[MS-XMLMC]:

XML Schema for Media Control Extensions

Revision Summary

Date	Revision History	Revision Class	Comments
4/4/2008	0.1	New	Initial version
4/25/2008	0.2	Minor	Updated based on feedback
6/27/2008	1.0	Major	Updated and revised the technical content.
8/15/2008	1.01	Minor	Revised and edited the technical content.
12/12/2008	2.0	Major	Updated and revised the technical content
2/13/2009	2.01	Minor	Updated with latest template bug fixes (redlined)
3/13/2009	2.02	Minor	Revised and edited the technical content.
7/13/2009	2.03	Major	Revised and edited the technical content
8/28/2009	2.04	Editorial	Revised and edited the technical content
11/6/2009	2.05	Editorial	Revised and edited the technical content
2/19/2010	2.06	Editorial	Revised and edited the technical content
3/31/2010	2.07	Major	Updated and revised the technical content
4/30/2010	2.08	Editorial	Revised and edited the technical content
6/7/2010	2.09	Editorial	Revised and edited the technical content
6/29/2010	2.10	Editorial	Changed language and formatting in the technical content.
7/23/2010	2.10	None	No changes to the meaning, language, or formatting of the technical content.
9/27/2010	3.0	Major	Significantly changed the technical content.
11/15/2010	3.0	None	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	3.0	None	No changes to the meaning, language, or formatting of the technical content.
3/18/2011	3.0	None	No changes to the meaning, language, or formatting of the technical content.
6/10/2011	3.0	None	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	3.1	Minor	Clarified the meaning of the technical content.
4/11/2012	3.1	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	3.1	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	3.1	None	No changes to the meaning, language, or formatting of the technical content.
2/11/2013	3.1	None	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
7/30/2013	3.1	None	No changes to the meaning, language, or formatting of the technical content.
11/18/2013	3.1	None	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	3.1	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	3.2	Minor	Clarified the meaning of the technical content.
7/31/2014	3.2	None	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	3.2	None	No changes to the meaning, language, or formatting of the technical content.
3/30/2015	4.0	Major	Significantly changed the technical content.
9/4/2015	4.0	None	No changes to the meaning, language, or formatting of the technical content.
7/15/2016	4.0	None	No changes to the meaning, language, or formatting of the technical content.
9/14/2016	4.0	None	No changes to the meaning, language, or formatting of the technical content.
4/27/2018	5.0	Major	Significantly changed the technical content.
7/24/2018	6.0	Major	Significantly changed the technical content.
8/28/2018	7.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction

This document specifies the XML Schema for Media Control Extensions Protocol. This protocol is a proprietary extension to an Internet-Draft Proposal entitled "XML Schema for Media Control," which is described in [IETFDRAFT-XMLSMC-12]. [IETFDRAFT-XMLSMC-12] describes media control messages for Session Initiation Protocol (SIP)-based systems that send or receive video using the Real-Time Transport Protocol (RTP).

This protocol extends [IETFDRAFT-XMLSMC-12] by adding one new media control command instructing a sender to stop or suspend transmission of real-time video streams during a multimedia session.

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

Central Video Processor (CVP): An entity that centrally processes multiple received video streams and distributes the resulting, processed streams to the parties in a multiparty video conference. An example of a CVP is a video Multipoint Control Unit (MCU).

Multipoint Control Unit (MCU): A server endpoint that offers mixing services for multiparty, multiuser conferencing. An MCU typically supports one or more media types, such as audio, video, and data.

Multipurpose Internet Mail Extensions (MIME): A set of extensions that redefines and expands support for various types of content in email messages, as described in [RFC2045], [RFC2046], and [RFC2047].

originating video source (OVS): An entity that locally produces a video stream and sends the video stream to another party or to a **Multipoint Control Unit (MCU)**. For example, a protocol client that is configured with a video camera.

primitive: A basic or fundamental message-based operation that is defined by a communications protocol.

