

[MS-CONFAS]:

Centralized Conference Control Protocol: Application Sharing Extensions

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft [Open Specification Promise](#) or the [Community Promise](#). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications do not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them. Certain Open Specifications are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

Revision Summary

Date	Revision History	Revision Class	Comments
12/12/2008	1.0		Initial version
2/13/2009	1.01		Revised and edited the technical content
3/13/2009	1.02		Edited the technical content
7/13/2009	1.03	Major	Revised and edited the technical content
8/28/2009	1.04	Editorial	Revised and edited the technical content
11/6/2009	1.05	Editorial	Revised and edited the technical content
2/19/2010	1.06	Editorial	Revised and edited the technical content
3/31/2010	1.07	Major	Updated and revised the technical content
4/30/2010	1.08	Editorial	Revised and edited the technical content
6/7/2010	1.09	Editorial	Revised and edited the technical content
6/29/2010	1.10	Editorial	Changed language and formatting in the technical content.
7/23/2010	1.10	No Change	No changes to the meaning, language, or formatting of the technical content.
9/27/2010	2.0	Major	Significantly changed the technical content.
11/15/2010	2.0	No Change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	2.0	No Change	No changes to the meaning, language, or formatting of the technical content.
3/18/2011	2.0	No Change	No changes to the meaning, language, or formatting of the technical content.
6/10/2011	2.0	No Change	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	3.0	Major	Significantly changed the technical content.
4/11/2012	3.0	No Change	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	3.0	No Change	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	3.1	Minor	Clarified the meaning of the technical content.
2/11/2013	3.1	No Change	No changes to the meaning, language, or formatting of the technical content.
7/30/2013	3.2	Minor	Clarified the meaning of the technical content.
11/18/2013	3.2	No Change	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	3.2	No Change	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
4/30/2014	3.2	No Change	No changes to the meaning, language, or formatting of the technical content.
7/31/2014	3.2	No Change	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	3.3	Minor	Clarified the meaning of the technical content.
9/4/2015	3.3	No Change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1	Introduction	6
1.1	Glossary	6
1.2	References	7
1.2.1	Normative References	7
1.2.2	Informative References	7
1.3	Overview	8
1.4	Relationship to Other Protocols	9
1.5	Prerequisites/Preconditions	9
1.6	Applicability Statement	9
1.7	Versioning and Capability Negotiation	9
1.8	Vendor-Extensible Fields	9
1.9	Standards Assignments.....	9
2	Messages.....	10
2.1	Transport	10
2.2	Message Syntax.....	10
2.2.1	MCU Conference Roster Document Format.....	10
2.2.1.1	MCU endpoint Element Syntax.....	10
2.2.1.1.1	endpoint Element Extension Elements	10
2.2.1.2	MCU conference-view Element Syntax	11
2.2.1.2.1	entity-state Extension Elements	11
2.2.1.2.1.1	media Element Extensions	11
2.2.1.2.1.2	session-ids-type Element	11
2.2.2	C3P request/response Document Content	11
2.2.2.1	addUser Dial-out Request Document Syntax.....	11
2.2.2.2	addUser Dial-in Request Document Syntax.....	11
2.2.2.2.1	media Element.....	12
3	Protocol Details.....	13
3.1	Common Details	13
3.1.1	Abstract Data Model.....	13
3.1.2	Timers	13
3.1.3	Initialization.....	13
3.1.4	Higher-Layer Triggered Events	13
3.1.5	Message Processing Events and Sequencing Rules	13
3.1.6	Timer Events.....	13
3.1.7	Other Local Events.....	13
3.2	Client Details	13
3.2.1	Abstract Data Model.....	13
3.2.2	Timers	13
3.2.3	Initialization	13
3.2.4	Higher-Layer Triggered Events	13
3.2.5	Message Processing Events and Sequencing Rules	14
3.2.5.1	SIP Dialog Events and Sequencing Rules.....	14
3.2.5.1.1	Establishing a SIP INVITE Dialog	14
3.2.5.1.2	Constructing the SDP Offer and handling the SDP Answer.....	14
3.2.5.1.3	ICE Re-Invite	14
3.2.5.1.4	Renegotiating Media	14
3.2.5.1.5	New Sharer joins conference.....	14
3.2.5.1.6	Specification of Capabilities	15
3.2.5.2	RDP Events and Sequencing Rules	15
3.2.5.2.1	Specification of GroupId	15
3.2.5.2.2	Specification of RemoteName.....	15
3.2.6	Timer Events.....	15
3.2.7	Other Local Events.....	16

3.3	Server Details.....	16
3.3.1	Abstract Data Model.....	16
3.3.1.1	Correlation of Media Parameters	16
3.3.2	Timers	17
3.3.3	Initialization.....	17
3.3.4	Higher-Layer Triggered Events	17
3.3.5	Message Processing Events and Sequencing Rules	17
3.3.5.1	Processing the addUser Dial-in Request	17
3.3.5.1.1	Constructing the addUser Dial-in Response.....	17
3.3.5.2	SIP Dialog Events and Sequencing Rules.....	17
3.3.5.2.1	Rules for processing of Received SDP Offer and Constructing an Answer	17
3.3.5.2.2	ICE Re-Invite	18
3.3.5.2.3	Renegotiating Media	18
3.3.5.3	RDP Events and Sequencing	19
3.3.5.3.1	Specification of RemoteName.....	19
3.3.5.4	User Notification Sequencing for Establishing a Media Connection	19
3.3.5.5	User Notification Sequencing for Disconnecting Media.....	20
3.3.5.6	User Notification Sequencing for Switching Sharer in a Conference.....	21
3.3.5.7	User Notification Sequencing for Policy Enforcement.....	23
3.3.5.7.1	Specification of Appsharing Allowed Policy	23
3.3.5.7.2	Specification of Attendees Cannot Share Policy.....	23
3.3.5.7.3	Specification of Supported Capabilities	23
3.3.6	Timer Events.....	24
3.3.7	Other Local Events.....	24
4	Protocol Examples	25
4.1	addUser Dial-In	25
4.2	Sharer Switch.....	31
5	Security	37
5.1	Security Considerations for Implementers	37
5.2	Index of Security Parameters	37
6	Appendix A: Full XML Schema.....	38
6.1	asconfinfoextensions Namespace Schema	38
7	Appendix B: Product Behavior	39
8	Change Tracking.....	40
9	Index.....	41

1 Introduction

This document specifies proprietary extensions to the Centralized Conference Control Protocol that can be used to integrate application sharing conference modes within the framework defined in the Basic Architecture and Signaling protocol, as described in [\[MS-CONFAS\]](#).

