=80,vsize=15,priority=0.4,overlay_bgcolor=springgreen
,textstyle_id=textStyle1,fontFamily=Arial,fontStyle=normal,fontWeight=bold,fontEffects=oblique,fo
ntSize=90,fontColor=firebrick,fontDirection=lr,textstyle_bgcolor=gray,textAlignment=center,wrapOp
tion=noWrap,content_id=body1,applyMode=replace,p_id=textstring1,p_style=textStyle2,p_duration=30s
,encoding=UTF8,text=Region1;region=2,overlay_id=dialogic2,left=10,top=80,hsize=80,vsize=15,priori
ty=0.4,overlay_bgcolor=springgreen,textstyle_id=textStyle2,fontFamily=Arial,fontStyle=normal,font
Weight=bold,fontEffects=oblique,fontSize=90,fontColor=firebrick,fontDirection=lr,textstyle_bgcolo
r=gray,textAlignment=center,wrapOption=noWrap,content_id=body1,applyMode=replace,p_id=textstring1
,p_style=textStyle2,p_duration=30s,encoding=UTF8,text=Region2;...
```

The following tables list the parameters.

Overlay Bounding Frame Parameters

Parameter	Default	Optional	Description
overlay_id	(none)		Overlay identifier.
left	"0"	*	The position of the overlay from the left side of the reference window, defined as a % of the overall width of the reference window or conference region. Supported values range from 100.0000 to 100.0000 (%).

Parameter	Default	Optional	Description
top	"0"	*	The position of the overlay from the top of the reference window, defined as a % of the overall height of the reference window or conference region. Supported values range from 100.0000 to 100.0000 (%).
hsize	"0"	*	The horizontal size of the layout expressed as a % of the reference window or conference region horizontal size. Supported values range from 100.0000 to 100.0000 (%).
vsize	"0"	*	The vertical size of the layout expressed as a % of the reference window or conference region horizontal size. Supported values range from 100.0000 to 100.0000 (%)
priority		*	A number between 0 and 1 that is used to define the precedence when rendering overlapping layouts. When areas of different layouts overlap, they are layered in order of their "priority" attribute. The layout with the highest value for the "priority" attribute is below all other layouts and may be hidden by overlapping layouts. The layout with the lowest non-zero value for the "priority" attribute is on top of all other layouts and will not be hidden by overlapping layouts. The priority attribute may be assigned values between 0 and 1. A value of zero disables the layout deleting it and freeing any resources associated with the layout.
bwidth	"0 (no border)"	*	Horizontal and vertical border width of the overlay defined as a % of the overall height of the layout window. Supported values range from 0.0000 to 20.0000 (%).
hbwidth	"0 (no border)"	*	Horizontal border width of the overlay defined as a % of the overall height of the layout window. This parameter may be used if there is a desire to have different settings for the horizontal and vertical borders. For example, a vertical border may be desired while a horizontal border may not be desired. Supported values range from 0.0000 to 20.0000 (%).

Parameter	Default	Optional	Description
vbwidth	"0 (no border)"	*	Vertical border width of the overlay defined as a % of the overall height of the layout window. This parameter may be used if there is a desire to have different settings for the horizontal and vertical borders. For example, a vertical border may be desired while a horizontal border may not be desired. Supported values range from 0.0000 to 20.0000 (%).
bcolor	"gray"	*	Color of the overlay border.
bopacity	"100"	*	Defines the opacity of the border of the overlay region. It accepts a percentage value in the range 0-100(%), with 100 meaning fully opaque.
overlay_bgcolor	"blue"	*	Background color of the layout.
overlay_bgopacity	"100"	*	Defines the opacity of the background of the overlay region. It accepts a percentage value in the range 0-100(%), with 100 meaning fully opaque
overlay_duration	"persistent"	*	Specifies the duration for the overlay. Values may be: • persistent - The overlay region will persist until expressly deleted by the application. • lifeOfContent - The overlay region will persist until the content_duration expires.

Text Style Parameters

Parameter	Default	Optional	Description
textstyle_id	(none)		A name that can be used by p_style to refer to a style (style id).
fontfamily		*	Name of the font family. Values are "Arial", "Courier New", "Tahoma", "Times New Roman", "Verdana", etc.
fontstyle	"normal"	*	Font style. Values are "normal" and "italic".
fontweight	"normal"	*	Font weight. Values are "normal" and "bold".
fonteffects	"none"	*	Font effects. Values are "none" and "underlined", and "outlined".

Parameter	Default	Optional	Description
fontsize	"90"	*	Font size. Values are specified as a % of the vertical size of the layout region. Supported values range from 0.0 to 100.0 (%).
fontcolor	"white"	*	Color of text.