Real-Time Transport Control Protocol (RTCP): A network transport protocol that enables monitoring of Real-Time Transport Protocol (RTP) data delivery and provides minimal control and identification functionality, as described in [RFC3550].

Real-Time Transport Protocol (RTP): A network transport protocol that provides end-to-end transport functions that are suitable for applications that transmit real-time data, such as audio and video, as described in [RFC3550].

Session Description Protocol (SDP): A protocol that is used for session announcement, session invitation, and other forms of multimedia session initiation. For more information see [MS-SDP] and [RFC3264].

Session Initiation Protocol (SIP): An application-layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants. **SIP** is defined in [RFC3261].

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as defined in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information.

[IETFDRAFT-XMLSMC-12] Levin, O., Even, R., and Hagendorf, P., "XML Schema for Media Control", draft-levin-mmusic-xml-media-control-12, November 2007, http://ietfreport.isoc.org/all-ids/draft-levin-mmusic-xml-media-control-12.txt

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.rfc-editor.org/rfc/rfc2119.txt

[RFC2976] Donovan, S., "The SIP INFO Method", RFC 2976, October 2000, http://www.rfc-editor.org/rfc/rfc2976.txt

[RFC3264] Rosenberg, J., and Schulzrinne, H., "An Offer/Answer Model with the Session Description Protocol (SDP)", RFC 3264, June 2002, http://www.rfc-editor.org/rfc/rfc3264.txt

[RFC4566] Handley, M., Jacobson, V., and Perkins, C., "SDP: Session Description Protocol", RFC 4566, July 2006, http://www.ietf.org/rfc/rfc4566.txt

1.2.2 Informative References

None.

1.3 Overview

The **Session Initiation Protocol (SIP)** and the **Session Description Protocol (SDP)** together describe the mechanisms and message syntax for establishing point-to-point multimedia sessions. Various media types are specified in SDP, including real-time video using the **Real-Time Transport Protocol (RTP)**. Separately, [RFC2976] section 1 describes the SIP INFO method as an extensibility mechanism for SIP. The SIP INFO method provides a general-purpose message transport for application-level information to be transferred within an existing SIP communication context. The SIP INFO method allows various message types to be transferred along the SIP signaling path, but does not specify message content or semantics.

[IETFDRAFT-XMLSMC-12] sections 4 and 5 specify message semantics and a schema for specific video media control messages to be carried in a SIP INFO method. Only the proprietary extensions to [IETFDRAFT-XMLSMC-12] are specified in this document.

This protocol is intended for use in the specific circumstance when video control messages are sent from a **Central Video Processor (CVP)**, such as a video **Multipoint Control Unit (MCU)**, to an **originating video source (OVS)**.

[IETFDRAFT-XMLSMC-12] section 9 describes a new **Multipurpose Internet Mail Extensions (MIME)** content type for Extensible Markup Language (XML)-encoded media control messages, and defines a schema for the message body. The message schema includes the hierarchical definition of a media control **primitive** type, a video control primitive type, and a video encoder control primitive type. Finally, the **picture_fast_update** element is defined as a video encoder control primitive type.

The **picture_fast_update** element is a media control message that requests the sender of an RTP video stream to send a full video frame update as soon as possible.

This protocol defines one new video encoder control primitive by adding the **picture_freeze** element to the XML schema. This element is a media control message that requests the sender of an RTP video stream to suspend, or stop, sending RTP video until further notice.

In multiparty video sessions where video is centrally switched by a CVP, there are typically many more video sources at any given moment than necessary. If all participants in a multiparty conference are sending video to the CVP and only a small number of these are actually distributed back to protocol clients, network bandwidth is unnecessarily consumed by the unused video streams. In addition to consuming network bandwidth, the unused streams also consume resources on the CVP, because the CVP continues to receive, and then discard, the unused video data.