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in [\[RFC2119\]](#). Sections 1.5 and 1.9 are also normative but do not contain those terms. All other sections and examples in this specification are informative.

1.1 Glossary

The following terms are specific to this document:

200 OK: A response to indicate that the request has succeeded.

Application Sharing Multipoint Control Unit (ASMCU): A **Multipoint Control Unit (MCU)** that supports application sharing conferencing.

endpoint: A device that is connected to a computer network.

focus: A single user agent that maintains a dialog and **Session Initiation Protocol (SIP)** signaling relationship with each participant (2), implements conference policies, and ensures that each participant receives the media that comprise the tightly coupled conference.

INVITE: A **Session Initiation Protocol (SIP)** method that is used to invite a user or a service to participate in a session.

MCU-Conference-URI: A literal that specifies a URI that can be used to access conferencing services in the context of a **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.

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 [\[RFC3550\]](#).

Remote Desktop Protocol (RDP): A multi-channel protocol that allows a user to connect to a computer running Microsoft Terminal Services (TS). RDP enables the exchange of client and server settings and also enables negotiation of common settings to use for the duration of the connection, so that input, graphics, and other data can be exchanged and processed between client and server.

SDP answer: A **Session Description Protocol (SDP)** message that is sent by an answerer in response to an offer that is received from an offerer.

SDP offer: A **Session Description Protocol (SDP)** message that is sent by an offerer.

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 [\[RFC3261\]](#).

sharer: A client that is participating in a conference and is transmitting keyboard and mouse signals, and desktop graphics through a Real-Time Transport Protocol (RTP) media connection.

Uniform Resource Identifier (URI): A string that identifies a resource. The URI is an addressing mechanism defined in Internet Engineering Task Force (IETF) Uniform Resource Identifier (URI): Generic Syntax [\[RFC3986\]](#).

viewer: A client that is participating in a conference and is receiving mouse and keyboard signals, and desktop graphics through a Real-Time Transport Protocol (RTP) media connection.

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.

[MS-CONFBAS] Microsoft Corporation, "[Centralized Conference Control Protocol: Basic Architecture and Signaling](#)".

[MS-CONFPRO] Microsoft Corporation, "[Centralized Conference Control Protocol: Provisioning](#)".

[MS-ICE] Microsoft Corporation, "[Interactive Connectivity Establishment \(ICE\) Extensions](#)".

[MS-RDPEMC] Microsoft Corporation, "[Remote Desktop Protocol: Multiparty Virtual Channel Extension](#)".

[MS-SDPEXT] Microsoft Corporation, "[Session Description Protocol \(SDP\) Version 2.0 Extensions](#)".

[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>

[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>

[RFC4028] Donovan, S., and Rosenberg, J., "Session Timers in the Session Initiation Protocol (SIP)", RFC 4028, April 2005, <http://www.rfc-editor.org/rfc/rfc4028.txt>

[RFC4574] Levin, O., and Camarillo, G., "The Session Description Protocol (SDP) Label Attribute", RFC 4574, August 2006, <http://www.rfc-editor.org/rfc/rfc4574.txt>

[RFC4575] Rosenberg, J., Schulzrinne, H., and Levin, O., "A Session Initiation Protocol (SIP) Event Package for Conference State", RFC 4575, August 2006, <http://www.rfc-editor.org/rfc/rfc4575.txt>

1.2.2 Informative References

[MS-ICE2] Microsoft Corporation, "[Interactive Connectivity Establishment \(ICE\) Extensions 2.0](#)".

[MS-RDPBCGR] Microsoft Corporation, "[Remote Desktop Protocol: Basic Connectivity and Graphics Remoting](#)".

[MS-RTASPF] Microsoft Corporation, "[RTP Payload Format for Application Sharing Extensions](#)".

[MS-RTP] Microsoft Corporation, "[Real-time Transport Protocol \(RTP\) Extensions](#)".

[MS-SIPRE] Microsoft Corporation, "[Session Initiation Protocol \(SIP\) Routing Extensions](#)".

1.3 Overview

The Centralized Conference Control Protocol (C3P), as described in [\[MS-CONFBAS\]](#), extends the definition of a **Session Initiation Protocol (SIP)** event package for conference state, as described in [\[RFC4575\]](#), and defines a framework for aggregating more than one **Multipoint Control Unit (MCU)** in the context of a single logical conference, as described in [\[RFC4575\]](#).

Within C3P, centralized processing of conference media content is delegated to specialized media-type-specific MCU entities. For example, a multiparty conference that simultaneously encompasses Instant Messaging (IM), application sharing, and audio-video media types is processed by three separate logical MCU entities: one for IM, one for audio-video, and one for application sharing.

The Centralized Conference Control Protocol: Application Sharing Extensions [\[MS-CONFAS\]](#) specifies extensions to C3P that enable Session Initiation Protocol (SIP), **Session Description Protocol (SDP)**, and **Real-Time Transport Protocol (RTP)**-based application sharing conference modalities and features within the multiple-MCU architecture that is described in [\[MS-CONFBAS\]](#).

