



Dialogic® PowerMedia™ XMS RESTful API

User's Guide

Copyright and Legal Notice

Copyright © 2012-2016 Dialogic Corporation. All Rights Reserved. You may not reproduce this document in whole or in part without permission in writing from Dialogic Corporation at the address provided below.

All contents of this document are furnished for informational use only and are subject to change without notice and do not represent a commitment on the part of Dialogic Corporation and its affiliates or subsidiaries ("Dialogic"). Reasonable effort is made to ensure the accuracy of the information contained in the document. However, Dialogic does not warrant the accuracy of this information and cannot accept responsibility for errors, inaccuracies or omissions that may be contained in this document.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH DIALOGIC® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN A SIGNED AGREEMENT BETWEEN YOU AND DIALOGIC, DIALOGIC ASSUMES NO LIABILITY WHATSOEVER, AND DIALOGIC DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF DIALOGIC PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHT OF A THIRD PARTY.

Dialogic products are not intended for use in certain safety-affecting situations. Please see <http://www.dialogic.com/company/terms-of-use.aspx> for more details.

Due to differing national regulations and approval requirements, certain Dialogic products may be suitable for use only in specific countries, and thus may not function properly in other countries. You are responsible for ensuring that your use of such products occurs only in the countries where such use is suitable. For information on specific products, contact Dialogic Corporation at the address indicated below or on the web at www.dialogic.com.

It is possible that the use or implementation of any one of the concepts, applications, or ideas described in this document, in marketing collateral produced by or on web pages maintained by Dialogic may infringe one or more patents or other intellectual property rights owned by third parties. Dialogic does not provide any intellectual property licenses with the sale of Dialogic products other than a license to use such product in accordance with intellectual property owned or validly licensed by Dialogic and no such licenses are provided except pursuant to a signed agreement with Dialogic. More detailed information about such intellectual property is available from Dialogic's legal department at 6700 Cote-de-Liesse Road, Suite 100, Borough of Saint-Laurent, Montreal, Quebec, Canada H4T 2B5. **Dialogic encourages all users of its products to procure all necessary intellectual property licenses required to implement any concepts or applications and does not condone or encourage any intellectual property infringement and disclaims any responsibility related thereto. These intellectual property licenses may differ from country to country and it is the responsibility of those who develop the concepts or applications to be aware of and comply with different national license requirements.**

Dialogic, Dialogic Pro, Dialogic Blue, Veraz, Brooktrout, Diva, BorderNet, PowerMedia, PowerVille, PowerNova, MSaaS, ControlSwitch, I-Gate, Mobile Experience Matters, Network Fuel, Video is the New Voice, Making Innovation Thrive, Diastar, Cantata, TruFax, SwitchKit, Eiconcard, NMS Communications, SIPcontrol, Exnet, EXS, Vision, inCloud9, NaturalAccess and Shiva, among others as well as related logos, are either registered trademarks or trademarks of Dialogic Corporation and its affiliates or subsidiaries. Dialogic's trademarks may be used publicly only with permission from Dialogic. Such permission may only be granted by Dialogic's legal department at 6700 Cote-de-Liesse Road, Suite 100, Borough of Saint-Laurent, Montreal, Quebec, Canada H4T 2B5. Any authorized use of Dialogic's trademarks will be subject to full respect of the trademark guidelines published by Dialogic from time to time and any use of Dialogic's trademarks requires proper acknowledgement.

The names of actual companies and products mentioned herein are the trademarks of their respective owners.

This document discusses one or more open source products, systems and/or releases. Dialogic is not responsible for your decision to use open source in connection with Dialogic products (including without limitation those referred to herein), nor is Dialogic responsible for any present or future effects such usage might have, including without limitation effects on your products, your business, or your intellectual property rights.

Table of Contents

Dialogic® PowerMedia™ XMS RESTful API	1
1. Welcome	10
2. Overview	11
RESTful API Description.....	12
Client Side Technologies	12
RESTful API with HTTP Methods	13
RESTful API Request/Response Model	13
XML Schema Definition.....	14
Event Streaming.....	15
3. Resource-Based Element Introduction.....	17
Applications and Application IDs (appid)	17
Time Values	18
4. API Resources	19
5. List of Available Resources	20
Call Resource	20
calls	20
HTTP GET	20
HTTP POST	21
Call Concepts	27
Call Sub-Resource	28
call	28
HTTP GET	28
HTTP PUT	28
HTTP DELETE	75
Conference Resource	75
conferences	75
HTTP GET	75
HTTP POST	75
Conference Concepts	80
Conference Sub-Resource.....	80
conference.....	80
HTTP GET	80
HTTP PUT	81
HTTP DELETE	91
Event Handler Resource	91
eventhandlers	92
HTTP GET	94
HTTP POST	94
Event Handler Sub-Resource.....	95
eventhandler.....	95
HTTP GET	95
HTTP PUT	97
HTTP DELETE	98
Events	98
Media Events	99
Call Events	103
MRCP Events.....	110
Conference Events	112

MRCP Resource	112
mrcps	113
HTTP GET	113
HTTP POST	113
MRCP Sub-Resource	114
mrcp	114
HTTP GET	114
HTTP PUT	114
HTTP DELETE	122
6. XML Schema Definition of Elements	123
7. Dynamic Text and Image Generation	140
Document Templates	140
Style Definition	141
Creating the Image using the RESTful API	143
Combining the Three Data Sources	144
8. XMSTool RESTful Utility	145
9. Appendix A: Media File Formats	146
Audio Play	146
Audio Record	149
Video Play	151
Video Record.....	152

Revision History

Revision	Release Date	Notes
05-2703-009 (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. Removed WebRTC support.
05-2703-009 (Updated)	September 2015	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.

Revision	Release Date	Notes
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. 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.

Revision	Release Date	Notes
		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 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.

Revision	Release Date	Notes
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 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.

Revision	Release Date	Notes
		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: April 2016		

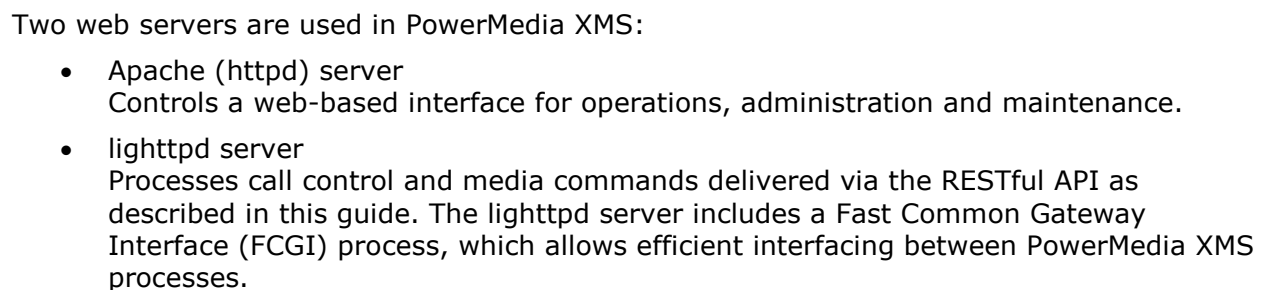
Refer to www.dialogic.com for product updates and for information about support policies, warranty information, and service offerings.

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.

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.



PowerMedia XMS provides two call control models:

- First party call control (1PCC)
The application sends commands to the PowerMedia XMS to establish SIP 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 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 protocol.

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 "".