A standard signaling mechanism for stopping and starting RTP streams, by changing directional attributes of **SDP**, is described in [RFC3264] section 8. Typical implementations of video-enabled applications using [RFC3264] assume application-specific behavior associated with changes in value of the directional attributes; for example, by closing local video preview windows and stopping video capture. This behavior is undesirable when the start/stop sequence is frequent and/or transient in nature.

This protocol provides the means for the receiver of video, or CVP, to request the sender to stop sending its video stream without changing any SIP session state. When the **picture_freeze** message defined in this protocol is used in conjunction with the existing **picture_fast_update** message, it provides a lightweight and low-latency signaling mechanism for pausing and re-starting real-time video streams.

1.4 Relationship to Other Protocols

This protocol depends on the Internet-Draft [IETFDRAFT-XMLSMC-12], which establishes the base schema for video control **primitive**s in **Session Initiation Protocol (SIP)**-based systems. [IETFDRAFT-XMLSMC-12] in turn depends on the SIP INFO Method extension, as described in [RFC2976]. This protocol relates specifically to **Session Description Protocol (SDP)** video media types, as described in [RFC4566].

1.5 Prerequisites/Preconditions

None.

1.6 Applicability Statement

This protocol is applicable only to protocol clients or servers participating in multiparty video sessions where video is centrally processed by a **Central Video Processor (CVP)**. The **picture_freeze** message is sent only by CVPs during the course of centralized multiparty conferences, and received **picture_freeze** messages are processed only by an **originating video source (OVS)**. This protocol is not applicable to generic point-to-point multimedia sessions and/or to media types other than **Real-Time Transport Protocol (RTP)** video.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

The payload for this protocol is carried as the message payload of the SIP INFO method, as specified in [RFC2976] section 3.

2.2 Message Syntax

The message schema on which this extension is based is defined in [IETFDRAFT-XMLSMC-12] section 5. The relevant sections of that schema are included in section 2.2.1 to show context. To view the full schema, see section 6.

This protocol adds one media control **primitive** to the media control primitives defined in [IETFDRAFT-XMLSMC-12] section 5, so one element representing this primitive is added to the schema defined in that same document.

2.2.1 picture_freeze

The **picture_freeze** message extends the schema with one element named "picture_freeze", as follows.

```
<xs:complexType name="to_encoder">
    <xs:choice>
        <xs:element name="picture fast update"/>
        <xs:element name="picture_freeze"/>
    </xs:choice>
</xs:complexType>
```

3 Protocol Details

3.1 Originating Video Source Details

This section describes the processing of received **picture_freeze** messages by an entity acting as an **originating video source (OVS)**. An OVS SHOULD NOT send the **picture_freeze** message.

3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

This protocol is stateless with respect to the sending of its messages. Because there is no synchronization of state between **picture_freeze** messages and the video media states established by the **Session Description Protocol (SDP)** offer/answer model, as specified in [RFC3264] section 4 and [RFC4566] section 4, an **originating video source (OVS)** MUST respond consistently to all SIP INFO messages that contain the **picture_freeze** message, regardless of what media control actions are taken, or not taken, at the time the individual messages are processed.

Section 3.1.5 describes the relationships between OVS behaviors and the receipt of the **picture_freeze** message.

3.1.2 Timers

None.

3.1.3 Initialization

No initialization is required, other than the constraints specified in [RFC2976] section 2, regarding establishment of a **Session Initiation Protocol (SIP)** session before a SIP INFO message can be transferred.

3.1.4 Higher-Layer Triggered Events

None.

3.1.5 Message Processing Events and Sequencing Rules

3.1.5.1 Processing a Received Picture_Freeze Message

An **originating video source (OVS)** that implements this protocol has only two possible courses of action to take when a **picture_freeze** message is received. It either suspends sending **Real-Time Transport Protocol (RTP)** frames containing video payloads, or it takes no action because the OVS is already in a state where no RTP video frames are being sent. This protocol extension is stateless, and thus subsequent to processing the received **picture_freeze** message according to the rules specified in this section, an OVS SHOULD NOT retain or persist any state related to the received **picture_freeze** message.