The framework described in [\[MS-CONFBAS\]](#) calls for MCU entities to maintain separate, media type-specific communication sessions with each protocol client. This specification assumes that the communication protocol for signaling and media handshakes between protocol clients and the logical **Application Sharing Multipoint Control Unit (ASMCU)** entity is the suite of protocols described by the following table:

Protocol	Content Relevant to This Protocol
[MS-SIPRE]	Extensions to SIP
[MS-SDPEXT]	Extensions to SDP
[RFC3264]	An offer/answer model with SDP
[MS-RTP]	Extensions to RTP
[MS-RTASPF]	Remote transport application sharing payload format
[MS-RDPBCGR]	Remote Desktop Protocol (RDP) , basic connectivity and graphics, remoting specification
[MS-RDPEMC]	Remote Desktop Protocol (RDP), multiparty virtual channel extension

This protocol specifies the necessary interactions between the previous protocols and the Centralized Conference Control Protocol (C3P). For example, this protocol specifies:

- Correlation of C3P conference state and message element and attribute values with a SIP **Uniform Resource Identifier (URI)** and SIP header values.
- Correlation of C3P conference state and message element and attribute values with the values of standard SDP attributes.
- Processing of C3P commands that result in an action on one or more SIP dialogs between the protocol server, or MCU, and protocol clients.
- Changes in SIP dialog states between a protocol client and the MCU that result in conference state changes and C3P notifications.

- Logic rules based on the C3P conference state that are required to be factored into media-negotiation behavior whenever a protocol client or MCU formulates an **SDP offer** or responds with an **SDP answer**.

This specification does not specify any XML schema extensions beyond that of [MS-CONFBAS]. It does define semantics of some parts of the XML schema and C3P message constructs in more detail than [MS-CONFBAS].

1.4 Relationship to Other Protocols

In addition to the dependencies described in [MS-CONFBAS] section 1.4, the following protocols are required components of a complete implementation:

- [MS-SDPEXT]
- [RFC3264]
- [MS-RTP]
- [MS-SIPRE]
- [MS-ICE]
- [MS-RTASPF]
- [MS-RDPBCGR]
- [MS-RDPEMC]
- [MS-ICE2]

Note that each of the previous protocols can be extended independently.

1.5 Prerequisites/Preconditions

In addition to the prerequisites and preconditions described in [MS-CONFBAS] section 1.5 and the protocol dependencies described in the previous list, the following assumptions apply:

- Both the protocol client and protocol server support mutually-interoperable implementations of all of the protocols listed in section 1.4.
- The application sharing payload format described in [MS-RTASPF] is supported. The protocol client and protocol server are required to be able to negotiate a viable RTP channel between them using the standard protocols mentioned in section 1.4.

1.6 Applicability Statement

The extensions defined in this specification apply when both of the following are true:

- The protocol client and protocol server both meet the prerequisites and preconditions in section 1.5.
- The protocol client and protocol server both intend to implement an application sharing conferencing mode within the framework and architecture described in [MS-CONFBAS].

1.7 Versioning and Capability Negotiation

This specification does not have any additional versioning and capability negotiation constraints beyond those described in [MS-CONFBAS].

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

This specification does not introduce a new transport to exchange messages; it is capable of being used with SIP Transport. The constraints and conditions for exchanging messages are specified in [\[MS-CONFBAS\]](#).

2.2 Message Syntax

This specification does not introduce new message formats outside of the encapsulating message structures and envelopes specified in [\[MS-CONFBAS\]](#). All messages within this section conform to the message syntax specification in [\[MS-CONFBAS\]](#) section 2.2.

Message elements and attributes that have specific semantics with respect to application sharing media are specified here. However, it is important to note that not all of the schema extension semantics specified in this specification are exclusive to application sharing media. They are emphasized in this specification to define them as they apply to the application sharing media type.

2.2.1 MCU Conference Roster Document Format

This section specifies extensions to the MCU Conference Roster Document Format specified in [\[MS-CONFBAS\]](#) section 2.2.5.

2.2.1.1 MCU endpoint Element Syntax

The model defined in [\[MS-CONFBAS\]](#) specifies the role of MCU entities in generating and maintaining MCU-specific **endpoint** elements. This section specifies extended message syntax and semantics of application sharing specific **endpoint** elements.

The XML schema for the type **endpoint-type** and the semantics of the elements it contains are originally established in [\[RFC4575\]](#). Extensions to [\[RFC4575\]](#) are specified in [\[MS-CONFBAS\]](#). This protocol further extends the semantics of the elements with **endpoint-type** relative to [\[RFC4575\]](#) and defines additional extension elements.

2.2.1.1.1 endpoint Element Extension Elements

This section defines the following extension elements of the **media** element within the **endpoint** element.

The **media** element is extended as follows:

session-id element: The **session-id** element carries the identifier that identifies the sharing session to which the **endpoint** is connected. This session identifier MUST be from the list of **session-ids** in **entity-state** defined in section [2.2.1.2.1.2](#). This element MUST be present. This element MUST match at least one **session-id** listed in **session-ids**. All **endpoint** elements with the same **session-id** are sharing or viewing in the same session.

status element: The **status** element declared in [\[RFC4575\]](#) section 5.7.3 has been reused here. An exception is that possible values MUST be limited to "sendonly" and "recvonly". "sendonly" denotes an endpoint which MUST be sharing in a conference and "recvonly" denotes an endpoint which MUST be viewing. This element MUST be present.

media-state element: The **media-state** element carries information about the connection state of the **endpoint** element's media. The value MUST be "joining" or "connected".

id attribute: The **id** attribute is an **xs:string** value that represents an integer. The value MUST be unique with respect to all users who are participants of the same conference.

2.2.1.2 MCU conference-view Element Syntax

This section specifies semantics for the notification message elements that reside within the MCU-specific **entity-view** element within the **conference-view** element defined in [\[MS-CONFAS\]](#) section 2.2.4.4.

2.2.1.2.1 entity-state Extension Elements

This specification defines the following extension elements of the **entity-state** element.