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

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>
```

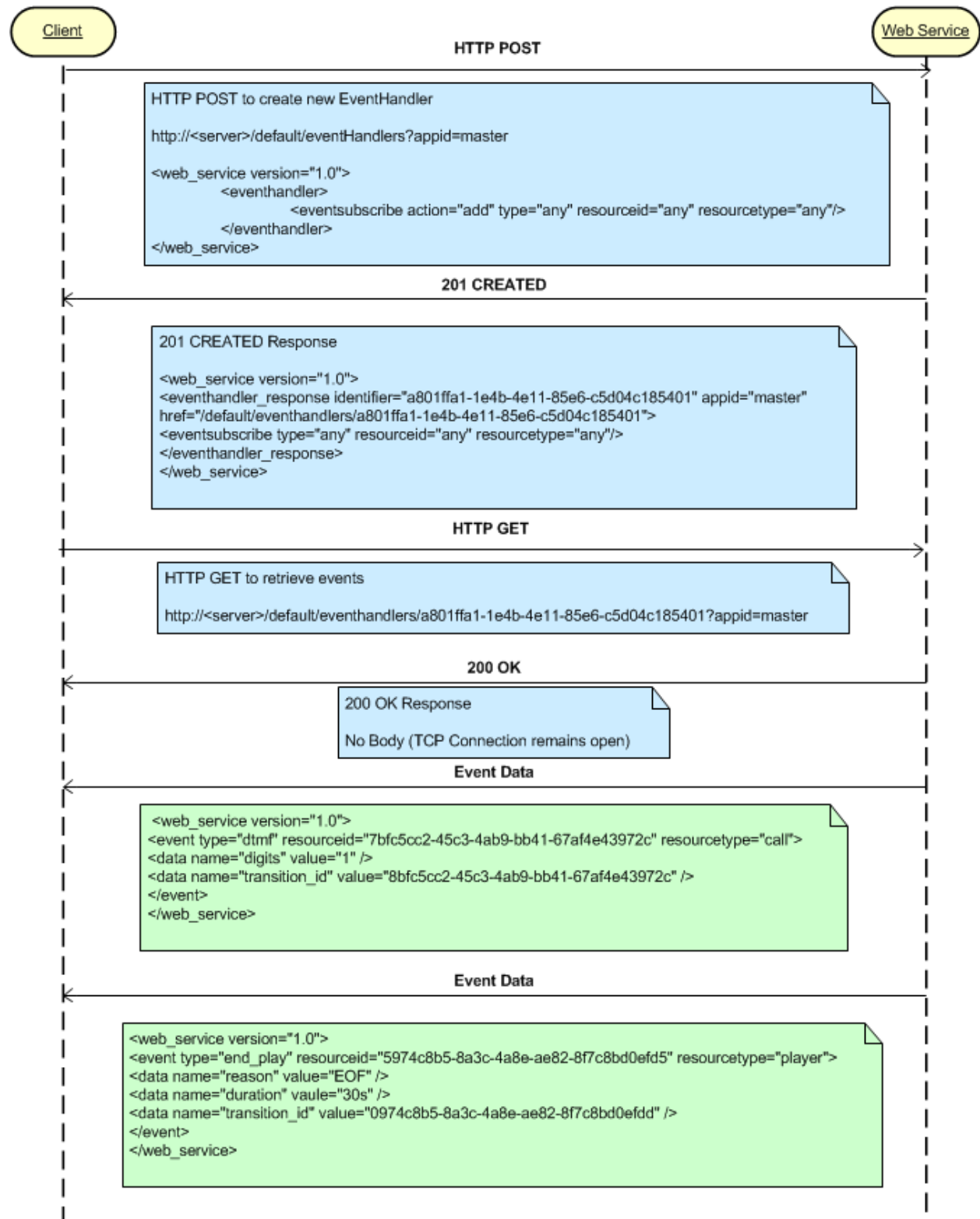
Refer to the specific resource-based element sections for more information.

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:



3. Resource-Based Element Introduction

There are four (4) 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 URI:

```
http://[ipaddress:port]/default/  
default port: 81
```

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 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 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 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](#), [overlay](#), [join/unjoin](#), and [stop](#).

For details on 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.

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-bec6-5d4ddeb5649" info_ack_mode="automatic"
media="audiovideo" signaling="yes"

        source_uri="sip:Username@10.20.129.113:5060" video="sendrecv">
    </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">
    </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 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 "
" prior to sending.
media	"audio"	*	Sets the media type supported by the call. Values: • "audio" • "audiovideo"

Parameter	Default	Optional	Description
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"
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.
source_uri	(none)	*	Caller address. For SIP, this is the From header.
called_uri	(none)	*	Logical destination address. For SIP, this is the To header.
display_name	(none)	*	Caller's display name.

Parameter	Default	Optional	Description
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.
headers	(none)	*	Raw SIP headers, delimited by the <CR><LF> end-of-line characters (outbound call only).

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">
  </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">
```

```
</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 PowerMedia XMS GUI in the Tones screen (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 on the PowerMedia XMS 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.

Thus, the file:// URL used in a RESTful Play command would point at a set of audio and video files named *xmstool_play.wav* and *xmstool_play.vid*, both located in the xmstool directory in the default media location:

```
<play_source location="file://xmstool/xmstool_play"/>
```

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

```
<playrecord recording_uri="file://xmstool/xmstool_recording" max_time="10s" offset="0s"
repeat="0" delay="1s"
```

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-bec6-5d4ddeb5649"
    identifier="8ae87129-b334-4d8a-bec6-5d4ddeb5649" info_ack_mode="automatic"
media="audiovideo" signaling="yes"
    source_uri="sip:Username@10.20.129.113:5060" video="sendrecv">
  </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" 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"
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.
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).

Parameter	Default	Optional	Description
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"
message	"sendrecv"	*	Direction of message media (MSRP). Values: "inactive" "sendonly" "recvonly" "sendrecv"
local_rtp_address	"sendrecv"	*	Sets the local IP address to use for RTP.

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">
  </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">
    </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 <code>async_dtmf</code> 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"
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"

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">
    </call_response>
</web_service>

```

Perform Call Action

- [play](#)
- [update_play](#)
- [record](#)
- [playcollect](#)
- [playrecord](#)
- [overlay](#)
- [stop](#)
- [join/unjoin](#)
- [add_party/update_party](#)
- [remove_party](#)
- [send_dtmf](#)
- [send_info/send_info_ack](#)
- [send_message](#)
- [transfer](#)
- [redirect](#)
- [hangup](#)
- [get_call_info](#)
- [add_ice_candidate](#)
- [send_hangup_ack](#)
- [get_last_action](#)
- [get_last_event](#)

play_source_attributes

Parameter	Default	Optional	Description
location	(none)		The URI of the media to be played e.g., "file://...", "rtsp://...", "image:", "http(s)://..." DEPRECATED use audio_uri and video_uri.
base_audio_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. • "audio/x-wav" • "audio/basic" • "audio/x-alaw- basic" • "audio/L8" • "audio/L16" • "audio/x-aud"
audio_rate	(none)		The mime type of the audio media. • "8000" • "11025" • "16000"
base_video_uri	(none)	*	Base URI prefix for video_uri URIs e.g., "file://...", "rtsp://...", "http(s)://..."