fontopacity	"100"	*	This attribute defines the opacity of the font color to be applied to the font when this style is used in p_style. It accepts a percentage value in the range 0-100%, with 100 meaning fully opaque.
fontdirection	"lr"	*	Font direction. Values supported are: lr (left to right), rl (right to left), tb (top to bottom), bt (bottom to top).
textstyle_bgcolor	"blue"	*	Background color to be applied to the layout when this style is applied in p_style.
textstyle_bgopacity	"100"	*	This attribute defines the opacity of the background color to be applied to the layout when this style is used in p_style. It accepts a percentage value in the range 0-100%, with 100 meaning fully opaque.
textalignment	"center"	*	Alignment of text within the layout region. Values supported are: center, centerLeft, centerRight, topLeft, bottomLeft, topRight, bottomRight, topCenter, bottomCenter. For the case where content applied is static, textAlignment refers to the positioning of static text within the overlay window. For the value center, text is centered within the layout window. For the case where content applied is scrolled, textAlignment does not apply.
wrap	"nowrap"	*	Wrap option. Values are: "wrap" (word wrap) and "nowrap". Wrap direction is top to bottom when fontdirection is either lr or rl, and is left to right when fontdirection is either tb or bt

Image Style Parameters

Parameter	Default	Optional	Description
imgstyle_id	(none)		A name that can be used by img to refer to a style (style id).
imgstyle_applymode	"resizeToFit"	*	Fill mode. Specifies how the image will be applied to the overlay window. Values are as follows: • resizeToFit - The image will be resized to fit within the overlay window while maintaining the aspect ratio of the image. • resizeToFill - The image will be resized in both the horizontal and/vertical dimensions if necessary. • maintainSize - The image is not resized. If the overlay image is equal in size or smaller than the overlay window, the image will be displayed in its entirety. If the overlay image is larger than the overlay window, in either the horizontal or vertical dimension, the image will be cropped when displayed.
imgsize	"90"	*	Image size. Values are specified as a % of the vertical size of the layout region. Supported values, when expressed as a percent, range from 0.0 to 100.0
imgstyle_bgcolor	"blue"	*	Background color to be applied to the layout when this style is applied using img.
imgstyle_bgopacity	"nowrap"	*	This attribute defines the opacity of the background color to be applied to the layout when this style is applied using img. It accepts a percentage value in the range 0-100(%), with 100 meaning fully opaque. The default value of this attribute is 0%. A imgstyle_bgopacity=0 results in a fully transparent background.

Parameter	Default	Optional	Description
imgalignment	"center"	*	Alignment of the image within the layout region. This attribute does not apply when the imgstyle_applymode is set to resizeToFill. Values supported are: center, centerLeft, centerRight, topLeft, bottomLeft, topRight, bottomRight, topCenter, bottomCenter. For the case where content applied is static, imgalignment refers to the positioning of a static image within the overlay window. For the value center, the image is centered within the layout window. For the case where content applied is scrolled, imgalignment does not apply.

Content Body Parameters

Parameter	Default	Optional	Description
content_id	(none)		A name that can be used to refer to the specified body and its contents
content_applymode	"replace"	*	Defines how the content is to be applied. Values are: replace, append, and delete
content_duration	"0.0"	*	Specifies the minimum length of time to maintain the last content item being displayed prior to completing and terminating the content element. Values are specified in seconds with a resolution of 100 ms.
contentexit	"false"	*	Results in a CONTENT_EXPIRY notification event being sent when the execution of the instruction contained within the content_duration expires, or when there is an error in attempting to overlay the content. Values are true or false.
scroll_mode	"scrollOnce"	*	The scrolling mode. Values are: scrollOnce (scroll content one time) and scrollContinuous (scroll continuously).
speed	"25"	*	Speed of content scrolling in % per second relative to the text layout region. Values supported are from 1 to 100 in increments of 1%.
direction	"rl"	*	Direction of content to be scrolled. Values supported are: lr (left to right), rl (right to left), tb (top to bottom), bt (bottom to top).

Parameter	Default	Optional	Description
padding	"5"	*	Specifies minimum padding to be added before the first content element (p_id or img_id), between each content element, and at the end of the last content element to be scrolled. Values are in % relative to the text layout region in the dimension (width, height) of scrolling and supported values are from 1 to 100 in increments of 1%.