2.2.1.2.1.1 media Element Extensions

The Conference Document format defined in [\[MS-CONFAS\]](#) section 2.2.4 specifies the role of MCU entities in generating and maintaining MCU-specific **entity-view** elements and its sub-elements. This section specifies extended message syntax and semantics for MCU-specific entry elements within the **media** element within the **entity-state** element of the MCU-specific **entity-view** element.

The following extension semantics are defined relative to that of [\[RFC4575\]](#) section 5.3.4. Unless extension semantics are explicitly defined in this section or in [\[MS-CONFAS\]](#), the semantics specified in [\[RFC4575\]](#) section 5.3.4 apply.

label attribute: The **label** attribute is under the **entry** element, and it is the identifier for the MCU-centric view of the conference media. The **label** value for the application sharing conference modality MUST be "applicationsharing".

This attribute is the Session Description Protocol (SDP) **label** media attribute defined in [\[RFC4574\]](#) section 4.

2.2.1.2.1.2 session-ids-type Element

The XML type **msas:session-ids-type** is intended specifically for application sharing conference modalities. It is defined in the `asconfinfoextensions` namespace found at <http://schemas.microsoft.com/rtc/2005/08/asconfinfoextensions>.

The **session-ids** element MUST contain a list of **session-id** types, each of which represents the active sharing sessions within a conference. There SHOULD always be at least one **session-id** element in the **session-ids** list.

The semantics of the **session-id-type** is as follows:

session-id-type: Contains a **xs:string** which MUST be unique relative to other **session-id** elements in the **session-ids** list.

2.2.2 C3P request/response Document Content

2.2.2.1 addUser Dial-out Request Document Syntax

addUser dial-out requests are not supported by the ASMCU. Therefore, the rules specified in [\[MS-CONFAS\]](#) section 2.2.3.15 do not apply.

2.2.2.2 addUser Dial-in Request Document Syntax

In addition to the syntax rules given in [\[MS-CONFAS\]](#) section 2.2.3.17 for **addUser** dial-in requests, the additional rules in the following subsections apply.

2.2.2.2.1 media Element

Instances of the **media** element of the **endpoint** element can be present inside the **endpoint** element. If present, instances MUST conform to the specified **media** element syntax in section [2.2.1.2.1.1](#).

This specification does not define any processing rules or behavior related to **endpoint media** element(s) in **addUser** dial-in request messages. If instances of the **media** element are present in an **addUser** dial-in request, they are ignored.

3 Protocol Details

3.1 Common Details

3.1.1 Abstract Data Model

None.

3.1.2 Timers

For session-expires timer as defined in [\[RFC4028\]](#) section 4, a value of 600 seconds is implemented on the SIP dialog between the protocol client and protocol server.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

None.

3.1.5 Message Processing Events and Sequencing Rules

None.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

[None.](#)

3.2 Client Details

3.2.1 Abstract Data Model

None.

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Message Processing Events and Sequencing Rules

3.2.5.1 SIP Dialog Events and Sequencing Rules

This section specifies rules that apply to events triggered on the SIP Dialog.

3.2.5.1.1 Establishing a SIP INVITE Dialog

This section specifies application sharing media specific SDP content rules for SIP **INVITE** messages associated with an **addUser** dial-in. In addition to the rules specified in [\[MS-CONFBAS\]](#) section 3.11 for outgoing SIP INVITE requests, the following rules apply to SIP INVITE messages that follow **addUser** dial-in requests sent to the ASMCU.

It is assumed that protocol clients have subscribed to conference notifications and have followed all of the rules and recommendations specified in [\[MS-CONFBAS\]](#). Therefore, the protocol client is aware of the following information before constructing the SIP INVITE message and the SDP offer content contained within it.

The **MCU-Conference-URI** of the ASMCU that is extracted from the **conf-uris** element of the Conference Document, whose child **purpose** element contains the value "applicationsharing", as specified in [\[MS-CONFBAS\]](#) section 2.2.2.4.

3.2.5.1.2 Constructing the SDP Offer and handling the SDP Answer

The following rules apply when constructing the SDP offer and handling the SDP answer.

- When constructing the SDP offer, the protocol client is at liberty to negotiate only one media instance as a **sharer** and one media instance as a **viewer** where those values are defined in the **x-applicationsharing-role** of the SDP offer. If more than one **m** line with **x-applicationsharing-role** as "sharing" or more than one **m** line with **x-applicationsharing-role** as "viewing" is present, the MCU MUST reject the additional media instances using the conventional "port=0" semantics specified in [\[RFC3264\]](#) section 8.1.
- If there are no **media-type** "m=applicationsharing" lines present in the initial offer, the offer MUST be rejected with a 488 Not Acceptable Here response code.
- The protocol client MUST specify its desired role in the SDP offer. If the protocol client is negotiating as a viewer it MUST specify the **x-applicationsharing-role** to be "viewer". If the protocol client is negotiating as a sharer, it MUST specify the **x-applicationsharing-role** to be "sharer".
- In the received SDP answer from the MCU, the **x-applicationsharing-role** MUST have a "reverse" role than the desired role of the protocol client. For example, if the protocol client specifies its role as "sharer", the server specifies its role to be "viewer".

3.2.5.1.3 ICE Re-Invite

For completion of media negotiation, the rules specified in [\[MS-ICE\]](#) section 3.1.4 apply. A Final Offer and Answer with the chosen candidates MUST be exchanged to complete media negotiation.

3.2.5.1.4 Renegotiating Media

In the case that a protocol client wishes to renegotiate media, the protocol client MUST send a new SDP offer with a new **m=applicationsharing** line in the SDP media description with the existent **m=** line deactivated with conventional "port=0" semantics specified in [\[RFC3264\]](#).