Parameter	Default	Optional	Description
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. • "video/x-vid" • "image/jpeg"

recording_audio_mime_params

Parameter	Default	Optional	Description
codec	(none)	*	Sets the audio codec. Values: • "L16" • "mulaw" • "alaw" • "L8" • "AMR" • "AMR-WB"
rate	(none)	*	Sets the audio rate. Values: • "8000" • "11025" • "16000 (L16 only)"

Parameter	Default	Optional	Description
mode	(none)	*	Audio AMR/AMR-WB mode. Values: "mode=0..7 (AMR only)" "mode=0..8 (AMR-WB only)"

recording_video_mime_params

Parameter	Default	Optional	Description
codec	(none)	*	Sets the video codec. Values: "h263" "h264" "mp4v-es"
profile	(none)	*	Sets the video profile.
level	(none)	*	Sets the video level.
framerate	(none)	*	Sets the video framerate.
maxbitrate	(none)	*	Sets the video maxbitrate.
height	(none)	*	Sets the video height.
width	(none)	*	Sets the video width.

dvr_setting_attributes

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

Parameter	Default	Optional	Description
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.

content_element_attributes

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

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.

Parameter	Default	Optional	Description
content_disposition	"attachment"	*	The intended disposition of the file. Values: "attachment" "render"
content_id	(none)		Content identifier.

msg_payload_content attributes

Parameter	Default	Optional	Description
content	(none)		Data.

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.

Parameter	Default	Optional	Description
delay	"1s"	*	Time delay between repeated plays.
terminate_digits	"#"	*	The digit or digits [0-9,*,#] used to terminate the play (call only).
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 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">
  <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>
```

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">
    <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>

```

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.

Request Payload Example

```

<web_service version="1.0">
  <call>
    <call_action>
      <update_play transaction_id="9d608231-a164-4102-b9e6-3balf0671a53" 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">
    <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://"). See also media DEPRECATED.
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. "audio/x-wav" "audio/basic" "audio/x-alaw-basic" "audio/G723" "audio/G726" "audio/G729"
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. "video/x-vid"
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.
noinput_timeout	"infinite"	*	The maximum time to wait for audio to be detected. Use "infinite" for no limit.

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.wav"
        recording_audio_type="audio/x-wav"
        recording_video_uri="file://verification/beta_recorded_file.vid"
        recording_video_type="video/x-vid" >
        <recording_video_mime_params codec="h264" level="3.1" height="480" width="640"
maxbitrate="768000" framerate="15"/>
        <recording_audio_mime_params codec="mulaw" rate="8000"/>
      </record>
    </call_action>
  </call>
</web_service>
```

Response Payload Example

```
<web_service version="1.0">
  <call_response identifier="7b3652ed-594f-482d-a349-7625c7cf86ef" appid="app"
    href="http://10.20.129.61:81/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">
    <call_action>
      <record transaction_id="b3407431-801f-4a44-ba8b-fa8c9fddd653"
        terminate_digits="#"
        max_time="10s"
        max_silence="infinite"
        noinput_timeout="infinite"
        recording_audio_uri="file://verification/beta_recorded_file.wav"
        recording_audio_type="audio/x-wav"
        recording_video_type="video/x-vid"
```

```

        recording_video_uri="file://verification/beta_recorded_file.vid">
        <recording_audio_mime_params codec="mulaw" rate="8000"/>
        <recording_video_mime_params codec="h264" level="3.1" framerate="15"
maxbitrate="768000" height="480" width="640"/>
    </record>
</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. See also media DEPRECATED.
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. "audio/x-wav" "audio/basic" "audio/x-alaw-basic"

Parameter	Default	Optional	Description
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. "video/x-vid"
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.
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"

Parameter	Default	Optional	Description
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"
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.

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">
    <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>
```

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 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 <i>timeout</i> 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"

Parameter	Default	Optional	Description
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 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="c87fccal-b2d0-49c5-8b89-baaaae71cf695" appid="app"
    href="/default/calls/c87fccal-b2d0-49c5-8b89-baaaae71cf695"
    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">
    <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. ""image:"id=template&a=b...".
duration	"infinite"	*	The length of time that the overlay is shown. Use "infinite" without the quotes to display the overlay until explicitly stopped.

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">
    <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">
    <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>
```

stop

Request Payload Attributes

Parameter	Default	Optional	Description
transaction_id	(none)		Identifier returned by play, playcollect, playrecord, overlay, 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">
    <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 (add_party only).
caption	(none)	*	Text for caption e.g., caller name.
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 <i>no</i> preference and replaced by this party.

Parameter	Default	Optional	Description
audio	"recvonly"	*	Sets the conference audio participation. Values: • "inactive" - No audio • "sendonly" - Only transmit audio • "recvonly" - Only receive audio • "sendrecv" - Full duplex audio
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"

Parameter	Default	Optional	Description
echo_cancellation	The value specified create conference	*	Determines if echo cancellation should be used. Values: • "yes" • "no"
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.

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 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">
    <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">
        <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 Example

```

<web_service version="1.0">
    <call>
        <call_action>
            <remove_party />
        </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">
    <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 (join only).

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_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">
        <call_action>
            <join call_id="830b9fda-d89e-495d-b7a1-6a63402bcdcf" />

```

```

        </call_action>
    </call_response>
</web_service>

unjoin
<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">
        <call_action>
            <unjoin call_id="830b9fda-d89e-495d-b7a1-6a63402bcdcf"/>
        </call_action>
    </call_response>
</web_service>

```

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).

Request Payload Example

send_info

```

<web_service version="1.0">
    <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">
    <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">
    <call_action>
      <send_info_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 <i>n</i> times.
content_element	(none)	*	Refer to content_element . Can be repeated <i>n</i> times.

Parameter	Default	Optional	Description
mode	"signalling"	*	Message transfer mode. Values: 1. "signalling" rtc or sip (future). 2. "msrp" MRSP instant message (rfc4975). Existing call only. 3. "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.
msg_multipart_type	(none)	* (required if there is more than 1 msg_payload)	Multi-part mime type for the msg_payload[] pseudo array. rfc4975 only.
report	(none)	*	Specifies the report type in the END_SEND_MESSAGE event. Values: 1. "success" 2. "failure" 3. "both"

Parameter	Default	Optional	Description
headers	(none)	*	Raw SIP headers, delimited by the <CR><LF> end-of-line characters ("rfc5547" only).

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">
    <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>
```

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">
    <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">
    <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">
        <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">
    <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">
    <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"
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.

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">
<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">
<call_action>
<get_last_action>
<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="#">
  <play_source location="file://verification/play_menu"/>
  <dvr_setting forward_key="1"
        backward_key="2"
        pause_key="3"
        resume_key="4"
        restart_key="5"/>
</play></get_last_action>
</call_action>
</call_response>
</web_service>

```

get_last_event

To get the last event.