Individual Image Parameters

Parameter	Default	Optional	Description
img_id	(none)		A name that can be used to refer to the image element.
img_style	(none)		Refers to an imgstyle_id to apply to the content.
img_duration	"0"	*	Specifies a time duration for the content to be displayed. Values are specified in seconds and milliseconds. A value of "0" indicates that the content should be displayed indefinitely.
img_uri	(none)	*	Identifies the location of the image to be overlaid. The file:// and http:// or https:// schemes are supported.
img_type	image/png	*	Specifies the image MIME type of the image to be overlaid. Values supported are: image/png and image/jpeg.

Individual Text Parameters

Parameter	Default	Optional	Description
p_id	(none)		A name that can be used to refer to the paragraph element.
p_style	(none)		Refers to a textstyle_id to apply to the content.
p_duration	"0"	*	Specifies time duration for the content to be displayed. Values are specified in seconds and milliseconds. A value of "0" indicates that the paragraph text should be displayed indefinitely. p_duration does not apply when content is scrolled.

Parameter	Default	Optional	Description
p_uri	(none)	*	Identifies the location of the text to be overlaid. The file:// and http:// or https:// schemes are supported. This attribute may be omitted if inline text is specified using the text attribute.
p_type	"text/plain"	*	Used with the p_uri attribute. Specifies the MIME type of the text to be overlaid. Values supported are: text/plain.
encoding	"UTF8"	*	The encoding type of the text. Values are: UTF8, ASCII, or GB18030.
text		*	Inline text (e.g., text=Red Alert).

8. XMSTool RESTful Utility

This section provides details about the XMSTool RESTful Utility (also referred to herein as "XMSTool" or "Utility"). XMSTool is used for developing, debugging, and supporting applications for the PowerMedia XMS using the HTTP RESTful API.

XMSTool is a Java-based test application for passing and receiving RESTful API messages to and from the PowerMedia XMS. It can be used to build and parse individual RESTful messages, and can drive and record simple applications.

The utility provides the following:

- Support for both 1PCC and 3PCC (see the Call Control Models)
- Ability to manually enter and execute the RESTful API commands and observe the results
- Method to record Macros for automated execution of command sequences (Demo mode), enabling users to create simple Demos and debug their applications
- Pre-recorded Macros available for commonly used call scenarios
- Logging capabilities

XMSTool can be run in two different modes:

- Demo/Simple Mode
  Uses predefined XML scripts; short application scenarios can be executed to demonstrate most of the PowerMedia XMS RESTful functionality. Session logging is available to examine the message interchange. Only sessions using inbound SIP calls are currently available in this mode.
- Advanced Mode
  Allows individual RESTful commands to be manually entered for full PowerMedia XMS control. This mode is intended to be used by developers who are looking to become familiar with the RESTful API messages used to control PowerMedia XMS. It also allows the individual commands that make up a Macro/Demo to be recorded for replay or to provide an accurate way to reproduce a problem in PowerMedia XMS.

For detailed information about using XMSTool, refer to the *Dialogic® PowerMedia™ XMS Installation and Configuration Guide*.

9. Appendix A: Media File Formats

The following section details the supported media containers and codecs.

Play

Audio

Format	Container	File Extension	audio_type	audio_rate
PCM (L8, L16, mulaw, alaw @ 8000, 11025 or 16000)	wav (RIFF)	.wav	audio/x-wav	None (RIFF header)
PCM (L8, L16, mulaw, alaw, AMR, AMR-WB @ 8000, 11025 or 16000)	Dialogic proprietary	.aud	audio/x-aud	None (header)
8bit/8kHz mu-law PCM	raw (header-less)	.ulaw .aud	audio/basic	None (fixed format)
8bit/8kHz a-law PCM	raw (header-less)	.alaw	audio/x-alaw-basic	None (fixed format)
8bit Linear PCM	raw (header-less)	.L8	audio/L8	rate=8000 or or 11025 or 16000
16bit linear PCM	raw (header-less)	.L16	audio/L16	rate=8000 or 11025 or 16000
G723 (not supported)	Dialogic proprietary		audio/G723	None (fixed format)
G726 (not supported)	Dialogic proprietary		audio/G726	None (fixed format)
G729 (not supported)	Dialogic proprietary		audio/G729	None (fixed format)
AMR	AMR, 3gp, mp4, mkv	.amr , .3gp, .mp4, .mkv	audio/AMR, audio/3gpp, audio/mp4, audio/mkv	None (header)
AMR-WB	AMR Wideband, 3gp, mp4, mkv	.awb , .3gp, .mp4, .mkv	audio/AMR-WB, audio/3gpp, audio/mp4, audio/mkv	None (header)

Format	Container	File Extension	audio_type	audio_rate
OPUS	mkv, webm	.mkv, .webm	audio/mkv, audio/webm	None (header)
EVS	evs	.evs	audio/evs	None (header)

Video

Format	Container	File Extension	video_type	video_mime_params
Dialogic proprietary	vid	.vid	video/x-vid	None (file header)