3.2.5.1.5 New Sharer joins conference

In the case that the protocol client is connected as a sharer, it is ejected from the conference if another protocol client negotiates media as a sharer. The protocol client MUST receive a SIP BYE with the following header indicating why the protocol client was ejected:

ms-diagnostics-public:21000;reason="New sharer joined conference"

3.2.5.1.6 Specification of Capabilities

This section specifies the rules that apply to the specification of client conferencing capabilities. Individual client capabilities are parsed in the incoming SDP INVITE and sent to all conferencing users as C3P through the SIP dialog. The **capabilities** line in the SDP INVITE SHOULD<1> be preceded by **a=x-capabilities**. Each capability SHOULD<2> exist as name value pair strings. The values SHOULD<3> be one of the following strings: "sendonly", "recvonly", "sendrecv", or "none". The **capability** name SHOULD<4> contain only the following characters:

- lowercase English letters
- underscore characters
- hyphen characters

Each name and value in the pair MUST be separated by "=". Capabilities SHOULD<5> be separated by ";". The following is an example of two valid capabilities:

```
a=x-capabilities:capability1="sendrecv";capability2="none"
```

If the **capabilities** line is incorrectly formatted, the SDP INVITE is rejected with a 488 Not Acceptable Here Response. After the Re-Invite sequence specified in section 3.2.5.1.3 completes, a C3P notification SHOULD<6> be sent to the client indicating the SDP capabilities. The C3P format MUST follow the C3P conferencing extensions defined in [MS-CONFAS] section 2.2.2.2. The **media** element contains a sub-element titled **media-capabilities**. This element contains a **media-capability** element which has a **name** and a **value**. The data model is as follows.

```
|--user
  |--media
    |--media-capabilities
      |--media-capability
        |--name
        |--value
```

The name and values specified in the SDP INVITE are stored in the corresponding fields of the **media-capability**.

3.2.5.2 RDP Events and Sequencing Rules

This section specifies rules that apply to events triggered by the RDP payload over the RTP media stream.

3.2.5.2.1 Specification of GroupId

As defined in [MS-RDPEMC] section 2.2.4.1, the **GroupId** of the Attendee Connected Protocol Data Unit MUST be set to a well-known value of "3766552189".

3.2.5.2.2 Specification of RemoteName

As defined in [MS-RDPEMC] section 2.2.4.1, the **FriendlyName** that is received on the Participant-Created PDU MUST match the **user@host** portion of the **endpoint-uri** of one user in the conference roster document.

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

3.3 Server Details

3.3.1 Abstract Data Model

An MCU SHOULD maintain an internal table of active conferences that it is currently servicing. This table SHOULD be keyed by **MCU-Conference-URI**.

An MCU that maintains an internal representation of each active conference is recommended so that:

- It can easily retrieve current state information when processing messages.
- It can easily construct the contents of MCU Roster notification messages it has to send.

Because external messages always relate in some way to the Conference Document structure that is described in this protocol's overview, it is convenient to use that as the conceptual data model. In other words, the abstract data model is represented by the structure defined by the XML schema for the **conference-info** element and its entire hierarchy of subelements as described in [\[MS-CONFAS\]](#) section 6.3.1.

Using the Conference Document structure as the basis for representing abstract state allows interim processing steps to be described in terms of data-modification operations made directly on a copy of the Conference Document. Where externally-visible C3P messages contain parts and fragments of the conference document, descriptions of the interim steps are used in subsequent sections to illustrate how the externally-visible state changes are realized.

Note that the actual data model can be implemented using a variety of techniques. An implementation is at liberty to represent such data in any way convenient.

3.3.1.1 Correlation of Media Parameters

The message processing and sequencing rules specified for the server role correlates media information across conference media instances, user **endpoint** element media instances, and media descriptions contained in the SDP section of SIP messages.

- A conference media instance is described using the XML type **conference-medium-type** in an instance of an **entry** element within the **media** element within the MCU-specific **entity-state** element.
- A user media instance, as described using the XML type **media-type** as described in [\[MS-CONFAS\]](#) section 6.3.1, is an instance of a **media** element within the **endpoint** element.
- Media instances are represented in SDP by the **m=** line.
- The **type** element in the **media-type** corresponds to the **m=applicationsharing** line in the SDP media description.
- The **status** element in the **media** element corresponds to the **a=x-applicationsharing-role** defined in [\[MS-SDPEXT\]](#) section 1.3, and MUST have a value of "sendonly" or "recvonly".
- If the protocol client is negotiating as a sharer, the **x-applicationsharing-role** value is set to "sharer". If the protocol client is negotiating as a viewer, this value is set to "viewer".

- An **x-applicationsharing-role** of "sharer" in the media description MUST correlate to a **status** of "sendonly" in the conference roster document. An **x-application-sharing-role** of "viewer" in the media description MUST correlate to a **status** of "recvonly".

3.3.2 Timers

None.

3.3.3 Initialization

Upon conference creation, the MCU MUST publish a full notification populating the **conference-view** with the **entity-state** element.

3.3.4 Higher-Layer Triggered Events

None.

3.3.5 Message Processing Events and Sequencing Rules

Unless otherwise specified, the message processing rules defined in this specification assume that the commands documented herein are executed to their typical conclusion.

3.3.5.1 Processing the addUser Dial-in Request

Processing the **addUser** dial-in message consists of three general steps:

- Validating the message syntax and contents.
- Saving a record of the message contents for later reference when processing SIP INVITE messages.
- Constructing and sending the **addUser** dial-in response message.

On receipt of the **addUser** dial-in message, the MCU first validates the message syntax according to rules specified in [\[MS-CONFAS\]](#) section 2.2.3.17. If message validation fails, the MCU MUST send a "requestMalformed" C3P Response, as defined in [\[MS-CONFAS\]](#) section 3.5.4.2.3.2.

Once the request is deemed valid, the MCU saves the contents of the message for later reference when processing the received SIP INVITE message for this user.

3.3.5.1.1 Constructing the addUser Dial-in Response

When constructing and sending the **addUser** dial-in response, the rules specified in [\[MS-CONFAS\]](#) section 3.11.5.2.1 apply.