Request Payload Example

```

<web_service version="1.0">
  <call>
    <call_action>
      <get_last_event/>
    </call_action>
  </call>
</web_service>

```

Response Payload Example

```

<web_service version="1.0">
  <event type="end_play" resource_id="76618368-ac8a-4288-8bad-8a3ce3d4b1b5" resource_type="call">
    <event_data name="duration" value="6180" />
    <event_data name="reason" value="end" />
    <event_data name="status" value="0 No Error" />
    <event_data name="transaction_id" value="351d7173-f774-4c20-a279-8640359fc9f0" />
  </event>
</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](#).

For details on 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
type	"audio"	*	Sets the media supported by conference. Values: "audio" "audiovideo"
max_parties	"9"	*	Maximum number of parties in a conference. Range: 2-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.

Parameter	Default	Optional	Description
layout_regions		*	Specifies a custom video layout. When defined this takes precedence over the layout parameter. The layout is defined a semicolon delimited list of region definitions. Each region is defined using the format: <ID> EQUALS <LEFT> COMMA <TOP> COMMA <RELATIVE_SIZE>[COMMA PRIORITY]. • ID uniquely identifies the region. Note, the value "0" is reserved and may not be used as a region identifier. • The LEFT and TOP parameters define the position of the top left corner of the region as an offset from the top left corner of the root window, they are expressed as percentages. • RELATIVE_SIZE defines the size of a region relative to the size of the root window 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: "left = 5, 5, 50, 2; right = 45, 45, 1/2, 1".

Parameter	Default	Optional	Description
layout_size	"automatic"	*	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.
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"

Parameter	Default	Optional	Description
auto_gain_control	"yes"	*	Determines if automatic gain control should be used. Values: "yes" "no"
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. Values range from 2 to 10 inclusive.

Request Payload Example

```

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

```

Response Payload Example

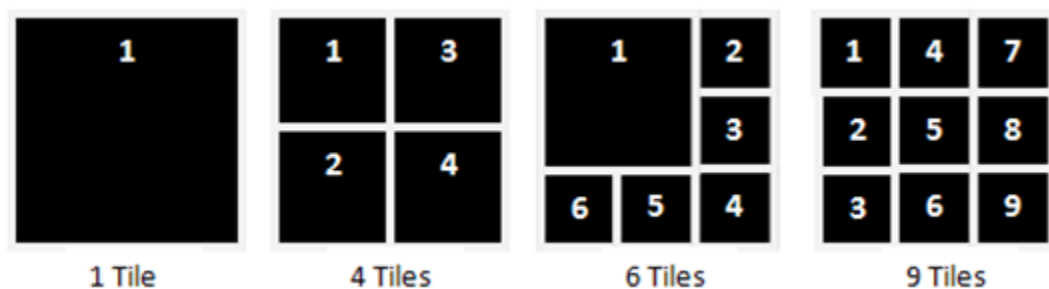
```
<web_service version="1.0">
  <conference_response appid="app" auto_gain_control="yes" beep="yes" caption="yes"
caption_duration="30s"
                                clamp_dtmf="yes" echo_cancellation="yes"
                                href="/default/conferences/830b9fda-d89e-495d-b7a1-
6a63402bcdcf"
                                identifier="830b9fda-d89e-495d-b7a1-6a63402bcdcf" layout="2"
                                layout_size="automatic"
                                max_parties="2" reserve="2" type="audiovideo">
    </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"
                                mode="normal" mute="no" privilege="no"
                                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.

Parameter	Default	Optional	Description
layout_regions	(none)	*	Specifies a custom video layout. When defined this takes precedence over the layout parameter. The layout is defined a semicolon delimited list of region definitions. Each region is defined using the format: <ID> EQUALS <LEFT> COMMA <TOP> COMMA <RELATIVE_SIZE>[COMMA PRIORITY]. • ID uniquely identifies the region. Note, the value "0" is reserved and may not be used as a region identifier. • The LEFT and TOP parameters define the position of the top left corner of the region as an offset from the top left corner of the root window, they are expressed as percentages. • RELATIVE_SIZE defines the size of a region relative to the size of the root window 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: "left = 5, 5, 50, 2; right = 45, 45, 1/2, 1".

Parameter	Default	Optional	Description
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.
active_talker_region	(none)	*	Specifies the ID of the region used to display the active talker's video. Use "0" to disable.
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.

Request Payload Example

```
<web_service version="1.0">
  <conference layout_size="vga"/>
</web_service>
```

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">
```

```

6a63402bcdcf"
layout_size="vga"
bbaae71cf695"
echo_cancellation="yes"
region="0"

href="/default/conferences/830b9fda-d89e-495d-b7a1-
identifier="830b9fda-d89e-495d-b7a1-6a63402bcdcf" layout="2"
max_parties="2" reserve="2" type="audiovideo">
<conf_participant audio="sendrecv" auto_gain_control="yes"
call_id="c87fccal-b2d0-49c5-8b89-
caption="Username" clamp_dtmf="yes"
mode="normal" mute="no" privilege="no"
tx_mute="no" video="sendrecv"/>

</conference_response>
</web_service>

```

Perform Conference Action

- [play](#)
- [update_play](#)
- [record](#)
- [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.
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.

Parameter	Default	Optional	Description
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 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>
      <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="c87fccal-b2d0-49c5-8b89-baaaae71cf695"
            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.

Parameter	Default	Optional	Description
dvr_action	(none)		Values: • "backward" - skip backwards. • "forward" - skip forward. • "pause" - pause playback. • "restart" - jump back to the start. • "resume" - resume paused playback.

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="c87fccal-b2d0-49c5-8b89-baaaae71cf695"
      audio="sendrecv"
      clamp_dtmf="yes"
      mute="no" tx_mute="no"
      video="sendrecv" caption="Username" region="0"
      auto_gain_control="yes" echo_cancellation="yes"
      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://..." which must refer to an existing directory or "http://" ("https://"). See also media DEPRECATED.
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. • "audio/x-wav" • "audio/basic" • "audio/x-alaw-basic"
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. • "video/x-vid"
max_time	"infinite"	*	The maximum length of time to record. Use "infinite" for no limit.

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>
```

```

    </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="c87fccal-b2d0-49c5-8b89-baaae71cf695"
      video="sendrecv" caption="Username" region="0"
      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>

```

stop

Request Payload Attributes

Parameter	Default	Optional	Description
transaction_id	(none)		Identifier returned by play or 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="c87fccal-b2d0-49c5-8b89-baaaae71cf695"
      audio="sendrecv"
      video="sendrecv" caption="Username" region="0"
      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="c87fccal-b2d0-49c5-8b89-baaaae71cf695"
      audio="sendrecv"
      video="sendrecv" caption="Username" region="0"
      auto_gain_control="yes" echo_cancellation="yes"
      clamp_dtmf="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 details on 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]

eventssubscribe_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"

Parameter	Default	Optional	Description
			• "dtmf" • "tone" • "any" • "end_speak" • "start_of_input" • "end_recognize" • "accepted" • "answered" • "updated" • "active_talker" • "alarm" • "prack" • "stream" • "message" • "conf_overlay_expired" • "end_send_message" • "speech_marker" (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" • "mrpc" • "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">
      <eventssubscribe 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
eventssubscribe		*	Refer to eventssubscribe_attributes .

Request Payload Example

```
<web_service version="1.0">
  <eventhandler>
    <eventssubscribe 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">
    <eventssubscribe 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" • "accepted" • "answered" • "updated" • "active_talker" • "alarm" • "prack" • "stream" • "message" • "conf_overlay_expired" • "end_send_message" • "speech_marker"
resource_id	(none)	*	Monitor events for a specific resource.

Parameter	Default	Optional	Description
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="c87fcc1-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
eventssubscribe	(none)		Refer to eventssubscribe_attributes .

Request Payload Example

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

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.

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" value="30000ms" />
    <event_data name="transaction_id" value="0974c8b5-8a3c-4a8e-ae82-8f7c8bd0efdd" />
  </event>
</web_service>
```

end_playcollect

Completion event for playcollect ([call_playcollect](#)).

- **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.

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>
</web_service>
```

end_record

Completion event for record ([call_record](#)).

- **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.

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>
</web_service>
```

end_playrecord

Completion event for playrecord ([call_playrecord](#)).

- **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.