3gp	3gp	.3gp	video/3gpp	None (file header)
mp4	mp4	.mp4	video/mp4	None (file header)
mkv	mkv	.mkv	video/mkv	None (file header)
webm	webm	.webm	video/webm	None (file header)
image	JPEG	.jpeg, .jpg	image/jpeg	None (file header)

Record

Audio

Format	Container	File Extension	recording_audio_type	recording_audio_mime_params
PCM	wav (RIFF)	.wav	audio/x-wav	codec=L8 or L16 or mulaw or alaw or native **Note:** If the native codec is not supported by the container, L16 will be used as a fallback. rate=8000 or 11025(L8 and L16) or 16000(L8 and L16)
PCM	Dialogic proprietary	.aud	audio/x-aud	codec=L8 or L16 or mulaw or alaw or amr or amr-wb or native **Note:** If the native codec is not supported by the container, L16 will be used as a fallback. rate=8000 or 11025(L8 and L16) or 16000(L8 and L16) mode=0..7 (amr only) mode=0..8 (amr-wb only)

Format	Container	File Extension	recording_audio_type	recording_audio_mime_params
8bit/8kHz mu-law PCM	raw (header-less)	.ulaw, .aud	audio/basic	(none)
8bit/8kHz a-law PCM	raw (header-less)	.alaw	audio/x-alaw-basic	(none)
8bit Linear PCM	raw (header-less)	.L8	audio/L8	rate=8000 (default) or 11025 or 16000
16bit linear PCM	raw (header-less)	.L16	audio/L16	rate=8000 (default) or 11025 or 16000
G723 (not supported)	Dialogic proprietary		audio/G723	(none)
G726 (not supported)	Dialogic proprietary		audio/G726	(none)
G729 (not supported)	Dialogic proprietary		audio/G729	(none)
AMR	AMR	.amr	audio/AMR	mode=0..7 (default is 7)
AMR	3gp	.3gp	audio/3gpp	codec=AMR mode=0..7 (default is 7)
AMR	mp4	.mp4	audio/mp4	codec=AMR mode=0..7 (default is 7)
AMR	mkv	.mkv	audio/mkv	codec=AMR mode=0..7 (default is 7)
AMR-WB	AMR Wideband	.awb	audio/AMR-WB	mode=0..8 (default is 8)
AMR-WB	3gp	.3gp	audio/3gpp	codec=AMR-WB mode=0..8 (default is 8)
AMR-WB	mp4	.mp4	audio/mp4	codec=AMR-WB mode=0..8 (default is 8)
AMR-WB	mkv	.mkv	audio/mkv	codec=AMR-WB mode=0..8 (default is 8)
OPUS	mkv	.mkv	audio/mkv	codec=OPUS rate=16000
OPUS	webm	.webm	audio/webm	codec=OPUS rate=16000

Format	Container	File Extension	recording_audio_ type	recording_audio_ mime_params
EVS	evs	.evs	audio/evs	codec=EVS rate=8000 or 16000 (EVS Primary mode) bitrate=7200, 8000, 9600, 13200, 16400, 24400, 32000, 48000, 64000, 96000, or 128000 (EVS AMR-WB IO mode) bitrate=6600, 8850, 12650, 14250, 15850, 18250, 19850, 23050, or 23850

Video

Format	Container	File Extension	recording_video_ type	recording_video_ mime_params
Video	Dialogic proprietary	.vid	video/x-vid	codec=h263 or h264 or mp4v-es profile=[NUMBER] level=[NUMBER] framerate=[NUMBER] maxbitrate=[NUMBER] height=[NUMBER] width=[NUMBER]
Video	3gp	.3gp	video/3gpp	codec=h263 or h264 or mp4v-es profile=[NUMBER] level=[NUMBER] framerate=[NUMBER] maxbitrate=[NUMBER] height=[NUMBER] width=[NUMBER]
Video	mp4	.mp4	video/mp4	codec=h263 or h264 or mp4v-es profile=[NUMBER] level=[NUMBER] framerate=[NUMBER] maxbitrate=[NUMBER] height=[NUMBER] width=[NUMBER]

Format	Container	File Extension	recording_video_type	recording_video_mime_params
Video	mkv	.mkv	video/mkv	codec=h263 or h264 or mp4v-es or vp8 or vp9 profile=[NUMBER] level=[NUMBER] framerate=[NUMBER] maxbitrate=[NUMBER] height=[NUMBER] width=[NUMBER]
Video	webm	.webm	video/webm	codec=vp8 or vp9 profile=[NUMBER] level=[NUMBER] framerate=[NUMBER] maxbitrate=[NUMBER] height=[NUMBER] width=[NUMBER]
Image	JPEG	.jpeg, .jpg	image/jpeg	(none)

10. Appendix B: Feature Details

The following section describes the features supported and provides information on how to use these features.

Use Case for Dialogic Proprietary Header Tags in a SIP/WebRTC INVITE

PowerMedia XMS supports various combinations of SIP header indications to specify NAT and RTP profiles on SIP calls. This feature supports ICE (Lite), SDES, DTLS, AVPF/SAVPF, and combinations in a SIP INVITE when using Dialogic proprietary Supported header tags. The tags are provided to the customer application through the SIP Header in RESTful API calls. The RESTful application can extract the SIP header indications to control the call session response. Refer to the following table for a list of the header tags.

Tag	API Options	Description
dlgc-encryption-sdes	encryption=srtp	Enables sdes-srtp
dlgc-encryption-dtls	encryption=dtls	Enables dtls-srtp
dlgc-ice	ice=yes	Enables ICE (Lite)
dlgc-rtcp-feedback-audio	rtcp_feedback=audio	Enables AVPF/SAVPF for audio (not currently supported)
dlgc-rtcp-feedback-video	rtcp_feedback=video	Enables AVPF/SAVPF for video
dlgc-rtcp-feedback-audiovideo	rtcp_feedback=audiovideo	Enables AVPF/SAVPF for audio and video (only video is currently supported)
dlgc-rtcp-feedback-none	rtcp_feedback=none	Overrides configuration and disables RTCP feedback on audio and video (only video is currently supported) if configured to be enabled by default

Use Case 1

This use case describes how to enable ICE (Lite), SDES, and AVPF/SAVPF with the RESTful API in 1PCC mode. The application receives an incoming call event as follows:

```
<web_service version="1.0">
 <event type="incoming" … >
  ….
 <event_data name="headers.Supported" value="dlgc-encryption-sdes, dlgc-ice, dlg-rtcp-feedback-
 video"
  />
  ….
  <event_data name="uri" value="sip:rest@146.152.122.4" />
 </event>
</web_service>
```

To accept/answer the call:

```
<call  answer="yes" encryption="srtp" ice="yes" rtcp_feedback="video" />
```

Use Case 2

This use case describes how to enable ICE (Lite), DTLS, and AVPF/SAVPF with the RESTful API in 1PCC mode. The application receives an incoming call event as follows:

```
<web_service version="1.0">
 <event type="incoming" … >
  ….
 <event_data name="headers.Supported" value="dlgc-encryption-dtls, dlgc-ice, dlg-rtcp-feedback-
 video"
  />
  ….
  <event_data name="uri" value="sip:rest@146.152.122.4" />
 </event>
</web_service>
```

To accept/answer the call:

```
<call  answer="yes" encryption="dtls" ice="yes" rtcp_feedback="video" />
```

Enhanced Video Conference Layout Sizing

This section provides usage examples of how the enhanced video conference layout sizing (EVCLS) feature and associated attributes are referenced in RESTful request and control messages to manipulate and manage the video display of PowerMedia XMS conference resources. The three examples highlight the XML payload bodies within a series of RESTful requests (i.e., POST, PUT, GET, and DELETE) that illustrate the use of new attributes introduced with the EVCLS feature.

Create a Conference

The following example creates a conference with up to nine parties. Parties are added to the conference (not shown) and the active talker is assigned to region 1. The active talker is viewed by all parties in the conference. The new EVCLS attributes are used to customize the height, width, and background color of region 1. After creating the conference resource, the input stream of the active talker is placed in region 1 of the conference layout in "fill" aspect mode.