In addition, the following recommendation, as specified in [\[MS-CONFAS\]](#) section 3.11.5.2.1, SHOULD be followed:

- The MCU SHOULD populate the **connection-info** element with the key-value pairs using the recommended key values for **mcu-server-uri** and **mcu-conference-uri**.

3.3.5.2 SIP Dialog Events and Sequencing Rules

This section specifies rules that apply to events triggered on the SIP dialog.

3.3.5.2.1 Rules for processing of Received SDP Offer and Constructing an Answer

This section specifies common rules that apply to handling application sharing specific SDP offers or answers within the MCU entity. The rules specified in this section assume that the correlation relationships between SDP media instances and user media instances have been established.

The following conceptual steps specify the requirements for processing a received SDP offer.

For each SDP media instance that is correlated with a user media instance:

- If the SDP offer specifies that a previously negotiated media stream has been removed, as specified in [\[RFC3264\]](#) section 8.2, the MCU MUST omit the user media instance from subsequent MCU Roster (user) notifications.
- If the SDP offer specifies that a previously rejected or removed media stream has been re-instantiated using the same SDP media "slot", as specified in [\[RFC3264\]](#) section 8.1, the MCU MUST include the user media instance in subsequent MCU Roster (user) notifications and continue.
- The MCU MUST determine the value of the **status** element in the **media** element in the user notification by analyzing the offered SDP direction attribute, **a=x-applicationsharing-role**. If the SDP direction attribute is "sharer", the **status** element MUST be set to "sendonly". If the attribute is "viewer", the element MUST be set to "recvonly".
- If the media capabilities of the implementation do not support the parameters of the offered media instance, the MCU MUST reject the media instance using the conventional "port=0" semantics specified in [\[RFC3264\]](#) section 8.1.
- When constructing the SDP answer for this media instance, the MCU MUST specify the "reverse" direction for media flow. For example, if the SDP offer is received with an attribute **a=x-applicationsharing-role** with the value "sharer", the MCU responds with a direction of **a=x-applicationsharing-role** with the value "viewer".
- When constructing the SDP answer for this media instance, the MCU MUST specify the **session-id** that the protocol client is joining in the **a=x-applicationsharing-session-id** field.
- If the offered media instance has more than one **m=** line for "sharing" or more than one **m=** line for "viewing", the offer MUST be rejected entirely.
- If the resulting SDP answer would reject all offered media instances, the MCU MUST respond to the INVITE message with a SIP 488 Reason Code with a "Not Acceptable Here" **reason** phrase and do no further processing.
- The remainder of the SDP media description (**m** line) follows the specifications in [\[MS-SDPEXT\]](#) section 1.3, and is beyond the scope of this specification.

3.3.5.2.2 ICE Re-Invite

For completion of media negotiation, the rules specified in [\[MS-ICE\]](#) section 3.1.4 apply. A final offer and answer with the chosen candidates MUST be generated to complete media negotiation.

3.3.5.2.3 Renegotiating Media

This section describes processing rules for received SIP re-INVITE messages that occur on existing SIP dialogs. Renegotiation can occur when the protocol client wishes to switch from a sharer to viewer or a viewer to sharer.

When SIP re-INVITE messages are received, only the SDP content needs to be processed. It is assumed that a reasonable implementation would preserve the correlated relationships between media instances that were established or constructed during processing of the initially received SDP offer and thus those steps do not have to be repeated.

The following rules govern the way re-INVITE messages are processed:

- If the SDP offer contains new media instances, such as **m=** lines that have not previously appeared in any SDP offer, the new instances MUST be correlated with user **media** instances using the rules specified in section [3.3.1.1](#).
- The MCU MUST send a SIP **200 OK** message containing the SDP answer in response to the received INVITE message.

3.3.5.3 RDP Events and Sequencing

This section specifies rules that apply to events triggered by the RDP payload over the RTP media stream.

3.3.5.3.1 Specification of RemoteName

As defined in [MS-RDPEMC] section 2.2.4.1, the **FriendlyName** that is sent in the Participant-Created PDU MUST be equal to the **user@host** portion of the **endpoint-uri** in the **addUser** request for that user.

3.3.5.4 User Notification Sequencing for Establishing a Media Connection

This section specifies the sequence of notifications that are sent when a user connects media to an application sharing conference.

After sending the 200 OK message containing the SDP answer with the rules defined in the previous section 3.3.5.2.1, the MCU MUST send an MCU Conference Roster (user) notification to the **focus** containing a "full" user state for the user that has just sent the INVITE message. This notification contains information indicating that the SIP dialog has been connected, as shown in the following example.

```
<user entity="sip:adams@fabrikam.com">
  <endpoint p7:endpoint-uri="sip:adams@fabrikam.com;opaque=user:epid:cZ3vlGz54F-
0PtUJu9gQtwAA;gruu" entity="{25B88C75-A55D-4852-B36E-91C74046DA6D}" p7:session-
type="applicationsharing"
xmlns:p7="http://schemas.microsoft.com/rtc/2005/08/confinfoextensions">
  <status>connected</status>
  <joining-method>dialed-in</joining-method>
  <p7:roles>
    <entry>presenter</entry>
  </p7:roles>
  <p7:authMethod>enterprise</p7:authMethod>
  <p7:accessMethod>internal</p7:accessMethod>
</endpoint>
</user>
```