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>
</web_service>
```

end_overlay

Completion event for overlay ([call_overlay](#)).

- **transaction_id**
- **reason**
 - "stopped"
 - "max-time"
- **duration** - in milliseconds.

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="stopped" />
    <event_data name="transaction_id" value="0974c8b5-8a3c-4a8e-ae82-8f7c8bd0efdd"
/>
  </event>
</web_service>
```

end_send_message

Completion event for send_message ([send_message](#)).

- **media_id**
- **call_id**
- **reason**
 - "success"
 - "failure"
- **result**
- **duration** - in milliseconds.

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>
</web_service>
```

end_dtmf

Completion event for send_dtmf ([send_dtmf](#)).

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

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>
</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.

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>
</web_service>
```

ringing

The remote party of an outbound call is ringing.

- **call_id**

Event Payload Example

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

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)

Event Payload Example

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

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.

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>
</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**

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>
</web_service>
```

info

Unsolicited user information (for example, SIP INFO).

- **call_id**
- **content_type**
- **content**

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>
</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**

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>
</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"

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>
</web_service>
```

alarm

Generic alarm notification.

- **call_id**
- **alarm**
 - "rtp-timeout"
 - "rtcp-timeout"
- **state**
 - "on"
 - "off"

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>
</web_service>
```

stream

RTP streaming status notification.

- **call_id**
- **state**
 - "started"
 - "stopped"

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>
</web_service>
```

prack

Provisional Acknowledgement received on an accepted *inbound* call.

- **call_id**

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>
</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)

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>
</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)

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>
</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.

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>
</web_service>
```

speech_marker

- **mrcp_id**
- **id**
- **transaction_id**
- **name**
- **timestamp**

Event Payload Example

```
<web_service version="1.0">
  <event type="speech_marker" 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="name" value="xyz" />
    <event_data name="timestamp" value="2014-2-2 1900:00" />
  </event>
</web_service>
```

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**

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="content" value="data" />
    <event_data name="content_type" value="plain/text" />
  </event>
</web_service>
```

start_of_input

Input detected event for [mrcp_recognize](#).

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

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>
</web_service>
```

Conference Events

active_talker

- **conf_id**
- **talkers** - space delimited list of call IDs.

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>
</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>
```

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 details on 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"

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.
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="c87fccal-b2d0-49c5-8b89-baaaae71cf695" 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="c87fccal-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="c87fccal-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="c87fccal-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="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: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>

<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:restriction>
</xs:simpleType>

<xs:simpleType name="video_type_option">
  <xs:restriction base="xs:string">
    <xs:enumeration value="video/x-vid" />
    <xs:enumeration value="image/jpeg" />
  </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="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="image/jpeg" />
  </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: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)"/>
  </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:restriction>
</xs:simpleType>

```

```

        <xs:enumeration value="end_recognize" />
        <xs:enumeration value="accepted" />
        <xs:enumeration value="answered" />
        <xs:enumeration value="updated" />
        <xs:enumeration value="active_talker" />
        <xs:enumeration value="alarm" />
        <xs:enumeration value="prack" />
        <xs:enumeration value="stream " />
        <xs:enumeration value="message" />
        <xs:enumeration value="conf overlay expired " />
        <xs:enumeration value="end_send_message " />
        <xs:enumeration value="speech_marker" />
    </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="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:choice>
        </xs:complexType>
    </xs:element>

```

```

<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="get_last_event" />
</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="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="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="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:complexType>
</xs:element>

<xs:element name="unjoin">
    <xs:complexType>
        <xs:attribute name="call_id" type="xs:string" />
    </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="send_info">
    <xs:complexType>
        <xs:attribute name="content_type" type="xs:string" use="required"/>
        <xs:attribute name="content" type="xs:string" use="required"/>
    </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: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:element ref="msg_payload" minOccurs="0"
maxOccurs="unbounded"/>
        </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="headers" type="xs:string" />
        <xs:attribute name="transaction_id" type="xs:string" />
    </xs:complexType>
</xs:element>

<xs:element name="send_info_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="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: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: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: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:complexType>
</xs:element>

```

```

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

<xs:element name="mrccp action">
    <xs:complexType>
        <xs:choice minOccurs="1" maxOccurs="1">
            <xs:element ref="speak" />
            <xs:element ref="recognize" />
            <xs:element ref="mrccp-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: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="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" />
            <xs:attribute name="base_audio_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" />
            <xs:attribute name="base_video_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:complexType>
    </xs:element>

```

```

        <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:element ref="call_action" minOccurs="0" maxOccurs="1" />
        </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="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" type="xs:string" />
    </xs:complexType>
    <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="headers" type="xs:string" />
</xs:complexType>
</xs:element>
<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" />
        <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="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: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="eventssubscribe" minOccurs="1"
maxOccurs="unbounded" />
        </xs:sequence>

```

```

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

    <xs:element name="eventhandler response">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="eventssubscribe" 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="eventssubscribe">
        <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: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 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: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. Dynamic Text and Image Generation

In RESTful mode, PowerMedia XMS uses ImageMagick libraries (<http://www.imagemagick.org>) to enable images and captions to be created dynamically by the application. This simplifies the task of building menus and other static images. Instead of having to create every image in advance and maintain them as the application evolves, a template for image creation is used, and text and images plugged into the template. This feature may also be used to create graphics for simple presentations and games.

Images are built from three data sources:

- A predefined document template, stored on PowerMedia XMS
- A predefined style definition, stored on PowerMedia XMS
- Dynamic content data provided by the application, via the PowerMedia XMS RESTful API, at run-time

PowerMedia XMS combines these to create and display a JPEG image at runtime.

Document Templates

Templates are used to define the structural elements that make up the image.

Template files are stored in the directory: `/etc/xms/imagemaker.d` and have names ending with the extension `.conf`. PowerMedia XMS will load such named files in this directory at start up. These files use the familiar "ini" file format and may contain multiple template definitions. Each section defines a single template with the name of the section serving as a globally unique name.

Note: The PowerMedia XMS Admin Console cannot be used to modify the `.conf` files. You will need to SSH into the system to make any modifications ("root" user with password "powermedia" or whatever you may have changed the root password to).

The following is an example template that defines a basic menu:

```
;-----  
; Example ImageMaker document templates.  
; This file should be placed into the imagemaker.d directory.  
;-----  
;  
; Each section defines one document template.  
;  
;-----  
; | Parameter | Value      | Description |  
; |-----+-----+-----+  
; | style     | filename   | File containing style information for the doc. |  
; |-----+-----+-----+  
; | text      | Unique id  | A text element |  
; |-----+-----+-----+  
; | list      | Unique id  | A list element (text) |  
; |-----+-----+-----+  
; | image     | Unique id  | An image element |  
; |-----+-----+-----+  
;  
;  
; The following values are reserved and must not be used for  
; element ids: id document  
;-----  
;  
; A basic menu. Three elements: A title, list of options and a footer.  
[menu]  
style = /var/lib/xms/imagemaker/style/menu.style  
text = header  
list = items  
text = footer  
; End of template
```

Style Definition

The appearance of the template's elements is defined using a separate style document, which is located in the directory: `/var/lib/xms/imagemaker/style`. A single style document may be shared by multiple templates. Style definitions for unknown elements are simply ignored. This behavior allows several templates to use a single style document and provide a common theme.

Rather than simply setting a background color for a style, the style file may specify an image file to be used as the background for the image being generated. This allows a common theme or logo to be included in each.

The following is an example of a CIF resolution JPEG background file:



The image generator supports many image formats including PNG. Using the PNG format allows layering effects to be achieved through the use of transparency in background images. All image files used to compose the output image are stored in the directory: `/var/lib/xms/imagemaker/images`.

