

Network Working Group
Internet Draft
Intended status: Informational
Expires: July 2014

A. Saleem
S. Dunn
Radisys
January 2, 2014

MSML Package for the Media Control Channel Framework
draft-saleem-mediactrl-msml-package-03.txt

Abstract

The Media Server Markup Language [[RFC5707](#)] is used to control and invoke many different types of services on IP media servers. MSML can be used, for example, to control media server conferencing features such as video layout and audio mixing, create sidebar conferences or personal mixes, and set the properties of media streams. As well, clients can use MSML to define media processing dialogs, which may be used as parts of application interactions with users or conferences. This document describes the use of MSML [[RFC5707](#)] language used within the context of Media Control Channel Framework [[RFC6230](#)]. The use of MSML [[RFC5707](#)] is described here as a standalone package for use within and compliant with the Media Control Channel Framework [[RFC6230](#)].

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 1, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction.....	2
2.	Control Package Definition.....	3
2.1.	Control Package Name.....	3
2.2.	Framework Message Usage.....	3
2.3.	Common XML Support.....	4
2.4.	CONTROL Message Body.....	4
2.5.	REPORT Message Body.....	5
2.6.	Audit.....	5
2.7.	Examples.....	6
2.7.1.	Audio Conference with Active Speaker Notification....	6
3.	Element Definitions.....	7
4.	Formal Syntax.....	7
5.	Security Considerations.....	8
6.	IANA Considerations.....	8
6.1.	Control Package Registration.....	8
6.2.	URN Sub-Namespace Registration.....	8
6.3.	XML Schema Registration.....	8
7.	References.....	8
7.1.	Normative References.....	8
7.2.	Informative References.....	9
8.	Acknowledgments.....	9

[1. Introduction](#)

The architectural framework for media server control is described in [\[RFC5567\]](#) and forms the basis for Media Control Channel Framework [\[RFC6230\]](#). In this framework, SIP is used by application servers to both terminate media streams on media servers and to create and manage media server control channels between themselves and media servers.

The SIP based control channel framework is described in [\[RFC6230\]](#). It describes the establishment (via SDP negotiation), use, and termination of a reliable transport (TCP) connection between

application server and media server for transport of media server control messages. The "COMEDIA" specification [[RFC4145](#)] for setting up and maintaining reliable connections is used as part of the negotiation mechanism.

Control packages define a set of request, response, and notification messages that can be sent over a SIP control channel for managing services on a media server. The set of control packages supported by both the application server and the media server is negotiated as part of the control channel set up.

This document extends the Media Control Channel Framework [[RFC6230](#)] by defining the MSML Package referred to in [Section 3.2](#) of Media Server Markup Language [[RFC5707](#)].

2. Control Package Definition

This section fulfills the mandatory requirements detailed in [Section 8 of \[RFC6230\]](#) that MUST be specified in the definition of a new Control Framework Package. The Control Framework Package described in the subsequent sections MUST be functionality compliant with Media Server Markup Language, as described in detail in [[RFC5707](#)]. The XML schema and all features and functions of this package MUST be compliant to [[RFC5707](#)].

[2.1.](#) Control Package Name

The Control Framework requires a Control Package to specify and register a unique name with IANA.

The name of this Control Package is "msml/1.1" (Media Server Markup Language [[RFC5707](#)], version 1.1). The IANA registration of this package name is specified in [section 7](#).

Due to historic legacy versions of MSML IETF drafts leading up to [[RFC5707](#)] had originally started with version 1.0, this package MUST be compliant to [[RFC5707](#)] which is associated with version 1.1 of the Media Server Markup Language.

[2.2.](#) Framework Message Usage

This section details the Framework messages that can be used over the established Media Control Framework channel as per [6230].

The MSML Package supports the CONTROL method for a Control Client to send a request, response, or event message and the REPORT method for a Control Server to send updates and final responses.

The message bodies shall consist of MIME media types. The messages consist of XML encoded elements and attributes for requests, responses, and event notifications as defined in [[RFC5707](#)] and are contained within the root element <msml>.

This package defines a client server interface where the media server acts as the server and typically application servers (or other network elements such as SIP endpoints) act as the clients. The client server interface defined by this package allows for many to one relationship between the clients and the media server. Additionally, this package also allows the media server to act as a client for certain transactions which require unsolicited event notifications, eg: DTMF digits or announcement play complete events, etc, to application servers.

The scope of the package is defined in [[RFC5707](#)] which covers a wide variety of applications including, multimedia announcements, IVR/IVVR, multimedia conferencing, and numerous other services supported by this package where ever media plane processing is required.

All transaction types defined by this package are independent of the transactions of the Media Control Framework itself, and are defined within [[RFC5707](#)].

[2.3.](#) Common XML Support

This package requires that the XML schema as defined in [[RFC5707](#)] MUST be supported. This XML schema defines all transactions from Clients to Media Server as well as event notifications from Media Server to Clients.

[2.4.](#) CONTROL Message Body

The Media Control Framework requires a Control Package to define the control body that can be contained within a CONTROL command request and to indicate the location of detailed syntax definitions and semantics for the appropriate body types.

This package defines the Control message bodies with the MIME media type with XML encoded root element of <msml>, as defined within [[RFC5707](#)]. All other XML encoded requests are contained with the <msml> root element, such as <dialogstart> or <createconference> etc. The application server can include any child elements of <msml> in the CONTROL messages and MUST NOT include <result> or <event> elements.

The 200 response to a CONTROL message from the media server MUST include only the <result> element

Media servers supporting this package MUST also support event notifications, as defined in [[RFC5707](#)], originated by the media server and contained within the CONTROL message. These event notifications are also MIME media type using XML encoding as defined they schema definition in [[RFC5707](#)].