Once all SDP negotiation has finished, the MCU MUST send a notification indicating that media is connecting. The notification MUST contain a **media** element with the **media-state** "joining", as shown in the following example.

```
<user entity="sip:haas@fabrikam.com">
  <endpoint p7:endpoint-
uri="sip:haas@fabrikam.com;opaque=user:epid:2H2uT4mdmlWYztVQwEKDKQAA;gruu"
entity="{D8D9C858-265B-4BD8-AA94-578403B4A674}" p7:session-type="applicationsharing"
xmlns:p7="http://schemas.microsoft.com/rtc/2005/08/confinfoextensions">
  <status>connected</status>
  <joining-method>dialed-in</joining-method>
  <media id="4">
    <type>applicationsharing</type>
    <status>sendonly</status>
    <separator xmlns="urn:ietf:params:xml:ns:conference-info-separator" />
    <p7:media-state>joining</p7:media-state>
    <p7:session-id>1</p7:session-id>
  </media>
  <p7:roles>
    <entry>attendee</entry>
  </p7:roles>
  <p7:authMethod>enterprise</p7:authMethod>
  <p7:accessMethod>internal</p7:accessMethod>
</endpoint>
```

```
</user>
```

Once the RDP connection is established, as defined in [\[MS-RDPEMC\]](#), a notification with **media-state** "connected" MUST be sent, as shown in the following example.

```
<user entity="sip:haas@fabrikam.com">
  <endpoint p7:endpoint-
uri="sip:haas@fabrikam.com;opaque=user:epid:2H2uT4mdmlWYztVQwEKDKQAA;gruu"
entity="{D8D9C858-265B-4BD8-AA94-578403B4A674}" p7:session-type="applicationsharing"
xmlns:p7="http://schemas.microsoft.com/rtc/2005/08/confinfoextensions">
    <status>connected</status>
    <joining-method>dialed-in</joining-method>
    <media id="4">
      <type>applicationsharing</type>
      <status>sendonly</status>
      <separator xmlns="urn:ietf:params:xml:ns:conference-info-separator" />
      <p7:media-state>connected</p7:media-state>
      <p7:session-id>1</p7:session-id>
    </media>
    <p7:roles>
      <entry>attendeer</entry>
    </p7:roles>
    <p7:authMethod>enterprise</p7:authMethod>
    <p7:accessMethod>internal</p7:accessMethod>
  </endpoint>
</user>
```

3.3.5.5 User Notification Sequencing for Disconnecting Media

Whenever a media instance is disconnected for a user, this information is conveyed in the notifications by omitting the **media** element for that connection.

The following notification indicates that a media connection does not exist for the protocol client.

```
<user entity="sip:haas@fabrikam.com">
  <endpoint p7:endpoint-
uri="sip:haas@fabrikam.com;opaque=user:epid:2H2uT4mdmlWYztVQwEKDKQAA;gruu"
entity="{D8D9C858-265B-4BD8-AA94-578403B4A674}" p7:session-type="applicationsharing"
xmlns:p7="http://schemas.microsoft.com/rtc/2005/08/confinfoextensions">
    <status>connected</status>
    <joining-method>dialed-in</joining-method>
    <p7:roles>
      <entry>attendeer</entry>
    </p7:roles>
    <p7:authMethod>enterprise</p7:authMethod>
    <p7:accessMethod>internal</p7:accessMethod>
  </endpoint>
</user>
```

A sequence of the following two notifications indicates that media has been disconnected for the protocol client.

```
<user entity="sip:haas@fabrikam.com">
  <endpoint p7:endpoint-
uri="sip:haas@fabrikam.com;opaque=user:epid:2H2uT4mdmlWYztVQwEKDKQAA;gruu"
entity="{D8D9C858-265B-4BD8-AA94-578403B4A674}" p7:session-type="applicationsharing"
xmlns:p7="http://schemas.microsoft.com/rtc/2005/08/confinfoextensions">
    <status>connected</status>
    <joining-method>dialed-in</joining-method>
    <media id="4">
      <type>applicationsharing</type>
```

```

        <status>sendonly</status>
        <separator xmlns="urn:ietf:params:xml:ns:conference-info-separator" />
        <p7:media-state>connected</p7:media-state>
        <p7:session-id>1</p7:session-id>
    </media>
    <p7:roles>
        <entry>attendee</entry>
    </p7:roles>
    <p7:authMethod>enterprise</p7:authMethod>
    <p7:accessMethod>internal</p7:accessMethod>
</endpoint>
</user>

```

The next notification is the missing **media** element:

```

<user entity="sip:haas@fabrikam.com">
  <endpoint p7:endpoint-
uri="sip:haas@fabrikam.com;opaque=user:epid:2H2uT4mdmlWYztVQwEKDKQAA;gruu"
entity="{D8D9C858-265B-4BD8-AA94-578403B4A674}" p7:session-type="applicationsharing"
xmlns:p7="http://schemas.microsoft.com/rtc/2005/08/confinfoextensions">
    <status>connected</status>
    <joining-method>dialed-in</joining-method>
    <p7:roles>
        <entry>attendee</entry>
    </p7:roles>
    <p7:authMethod>enterprise</p7:authMethod>
    <p7:accessMethod>internal</p7:accessMethod>
  </endpoint>
</user>

```

3.3.5.6 User Notification Sequencing for Switching Sharer in a Conference

This section describes cases where a protocol client is connected to a conference as a sharer and another protocol client attempts to negotiate media as a sharer. The MCU MUST eject the previous sharer by sending a SIP BYE to the original sharing protocol client with the following signaling header indicating why the protocol client was ejected from the conference:

```
"ms-diagnostics-public:21000;reason="New sharer joined conference"
```

The following sequence specifies the roster notifications that are sent when the sharer switches in a conference.

The new sharer negotiates media, and the sequence of notifications in section [3.3.5](#) are sent until the **media-state** "joining" notification is sent.