The following is an example style document used for the menu defined earlier in this section:

Note: All values given in the style file are in pixels. Different resolutions are made up of a different arrays of pixels (CIF=352x288, VGA=640x480 and 720p=1280x720); therefore, a different style file with different values is required for each resolution. In addition, the aspect ratio (height to width proportion) also differs among resolutions and needs to be considered when defining layout.

```

; -----
; Style information for a full-screen menu.
;
; This file consists of a section named 'document' that defines the main
; style for the doc. This followed by sections that specify style information
; for the corresponding element in the document that this style is applied ; ; to.
; Colors may be defined as: a css hex value, a css color name or transparent.
; See http://www.w3.org/TR/SVG/types.html#ColorKeywords for color names.
;
; height          height of document or element in pixels (CIF is 288).
; width           width of document or element in pixels (CIF is 352).
; top             offset of top of an element relative to the document.
; left            offset of left of an element relative to the document.
; padding         offset of the content relative to top and left.
; background-image filename of an optional background image for the
;                 document.
; background-color color used to fill the document or element background.
; color           color used for text.
; border-radius   corner radius in pixels.
; border-color    color used for element border.
; border-width    element border width in pixels.
; text-align      justification of the text (left, center, right).
; font-size       size of the text font (small, medium, large).
; -----
[document]
height = 288
width = 352
background-image = /var/lib/xms/imagemaker/images/dialogic_background.png
background-color = #162660
color = #fff
[header]
height = 36
width = 328
top = 12
left = 12
padding = 4
background-color = aliceblue
color = #003
border-radius = 6
text-align = center
font-size = large

[items]
height = 184
width = 328
top = 60
left = 12
padding = 12
background-color = transparent
color = #fff
text-align = left
font-size = medium

[footer]
height = 20
width = 328
top = 260
left = 12
padding = 2
background-color = transparent
color = #9999a8
text-align = center
font-size = small

; End of style file
; -----

```

Creating the Image using the RESTful API

A PowerMedia XMS RESTful application is able to use the dynamic image creation feature with the following functions:

- play
- playcollect
- playrecord
- overlay

The image contents are specified as a single URI string that is prefixed by "image:", followed by a number of ampersand (&) separated parameters.

Each parameter is defined as a name followed by an equal sign (=) followed by a value. The first parameter uses the reserved name 'id' which specifies the name of the template being used. The remaining parameters specify template element names and their respective contents.

With the exception of the list element type, all elements take only a single value. The list of elements may be defined multiple times in order to define each item.

If a parameter value needs to contain ampersand or equal characters, these must be passed as percent escaped hex values, for example, an ampersand is represented by "%26".

An example of the URI used to create a simple menu is shown below. It contains a header, five (5) items and a footer.

```
uri="image:id=menu&header=Today's Menu&items=1 Salad&items=2 Chips&items=3 Burger&items=4  
Pizza&items=5 Panini&items=7 Pasta&footer=Tuesday"
```

When using XMSTool to drive an XMS application, (refer to [XMSTool RESTful Utility](#)) the ampersand (&) needs to be specified as "&". This same URI syntax should then appear as follows:

```
uri="image:id=menu&amp;header=Today's Menu&amp;items=1 Salad&amp;items=2 Chips&amp;items=3 Burger&amp;items=4  
Pizza&amp;items=5 Panini&amp;items=7 Pasta&amp;footer=Tuesday"
```

Overlay and the play/playcollect/playrecord commands are different in that the overlay command is used to superimpose the image on an existing media stream such as a mixed conference. With play/playcollect/playrecord, the defined image itself is used as basis for the video stream.

Note: Currently, only a single URI can be played for both audio and video. If an "image" URI is played, then any audio meant to accompany it will be lost. This restriction is planned to be lifted in a future release.

Combining the Three Data Sources

Combining the template file, style file, background image, and dynamic text representation results in the following menu:



There are several examples of different images included with PowerMedia XMS. Start with a working image and experiment/adjust until the desired results are obtained.

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. When selecting appropriate values for the <audio> or <video> element format attribute, refer to the following table.

Media File Formats

	Play	Record	Default File Extension
Audio Container	WAV	WAV	*.wav
	AUD	AUD	*.aud
	Raw (headerless)	Raw (headerless)	*.pcm
	AMR	AMR	*.amr
	AMR-WB	AMR-WB	*.awb
Video Container	VID	VID	*.vid
	3GP	3GP	*.3gp

Audio Play

Audio Play - WAV

Codec	Sample Size	Sample Rate	RESTful Attribute format	RESTful Attribute format codecs	RESTful Attribute audiosamplesize	RESTful Attribute audiosamplerate	XMS MIME Type	XMS MIME Parameters
Linear	8 16	8000 11025 16000	audio/x-wav audio/wav audio/vnd.wave	n/a	n/a	n/a	audio/x-wav	n/a
Alaw	8	8000	audio/x-wav audio/wav audio/vnd.wave	n/a	n/a	n/a	audio/x-wav	n/a
Mulaw	8	8000	audio/x-wav audio/wav audio/vnd.wave	n/a	n/a	n/a	audio/x-wav	n/a

Audio Play - Dialogic AUD (proprietary)

Codec	Sample Size	Sample Rate	RESTful Attribute format	RESTful Attribute format codecs	RESTful Attribute audiosamplesize	RESTful Attribute audiosamplerate	XMS MIME Type	XMS MIME Parameters
Linear	8 16	8000 11025 16000	audio/proprietary audio/x-aud	n/a	n/a	n/a	audio/x-aud	n/a
Alaw	8	8000	audio/proprietary audio/x-aud	n/a	n/a	n/a	audio/x-aud	n/a
Mulaw	8	8000	audio/proprietary audio/x-aud	n/a	n/a	n/a	audio/x-aud	n/a

Codec	Sample Size	Sample Rate	RESTful Attribute format	RESTful Attribute format codecs	RESTful Attribute audiosamplesize	RESTful Attribute audiosamplerate	XMS MIME Type	XMS MIME Parameters
AMR-NB (4.75k)	n/a	n/a	audio/proprietary audio/x-aud	n/a	n/a	n/a	audio/x-aud	n/a
AMR-NB (5.15k)	n/a	n/a	audio/proprietary audio/x-aud	n/a	n/a	n/a	audio/x-aud	n/a
AMR-NB (5.90k)	n/a	n/a	audio/proprietary audio/x-aud	n/a	n/a	n/a	audio/x-aud	n/a
AMR-NB (6.70k)	n/a	n/a	audio/proprietary audio/x-aud	n/a	n/a	n/a	audio/x-aud	n/a
AMR-NB (7.40k)	n/a	n/a	audio/proprietary audio/x-aud	n/a	n/a	n/a	audio/x-aud	n/a
AMR-NB (7.95k)	n/a	n/a	audio/proprietary audio/x-aud	n/a	n/a	n/a	audio/x-aud	n/a
AMR-NB (10.20k)	n/a	n/a	audio/proprietary audio/x-aud	n/a	n/a	n/a	audio/x-aud	n/a
AMR-NB (12.20k)	n/a	n/a	audio/proprietary audio/x-aud	n/a	n/a	n/a	audio/x-aud	n/a
AMR-WB (6.6k)	n/a	n/a	audio/proprietary audio/x-aud	n/a	n/a	n/a	audio/x-aud	n/a
AMR-WB (8.85k)	n/a	n/a	audio/proprietary audio/x-aud	n/a	n/a	n/a	audio/x-aud	n/a
AMR-WB (12.65k)	n/a	n/a	audio/proprietary audio/x-aud	n/a	n/a	n/a	audio/x-aud	n/a
AMR-WB (14.25k)	n/a	n/a	audio/proprietary audio/x-aud	n/a	n/a	n/a	audio/x-aud	n/a
AMR-WB (15.85k)	n/a	n/a	audio/proprietary audio/x-aud	n/a	n/a	n/a	audio/x-aud	n/a
AMR-WB (18.25k)	n/a	n/a	audio/proprietary audio/x-aud	n/a	n/a	n/a	audio/x-aud	n/a
AMR-WB (19.85k)	n/a	n/a	audio/proprietary audio/x-aud	n/a	n/a	n/a	audio/x-aud	n/a
AMR-WB (23.05k)	n/a	n/a	audio/proprietary audio/x-aud	n/a	n/a	n/a	audio/x-aud	n/a
AMR-WB (23.85k)	n/a	n/a	audio/proprietary audio/x-aud	n/a	n/a	n/a	audio/x-aud	n/a