The media server MUST only include the <event> child elements in a CONTROL message and the 200 response to a CONTROL message from the application server SHOULD have a zero length empty body.

[2.5. REPORT Message Body](#)

The Media Control Framework requires a Control Package definition to define the REPORT body that MAY be contained within a REPORT command request.

This package defines the use of REPORT message which MUST comply with the XML schema defined in [[RFC5707](#)]. The REPORT message MUST contain a single <msml> element containing child elements as defined by the XML schema.

The REPORT messages defined by this package MAY include any valid XML elements that can be in a 200 response to a CONTROL message as described in earlier section.

A 200 response to a REPORT message SHOULD have a zero length empty body.

[2.6. Audit](#)

This package MUST support the Audit request carried within the CONTROL message and the associated Audit response carried in the 200 response to the CONTROL message.

The Audit response alternately MAY be sent or carried within the REPORT message.

The Audit requests and responses MUST be XML encoded as specified in [[RFC5707](#)]. The <audit> element contained within the root <msml> element defines the audit request and subsequently the audit response generated by the media server is XML encoded as defined in [[RFC5707](#)].

2.7. Examples

This section provides examples of using the MSML Control Package. The examples assume a Media Control channel has been established and synced between application server (AS) and media server (MS) as per [\[RFC6230\]](#).

Additional examples of using MSML XML elements for conferencing, media processing dialogs, and auditing are provided in [\[RFC5707\]](#)

2.7.1. Audio Conference with Active Speaker Notification

An audio conference is created with a request for Active Speaker Notifications (ASN). A caller is joined to the conference. An ASN event is sent to the application server.

1. AS -> MS (Control Framework Message): CONTROL.

- create conference and join participant

```
CFW i387yeiqyiq CONTROL
Control-Package: msml/1.1
Content-Type: application/msml+xml
Content-Length: 202

<msml version="1.1">
<createconference name="9pi0d1287005361168169715739"
deletewhen="nocontrol" term="true">
<audiomix id="amix9pi0d1287005361168169715739"
samplerate="8000">
<n-loudest n="4"/>
<asn ri="10s" cvd:asth="-96"/>
</audiomix>
</createconference>
<join id1="conf:9pi0d1287005361168169715739"
id2="conn:10.29.162.235060+1+52030002+12ebfaa0">
<stream media="audio" dir="from-id1"/>
<stream media="audio" dir="to-id1">
<clamp dtmf="true"/>
</stream>
</join>
</msml>
```

2. MS -> AS (Control Framework Message): 200.

```
CFW i387yeiqyiq 200
Content-Type: application/msml+xml
Content-Length: 60
```

```
<msml version="1.1">
<result response="200">
</result>
</msml>
```

3. MS -> AS (Control Framework Message): CONTROL.

- Active speaker event

```
CFW abcdefghi CONTROL
Control-Package: msml/1.1
Content-Type: application/msml+xml
Content-Length: 101
```

```
<msml version="1.1">
<event name="msml.conf.asn"
id="conf:9pi0d1287005361168169715739">
<name>speaker</name>
<value>conn:10.29.162.235060+1+52030002+12ebfaa0</value>
</event>
</msml>
```

4. AS -> MS (Control Framework Message): 200.

```
CFW abcdefghi 200
```

3. Element Definitions

The XML elements for the MSML Package are described fully in [\[RFC5707\]](#).

4. Formal Syntax

The XML Schema for the MSML Package is described in [\[RFC5707\]](#).

5. Security Considerations

The security measures employed by the Channel Framework are described in [\[RFC6230\]](#).

Additional security considerations for MSML are described in [\[RFC5707\]](#).

6. IANA Considerations

6.1. Control Package Registration

This section registers a new Media Control Channel Framework package as per the instructions in [section 13.1 of \[RFC6230\]](#).

Package Name: msml/1.1

Published Specification: [RFC 5707](#)

Internet-Draft: MSML Package for the Media Control Channel Framework [\[draft-saleem-mediactrl-msml-package-01.txt\]](#)

Person & email address to contact for further information:

Adnan Saleem <adnan.saleem@radisys.com>

6.2. URN Sub-Namespace Registration

The URN Sub-Namespace Registration for the MSML Package of Media Control Channel is defined in [section 18.3 of \[RFC5707\]](#).

6.3. XML Schema Registration

The XML Schema Registration for the MSML Package of Media Control Channel is defined in [section 18.4 of \[RFC5707\]](#).

7. References

7.1. Normative References

[RFC5707] Saleem, A., Xin, Y., Sharratt, G., "Media Server Markup Language", [RFC 5707](#), February 2010

[RFC6230] Boulton, C., Melanchuk, T., and S. McGlashan, "Media Control Channel Framework", [RFC 6230](#), May 2011

7.2. Informative References

- [RFC4145] Yon, D. and G. Camarillo, "TCP-Based Media Transport in the Session Description Protocol (SDP)", [RFC 4145](#), September 2005.
- [RFC5567] Melanchuk, T., "An Architectural Framework for Media Server Control", [RFC 5567](#), June 2009.

8. Acknowledgments

This document was generated from initial inputs from Media Server Markup Language [[RFC5707](#)] and contributors to this RFC. Additional thanks for inputs and ideas on MSML MediaCtrl Package definition from Yong Xin, co-author of [[RFC5707](#)].

Authors' Addresses

Adnan Saleem
Radisys
[4190](#) Still Creek Drive, Suite 300
Burnaby, BC, V5C 6C6
Canada
Email : adnan.saleem@radisys.com

Steve Dunn
Radisys
[4190](#) Still Creek Drive, Suite 300
Burnaby, BC, V5C 6C6
Canada
Email : stephen.dunn@radisys.com