```
<web_service version="1.0">
```

```
   <conference type="audiovideo" max_parties="9" layout_regions="
   region=1,left=0,height=1/5,width=1/2,top=0,background_color=green,aspect_ratio_mode=fill"
   layout_bgcolor="blue" layout_size="vga" caption="no" beep="yes" clamp_dtmf="yes"
   auto_gain_control="yes" echo_cancellation="yes" active_talker_region="1"
   active_talker_interval="500ms" />
</web_service>
```

Update the Conference to Crop Mode

The following example changes the mode for the active talker in region 1 to "crop" mode.

```
<web_service version="1.0">
   <conference type="audiovideo" max_parties="9" layout_regions="
   region=1,left=0,height=1/5,width=1/2,top=0,background_color=green,aspect_ratio_mode=crop"
   layout_bgcolor="blue" layout_size="vga" caption="no" beep="yes" clamp_dtmf="yes"
   auto_gain_control="yes" echo_cancellation="yes" active_talker_region="1"
   active_talker_interval="500ms" />
</web_service>
```

Update the Conference to Fit Mode

The following example changes the mode for the active talker in region 1 to"fit" mode.

```
<web_service version="1.0">
   <conference type="audiovideo" max_parties="9" layout_regions="
   region=1,left=0,height=1/5,width=1/2,top=0,background_color=green,aspect_ratio_mode=fit"
   layout_bgcolor="blue" layout_size="vga" caption="no" beep="yes" clamp_dtmf="yes"
   auto_gain_control="yes" echo_cancellation="yes" active_talker_region="1"
   active_talker_interval="500ms" />
</web_service>
```

WebM Container

This section provides examples that highlight the XML payload bodies within a series of RESTful requests that illustrate the use of new attributes for WebM. These examples focus on the payload scripts in the RESTful request.

Record Audio and Video

In the following example, audio and video is recorded to a WebM container from a call leg.