Audio Play - Raw (headerless)

Codec	Sample Size	Sample Rate	RESTful Attribute format	RESTful Attribute format codecs	RESTful Attribute audiosamplesize	RESTful Attribute audiosamplerate	XMS MIME Type	XMS MIME Parameters
Linear	8 16	8000 11025 16000	audio	linear, pcm	8 16	8 11 16	audio/L8 audio/L16	rate=8000 rate=11025 rate=16000

Codec	Sample Size	Sample Rate	RESTful Attribute format	RESTful Attribute format codecs	RESTful Attribute audiosamplesize	RESTful Attribute audiosamplerate	XMS MIME Type	XMS MIME Parameters
Alaw	8	8000	audio	alaw	8	8	audio/x-alaw-basic audio/PCMA	n/a
Mulaw	8	8000	audio	mulaw	8	8	audio/basic audio/PCMU	n/a

Audio Play - AMR

Codec	Sample Size	Sample Rate	RESTful Attribute format	RESTful Attribute format codecs	RESTful Attribute audiosamplesize	RESTful Attribute audiosamplerate	XMS MIME Type	XMS MIME Parameters
AMR	n/a	n/a	audio/amr	n/a	n/a	n/a	audio/AMR	n/a

Audio Play - AMR-WB

Codec	Sample Size	Sample Rate	RESTful Attribute format	RESTful Attribute format codecs	RESTful Attribute audiosamplesize	RESTful Attribute audiosamplerate	XMS MIME Type	XMS MIME Parameters
AMR-WB	n/a	n/a	audio/amr-wb	n/a	n/a	n/a	audio/AMR-WB	n/a

Audio Play – 3GP

Codec	Sample Size	Sample Rate	RESTful Attribute format	RESTful Attribute format codecs	RESTful Attribute audiosamplesize	RESTful Attribute audiosamplerate	XMS MIME Type	XMS MIME Parameters
AMR	n/a	n/a	audio/3gpp*	n/a	n/a	n/a	audio/3gpp	n/a
AMR-WB	n/a	n/a	audio/3gpp*	n/a	n/a	n/a	audio/3gpp	n/a

* To play a file with audio and video tracks, specify video/3gpp.

Audio Record

Audio Record - WAV

Codec	Sample Size	Sample Rate	RESTful Attribute format	RESTful Attribute format codecs	RESTful Attribute audiosamplesize	RESTful Attribute audiosamplerate	XMS MIME Type	XMS MIME Parameters
Linear	8 16	8000 11025 16000	audio/x-wav audio/wav audio/vnd.wave	linear, pcm	8 16	8 11 16	audio/x-wav	codec=L8 rate=8000 codec=L8 rate=11025 codec=L8 rate=16000 codec=L16 rate=8000 codec=L16 rate=11025 codec=L16 rate=16000
Alaw	8	8000	audio/x-wav audio/wav audio/vnd.wave	alaw	8	n/a	audio/x-wav	codec=alaw rate=8000
Mulaw	8	8000	audio/x-wav audio/wav audio/vnd.wave	mulaw	8	n/a	audio/x-wav	codec=mulaw rate=8000

Audio Record - Dialogic AUD (proprietary)

Codec	Sample Size	Sample Rate	RESTful Attribute format	RESTful Attribute format codecs	RESTful Attribute audiosamplesize	RESTful Attribute audiosamplerate	XMS MIME Type	XMS MIME Parameters
Linear	8 16	8000 11025 16000	audio/proprietary audio/x-aud	linear, pcm	8 16	8 11 16	audio/x-aud	codec=L8 rate=8000 codec=L8 rate=11025 codec=L8 rate=16000 codec=L16 rate=8000 codec=L16 rate=11025 codec=L16 rate=16000
Alaw	8	8000	audio/proprietary audio/x-aud	alaw	8	8	audio/x-aud	codec=alaw rate=8000
Mulaw	8	8000	audio/proprietary audio/x-aud	mulaw	8	8	audio/x-aud	codec=mulaw rate=8000

Codec	Sample Size	Sample Rate	RESTful Attribute format	RESTful Attribute format codecs	RESTful Attribute audiosamplesize	RESTful Attribute audiosamplerate	XMS MIME Type	XMS MIME Parameters
AMR-NB (4.75k)	n/a	n/a	audio/proprietary audio/x-aud	amrnb_4_75k	n/a	n/a	audio/x-aud	codec=AMR mode=0
AMR-NB (5.15k)	n/a	n/a	audio/proprietary audio/x-aud	amrnb_5_15k	n/a	n/a	audio/x-aud	codec=AMR mode=1
AMR-NB (5.90k)	n/a	n/a	audio/proprietary audio/x-aud	amrnb_5_90k	n/a	n/a	audio/x-aud	codec=AMR mode=2
AMR-NB (6.70k)	n/a	n/a	audio/proprietary audio/x-aud	amrnb_6_70k	n/a	n/a	audio/x-aud	codec=AMR mode=3
AMR-NB (7.40k)	n/a	n/a	audio/proprietary audio/x-aud	amrnb_7_40k	n/a	n/a	audio/x-aud	codec=AMR mode=4
AMR-NB (7.95k)	n/a	n/a	audio/proprietary audio/x-aud	amrnb_7_95k	n/a	n/a	audio/x-aud	codec=AMR mode=5
AMR-NB (10.20k)	n/a	n/a	audio/proprietary audio/x-aud	amrnb_10_20k	n/a	n/a	audio/x-aud	codec=AMR mode=6
AMR-NB (12.20k)	n/a	n/a	audio/proprietary audio/x-aud	amrnb_12_20k	n/a	n/a	audio/x-aud	codec=AMR mode=7
AMR-WB (6.6k)	n/a	n/a	audio/proprietary audio/x-aud	amrwb_6_6k	n/a	n/a	audio/x-aud	codec=AMR-WB mode=0
AMR-WB (8.85k)	n/a	n/a	audio/proprietary audio/x-aud	amrwb_8_85k	n/a	n/a	audio/x-aud	codec=AMR-WB mode=1
AMR-WB (12.65k)	n/a	n/a	audio/proprietary audio/x-aud	amrwb_12_65k	n/a	n/a	audio/x-aud	codec=AMR-WB mode=2
AMR-WB (14.25k)	n/a	n/a	audio/proprietary audio/x-aud	amrwb_14_25k	n/a	n/a	audio/x-aud	codec=AMR-WB mode=3
AMR-WB (15.85k)	n/a	n/a	audio/proprietary audio/x-aud	amrwb_15_85k	n/a	n/a	audio/x-aud	codec=AMR-WB mode=4
AMR-WB (18.25k)	n/a	n/a	audio/proprietary audio/x-aud	amrwb_18_25k	n/a	n/a	audio/x-aud	codec=AMR-WB mode=5
AMR-WB (19.85k)	n/a	n/a	audio/proprietary audio/x-aud	amrwb_19_85k	n/a	n/a	audio/x-aud	codec=AMR-WB mode=6
AMR-WB (23.05k)	n/a	n/a	audio/proprietary audio/x-aud	amrwb_23_05k	n/a	n/a	audio/x-aud	codec=AMR-WB mode=7
AMR-WB (23.85k)	n/a	n/a	audio/proprietary audio/x-aud	amrwb_23_85k	n/a	n/a	audio/x-aud	codec=AMR-WB mode=8