An OVS is not required to implement this protocol. However, an OVS that implements this protocol MUST also implement processing of the **picture_fast_update** message, as specified in [IETFDRAFT-XMLSMC-12] section 4. This is because the **picture_freeze** message is used to temporarily suspend

sending RTP video packets, and the **picture_fast_update** message is used to resume sending RTP video packets.

When an OVS processes a received **picture_freeze** message, it SHOULD suspend sending RTP video packets if it is currently sending them.

If an OVS does suspend sending RTP video packets in response to processing a received **picture_freeze** message, it MUST continue to send **Real-Time Transport Control Protocol (RTCP)** packets.

Note: The same behavior is recommended, but not mandated, by [RFC4566] section 6, specifically regarding the **a=recvonly** and **a=inactive** attributes for RTP-based systems.

- Recvonly applies to the media only, not to any associated control protocol. An RTP-based system in recvonly mode SHOULD still send RTCP packets.
- An RTP-based system SHOULD still send RTCP, even if it started inactive.

3.1.5.2 Error Cases

This protocol does not introduce any new potential error conditions in addition to those specified in [IETFDRAFT-XMLSMC-12] section 7.2, which defines the format for reporting a parsing error.

Recall that there is no expectation of a synchronization state between **picture_freeze** messages and any other messages, including the negotiated **Session Description Protocol (SDP)** media state that is carried in offer/answer messages, as specified in [RFC3264] section 6. For that reason, an **originating video source (OVS)** MUST NOT return any error messages in response to a well-formed **picture_freeze** message, regardless of the relevance of the **picture_freeze** message to the current state of the OVS application.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Central Video Processor Details

Central Video Processor (CVP) entities can send **picture_freeze** messages.

3.2.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

This protocol is stateless with respect to the flow of messages on the wire.

The implementation of the **Central Video Processor (CVP)** determines what internal state logic best represents its video-processing features. A CVP implementation that centrally processes **Real-Time Transport Protocol (RTP)** video media might or might not operate in modes where only a subset of received RTP video media streams are actually processed at any given moment. This protocol is intended to reduce network bandwidth in cases where the CVP implementation intentionally stops

processing a received RTP video media stream for a finite period of time. The decision to stop processing a received RTP stream is entirely in the domain of the CVP implementation.

Regardless of the implementation and how many received streams are processed at any one time, a typical implementation might arrive at a decision to begin processing one received stream at the expense of another. Although implementations can take other forms, the following example illustrates how a typical CVP implementation uses this protocol.

Consider a multipoint conference in progress, with the CVP processing received RTP video from multiple endpoints. A **Session Initiation Protocol (SIP)** session has been established between the CVP and each of the protocol client endpoints. At the point in time where this example begins, the CVP is processing received RTP video from only a subset of all endpoints, one of which is named "endpoint A." The CVP is not processing received RTP video from the remaining endpoints, regardless of whether or not those endpoints are sending RTP video packets. One of the remaining endpoints is named "endpoint B."

When the CVP implementation logic determines that it needs to begin, or resume, processing RTP video from endpoint B instead of endpoint A, the CVP sends the following two messages:

- A **picture_fast_update** message to endpoint B.
- A **picture_freeze** message to endpoint A.

The CVP can send the messages in any order, although a prudent implementation ensures that RTP video was, in fact, being received from endpoint B before sending the **picture_freeze** to endpoint A.

3.2.1.1 Stream-Id-Specific Forms of Video Control Primitives

The XML schema specified in [IETFDRAFT-XMLSMC-12] section 5 defines an optional **stream_id** element within the **vc_primitive** element. The text of [IETFDRAFT-XMLSMC-12] does not specify the semantics of the **stream_id** element.

This specification assumes an environment where the video is centrally processed (see section 1.6, Applicability Statement earlier) and that the video control messages are sent only by CVPs. In that environment, it is assumed that if the **Central Video Processor (CVP)** is receiving multiple video streams per protocol client, all of a protocol client's video streams will be started and/or stopped in unison.