```
<web_service version="1.0">
<call>
   <call_action>
      <record
               recording_video_uri="file:///var/tmp/media/QUALITY-V_VP8_720p_15_2000-
A_OPUS_16K_20FS_1FPP.webm"
               recording_audio_uri="file:///var/tmp/media/QUALITY-V_VP8_720p_15_2000-
A_OPUS_16K_20FS_1FPP.webm"
               recording_video_type="video/webm"
               recording_audio_type="audio/webm" max_time="23s">
        <recording_audio_mime_params codec="OPUS"/>
        <recording_video_mime_params codec="vp8" level="3.1" profile="66" framerate="15"
maxbitrate="2000000" height="720" width="1280"/>
      </record>
   </call_action>
</call>
</web_service>
```

Play Back the Audio and Video Stream

In the following example, the audio and video stream that was recorded in the previous example is played back.

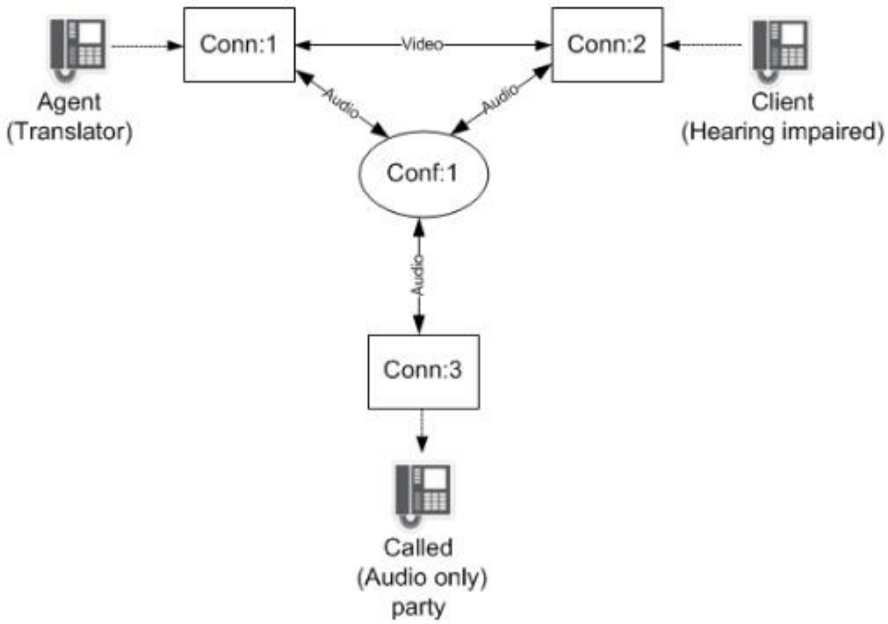
```
<web_service version="1.0">
<call>
   <call_action>
      <play offset="0s" repeat="0" delay="0s" terminate_digits="9" max_time="infinite"
skip_interval="0s">
         <play_source
            video_uri="file:///var/tmp/media/QUALITY-V_VP8_720p_15_2000-
A_OPUS_16K_20FS_1FPP.webm"
            audio_uri="file:///var/tmp/media/QUALITY-V_VP8_720p_15_2000-
A_OPUS_16K_20FS_1FPP.webm"
                />
      </play>
   </call_action>
</call>
</web_service>
```

Joining Separate Audio and Video Streams

PowerMedia XMS provides the ability to separate the audio and video streams so media can be routed separately on the join command. The ability to separate multimedia join into audio and video streams is available for connections between call connections and conferences.

With this feature, it is possible to join video between two callers while joining the audio of the callers into an audio conference. This capability enables an application to provide a peer-to-peer video connection between the two callers while sending the audio into an audio conference where other audio only callers can join the conversation. The use case is applicable when it is intended that only two video callers see each other and it is not desirable to utilize a video conference mix. Audio can be joined separately to an audio conference so that other audio only participants can be conferenced in later.

This section provides an example of joining video between two parties while sending their audio to a third, audio-only participant. Refer to the following diagram and example scripts.



Create the Audio Conference

The following example creates an audio conference.