Audio Record - Raw (headerless)

Codec	Sample Size	Sample Rate	RESTful Attribute format	RESTful Attribute format codecs	RESTful Attribute audiosamplesize	RESTful Attribute audiosamplerate	XMS MIME Type	XMS MIME Parameters
Linear	8 16	8000 11025 16000	audio	linear, pcm	8 16	8 11 16	audio/L8 audio/L16	rate=8000 rate=11025 rate=16000

Alaw	8	8000	audio	alaw	8	8	audio/x-alaw-basic audio/PCMA	n/a
Mulaw	8	8000	audio	mulaw	8	8	audio/basic audio/PCMU	n/a

Audio Record - AMR

Codec	Sample Size	Sample Rate	RESTful Attribute format	RESTful Attribute format codecs	RESTful Attribute audiosamplesize	RESTful Attribute audiosamplerate	XMS MIME Type	XMS MIME Parameters
AMR	n/a	n/a	audio/amr	n/a	n/a	n/a	audio/AMR	mode=0-7 (default is 7)

Audio Record - AMR-WB

Codec	Sample Size	Sample Rate	RESTful Attribute format	RESTful Attribute format codecs	RESTful Attribute audiosamplesize	RESTful Attribute audiosamplerate	XMS MIME Type	XMS MIME Parameters
AMR-WB	n/a	n/a	audio/amr-wb	n/a	n/a	n/a	audio/AMR-WB	mode=0-8 (default is 8)

Audio Record – 3GP

Codec	Sample Size	Sample Rate	RESTful Attribute format	RESTful Attribute format codecs	RESTful Attribute audiosamplesize	RESTful Attribute audiosamplerate	XMS MIME Type	XMS MIME Parameters
AMR	n/a	n/a	audio/3gpp	n/a	n/a	n/a	audio/3gpp	codec=amr mode=0-7 (default is 7)
AMR-WB	n/a	n/a	audio/3gpp	n/a	n/a	n/a	audio/3gpp	codec=amr-wb mode=0-8 (default is 8)

Video Play

Video Play - Dialogic VID (proprietary)

Codec	RESTful Attribute format	RESTful Attribute format codecs	XMS MIME Type	XMS MIME Parameters
h263	video/proprietary	n/a	video/x-vid	n/a
h263-1998	video/proprietary	n/a	video/x-vid	n/a
h264	video/proprietary	n/a	video/x-vid	n/a
mp4v_es	video/proprietary	n/a	video/x-vid	n/a
jpeg	image/jpeg	jpeg	image/jpeg	n/a

Video Play – 3GP

Codec	RESTful Attribute format	RESTful Attribute format codecs	XMS MIME Type	XMS MIME Parameters
h263	video/3gpp	n/a	video/3gpp	n/a
h263-1998	video/3gpp	n/a	video/3gpp	n/a
h264	video/3gpp	n/a	video/3gpp	n/a

Video Record

Video Record - Dialogic VID (proprietary)

Video Codec	Resolution	Codecs	RESTful Attribute profile	RESTful Attribute level	RESTful Attribute maxbitrate	RESTful Attribute framerate	RESTful Attribute imagewidth	RESTful Attribute imageheight
H.263 / H.263-1998	CIF	h263	0	10	128000	10	352	288
	CIF		0	20	128000	15	352	288
	CIF		0	30	384000	30	352	288
	QCIF		0	10	128000	15	176	144
	QCIF		0	20	128000	30	176	144
	n/a	native	n/a	n/a	n/a	n/a	n/a	n/a
H.264	CIF	h264	0	1.2	384000	15	352	288
	CIF		0	1.3	384000	30	352	288
	CIF		0	1.3	768000	30	352	288
	CIF		0	2.0	384000	30	352	288
	CIF		0	2.1	384000	30	352	288
	CIF		0	2.2	768000	25	352	288
	HD720p		0	3.1	2000000	15	1280	720
	QCIF		0	1.0	40000	15	176	144
	QCIF		0	1.1	128000	15	176	144
	QCIF		0	1.1	192000	30	176	144
	VGA		0	1.2	384000	15	640	480
	VGA		0	3.0	768000	25	640	480
	n/a	native	n/a	n/a	n/a	n/a	n/a	n/a
H.264 supports Baseline Profile (i.e., 66). This combined with level is used to set the profile_level_id. The level must be entered in "x.x" format (i.e., 10 and 1b are invalid).								
MPEG-4	CIF	mp4v-es	0	2	128000	15	352	288

Video Codec	Resolution	Codecs	RESTful Attribute profile	RESTful Attribute level	RESTful Attribute maxbitrate	RESTful Attribute framerate	RESTful Attribute imagewidth	RESTful Attribute imageheight
	CIF		0	3	384000	15	352	288
	QCIF		0	0	44000	15	176	144
	QCIF		0	1	64000	15	176	144
	QVGA		0	3	384000	30	320	240
	VGA		0	4	800000	30	640	480
	n/a	native	n/a	n/a	n/a	n/a	n/a	n/a

MPEG-4 supports Simple Profile (i.e., SP3).

Video Record - 3GP

Video Codec	Resolution	Codecs	MSML Attribute profile	MSML Attribute level	MSML Attribute maxbitrate	MSML Attribute framerate	MSML Attribute imagewidth	MSML Attribute imageheight
H.263 / H.263-1998	CIF	h263	0	10	128000	10	352	288
	CIF		0	20	128000	15	352	288
	CIF		0	30	384000	30	352	288
	QCIF		0	10	128000	15	176	144
	QCIF		0	20	128000	30	176	144
	n/a	native	n/a	n/a	n/a	n/a	n/a	n/a
H.264	CIF	h264	0	1.2	384000	15	352	288
	CIF		0	1.3	384000	30	352	288
	CIF		0	1.3	768000	30	352	288
	CIF		0	2.0	384000	30	352	288
	CIF		0	2.1	384000	30	352	288
	CIF		0	2.2	768000	25	352	288
	HD720p		0	3.1	2000000	15	1280	720
	QCIF		0	1.0	40000	15	176	144
	QCIF		0	1.1	128000	15	176	144
	QCIF		0	1.1	192000	30	176	144
	VGA		0	1.2	384000	15	640	480
	VGA		0	3.0	768000	25	640	480
	n/a	native	n/a	n/a	n/a	n/a	n/a	n/a

H.264 supports Baseline Profile (i.e., 66). This combined with level is used to set the profile_level_id. The level must be entered in "x.x" format (i.e., 10 and 1b are invalid).