Therefore, CVP implementations SHOULD NOT include the optional **stream_id** element when sending either the **picture_fast_update** or the **picture_freeze** message.

3.2.2 Timers

None.

3.2.3 Initialization

This protocol does not require any initialization, other than the constraints specified in [RFC2976] section 2, regarding establishment of a **Session Initiation Protocol (SIP)** session before a SIP INFO message can be transferred.

3.2.4 Higher-Layer Triggered Events

There are no prescribed triggered events required for implementation of this protocol.

The server can send a **picture_freeze** message when its internal video source selection logic determines that an input video stream is not being routed to any of the other clients and is therefore

unnecessary. An implementation can use any implementation-specific state changes or inputs as a factor in determining when to send the **picture_freeze** message.

If **picture_freeze** and **picture_fast_update** messages are not sent, clients continue to send video packets to the server, even when the server is discarding all video packets from that client.

3.2.5 Message Processing Events and Sequencing Rules

None.

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

4 Protocol Examples

A typical message body containing the **picture_freeze** message is shown in the following example. This message body can be carried in any valid SIP INFO message according to [RFC2976] section 2.

```
<?xml version="1.0" encoding="utf-8" ?>
<media_control>
    <vc_primitive>
        <to_encoder>
            <picture freeze/>
        </to_encoder>
    </vc primitive>
</media_control>
```

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

6 Appendix A: XML Schema for Media Control

The schema described in [IETFDRAFT-XMLSMC-12] section 5 is included here for reference. Note that the schema described in [IETFDRAFT-XMLSMC-12] section 5 contains a typographical error in the line that defines the **picture_fast_update** element.

```
<?xml version="1.0" encoding="utf-8" ?>

<xs:schema id="TightMediaControl"
   elementFormDefault="qualified"
   xmlns:xsd="http://www.w3.org/2001/XMLSchema">


      <xs:element name="media_control">
         <xs:complexType>
            <xs:sequence>
               <xs:element name="vc primitive"
                  type="vc_primitive"
                  minOccurs="0"
                  maxOccurs="unbounded" />
               <xs:element name="general_error"
                  type="xs:string"
                  minOccurs="0"
                  maxOccurs="unbounded" />
            </xs:sequence>
         </xs:complexType>
      </xs:element>

   <!-- Video control primitive.  -->

      <xs:complexType name="vc_primitive">
         <xs:sequence>
            <xs:element name="to_encoder" type="to_encoder" />
            <xs:element name="stream id"
               type="xsd:string"
               minOccurs="0"
               maxOccurs="unbounded" />
         </xs:sequence>
      </xs:complexType>

   <!-- Encoder Command:
      Picture Fast Update
   -->

      <xs:complexType name="to_encoder">
         <xs:choice>
               <xsd:element name="picture fast update"/>
         </xs:choice>
      </xs:complexType>
</xs:schema>
```

7 Appendix B: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include updates to those products.

- Microsoft Office Communications Server 2007
- Microsoft Office Communicator 2007
- Microsoft Office Communications Server 2007 R2
- Microsoft Office Communicator 2007 R2
- Microsoft Lync Server 2010
- Microsoft Lync 2010
- Microsoft Lync Server 2013
- Microsoft Lync Client 2013/Skype for Business
- Microsoft Skype for Business 2016
- Microsoft Skype for Business Server 2015
- Microsoft Skype for Business 2019
- Microsoft Skype for Business Server 2019

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms "SHOULD" or "SHOULD NOT" implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term "MAY" implies that the product does not follow the prescription.

8 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as Major, Minor, or None.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements.
- A document revision that captures changes to protocol functionality.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **None** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

The changes made to this document are listed in the following table. For more information, please contact dochelp@microsoft.com.

Section	Description	Revision class
7 Appendix B: Product Behavior	Updated list of supported products.	Major

9 Index

T

V

X