```
<web_service version="1.0">
     <web_service version="1.0">
       <conference auto_gain_control="yes" beep="yes" caption="yes" caption_duration="infinite"
       clamp_dtmf="yes" echo_cancellation="yes" layout="0" max_parties="3" reserve="3"
       type="audio"/>
</web_service>
```

Join Video Streams

The following example joins the video streams of parties 1 and 2 (full duplex).

```
<web_service version="1.0">
       <call>
       <call_action>
              <join audio="inactive" call_id="8de261c9-e027-4d92-8cda-7095b7ff16d8"
              video="sendrecv"/>
       </call_action>
    </call>
</web_service>
```

Join the Audio Streams to the Conference

The following example joins party 1's audio stream to the conference (full duplex).

```
<web_service version="1.0">
       <call>
       <call_action>
              <add_party audio="sendrecv" auto_gain_control="yes" clamp_dtmf="yes"
              conf_id="ddda0ccc-d6a9-4ec1-b32d-8d7f3a064d08" echo_cancellation="yes" region="0"
              video="inactive"/>
       </call_action>
```

```
    </call>
</web_service>
```

The following example joins party 2's audio stream to the conference (full duplex).

```
<web_service version="1.0">
       <call>
       <call_action>
              <add_party audio="sendrecv" auto_gain_control="yes" clamp_dtmf="yes"
              conf_id="ddda0ccc-d6a9-4ec1-b32d-8d7f3a064d08" echo_cancellation="yes" region="0"
              video="inactive"/>
       </call_action>
    </call>
</web_service>
```

The following example joins party 3's audio stream to the conference (full duplex). Party 3 is the audio-only caller.

```
<web_service version="1.0">
       <call>
       <call_action>
              <add_party audio="sendrecv" auto_gain_control="yes" clamp_dtmf="yes"
              conf_id="ddda0ccc-d6a9-4ec1-b32d-8d7f3a064d08" echo_cancellation="yes" region="0"
              video="inactive"/>
       </call_action>
    </call>
</web_service>
```

Record One of the Parties

In the following example, party 2 is recorded.

```
<web_service version="1.0">
       <call>
  <call_action>
     <record max_time="70s" recording_audio_type="audio/3gpp"
     recording_audio_uri="file:///testing/mxml/recordings/tr_caller_2.3gp"
     recording_video_type="video/3gpp"
     recording_video_uri="file:///testing/mxml/recordings/tr_caller_2.3gp">
     <recording_audio_mime_params codec="AMR"/>
     </record>
   </call_action>
    </call>
</web_service>
```

Automatic Deletion of Silence Recordings

The following describes the conditions that will result in PowerMedia XMS automatically deleting a recording, or the expected behavior in the record completion event in such a case.

In RESTful, a recording will be automatically deleted under the following conditions:

1. The recording terminates due to noinput_timeout trigger.
2. The recording is terminated by any means (max_time, max_silence, terminate_digits, stop, etc.) when noinput_timeout is not "infinite" and only silence data has been recorded.

When a recording is deleted:

1. The event data indicating the location of the recording will be empty (e.g., audio_location and video_location).
2. The reason will be set to "no input".
3. The duration will be set to the actual amount of time that the record operation was recording for. It will not be set to "0".

Encryption Record

PowerMedia XMS supports encryption record, enabling applications to record encrypted audio and video files. It is designed to provide secure recording capability within PowerMedia XMS so that recorded files are encrypted as they are stored, and the encryption key is secured by RSA key pair provided by the application. PowerMedia XMS will securely encrypt the file as it writes encoded data to the disk and at no time is unencrypted data written to the disk.

With this feature, the application requests an encrypted recording and provides a public key with the record function so the encryption key is returned securely to the application. The encrypted recording can only be decrypted by the client application that maintains the private/public key pair. This encryption record feature is supported with the following file formats: WebM (.webm) and MKV(.mkv) for audio only and multimedia files.

Note: Playback of encrypted files is not supported by PowerMedia XMS. If an attempt is made to playback an encrypted recording, the play will fail. The error generated by PowerMedia XMS is indeterminate. For example, it may generate "audio track does not exist", "video track does not exist", or other related errors.