The following is an example of the state of the user:

```

<user entity="sip:adams@fabrikam.com">
  <endpoint p7:endpoint-
uri="sip:haas@fabrikam.com;opaque=user:epid:2H2uT4mdmlWYztVQwEKDKQAA;gruu" entity="{D8D9C858-
265B-4BD8-AA94-578403B4A674}" p7:session-type="applicationsharing"
xmlns:p7="http://schemas.microsoft.com/rtc/2005/08/confinfoextensions">
    <status>connected</status>
    <joining-method>dialed-in</joining-method>
    <media id="4">
        <type>applicationsharing</type>
        <status>sendonly</status>
        <separator xmlns="urn:ietf:params:xml:ns:conference-info-separator" />
        <p7:media-state>joining</p7:media-state>
        <p7:session-id>1</p7:session-id>
    </media>
    <p7:roles>

```

```

    <entry>attendee</entry>
  </p7:roles>
  <p7:authMethod>enterprise</p7:authMethod>
  <p7:accessMethod>internal</p7:accessMethod>
</endpoint>
</user>

```

The old sharer has its media disconnected and is subsequently ejected from the conference. A status notification containing no **media** element, which indicates that there is no media connection, **MUST** be sent.

In the following examples, Bacon is the original sharer and Adams is the viewer switching to a sharer. An example of the notification indicating that there is no media is as follows:

```

<user entity="sip:bacon@fabrikam.com">
  <endpoint p7:endpoint-
uri="sip:haas@fabrikam.com;opaque=user:epid:2H2uT4mdmlWYztVQwEKDKQAA;gruu" entity="{D8D8C858-
265B-4BD8-AA94-578403B4A674}" p7:session-type="applicationsharing"
xmlns:p7="http://schemas.microsoft.com/rtc/2005/08/confinfoextensions">
  <status>connected</status>
  <joining-method>dialled-in</joining-method>
  <p7:roles>
    <entry>attendee</entry>
  </p7:roles>
  <p7:authMethod>enterprise</p7:authMethod>
  <p7:accessMethod>internal</p7:accessMethod>
</endpoint>
</user>

```

After the old sharer protocol client is ejected from the conference, a notification **MUST** be sent with a **state** attribute with the value "deleted," indicating that the protocol client has been ejected. An example of the notification that the client has been ejected is as follows:

```

<users state="partial">
  <user state="deleted" entity="sip:bacon@fabrikam.com" />
</users>
<separator xmlns="urn:ietf:params:xml:ns:conference-info-separator" />

```

The new sharer now connects its media and a notification, indicating that the **media-state** is "connected", **MUST** be sent.

An example of the notification indicating the media connection is as follows:

```

<user entity="sip:adams@fabrikam.com">
  <endpoint p7:endpoint-
uri="sip:haas@fabrikam.com;opaque=user:epid:2H2uT4mdmlWYztVQwEKDKQAA;gruu"
entity="{D8D9C858-265B-4BD8-AA94-578403B4A674}" p7:session-type="applicationsharing"
xmlns:p7="http://schemas.microsoft.com/rtc/2005/08/confinfoextensions">
  <status>connected</status>
  <joining-method>dialled-in</joining-method>
  <media id="4">
    <type>applicationsharing</type>
    <status>sendonly</status>
    <separator xmlns="urn:ietf:params:xml:ns:conference-info-separator" />
    <p7:media-state>connected</p7:media-state>
    <p7:session-id>1</p7:session-id>
  </media>
  <p7:roles>
    <entry>attendee</entry>
  </p7:roles>
  <p7:authMethod>enterprise</p7:authMethod>
  <p7:accessMethod>internal</p7:accessMethod>
</endpoint>

```

```
</user>
```

3.3.5.7 User Notification Sequencing for Policy Enforcement

3.3.5.7.1 Specification of Appsharing Allowed Policy

The MCU MUST [<7>](#) send a **permission** element in the **conference-view** specified in section 2.2.2.2 of [\[MS-CONFBAS\]](#) with the **name** "AllowUserToScheduleMeetingsWithAppSharing" and **value** either "true" or "false". If the value is "false", the MCU MUST reject any SIP INVITES to that conference with the following diagnostic code:

```
ms-diagnostics-public: 21001;reason="Attendees cannot share in this conference"
```

The C3P hierarchy is as follows:

```
|-- conference-view (+)
|   |-- entity-view (+)
|       |-- entity-state (+)
|           |-- permissions (+)
|               |-- permission-type (+)
|                   |-- name (+)
|                   |-- value (+)
```

3.3.5.7.2 Specification of Attendees Cannot Share Policy

If the **server-mode** element specified in section 2.2.2.1 of [\[MS-CONFPRO\]](#) is not set to "13" or is not specified, the MCU MUST reject all incoming SIP INVITES to the specified conference from users with **role** defined as "attendee" as specified in section 2.2.3.3 of [\[MS-CONFBAS\]](#), and with SDP **x-applicationsharingrole** defined as "sharer", with the following **ms-diagnostics** code:

```
ms-diagnostics-public: 21002;reason="Attendees cannot share in this conference"
```

A **permission** element called **AttendeesCanShare** will be sent to the clients in the **conference-view** conference roster. This element MUST have the value "true", if the **server-mode** is set to "13", and "false" if the **server-mode** is not equal to "13" or is not specified.

```
|-- conference-view (+)
|   |-- entity-view (+)
|       |-- entity-state (+)
|           |-- permissions (+)
|               |-- permission-type (+)
|                   |-- name (+)
|                   |-- value (+)
```

3.3.5.7.3 Specification of Supported Capabilities

The MCU MUST [<8>](#) send a **supports-application-desktop-sharing** element in the **conference-view** specified in section 2.2.2.2 of [\[MS-CONFBAS\]](#) with value either "desktop", "singleApplication" or "none". If the value is "singleApplication", clients MUST NOT initiate a sharing session to share the desktop. If the value is "desktop" then the Clients can initiate a sharing session to share the desktop and any application. The value of "none" restricts clients from initiating application sharing session.

The C3P hierarchy is as follows:

```
|-- conference-view (+)
|   |-- entity-view (+)
|       |-- entity-capabilities (+)
```



```
|          |- capabilities (+)
|          |- supports-application-desktop-sharing (+)
```

3.3.6 Timer Events

None.

3.3.7 Other Local Events

None.

4 Protocol Examples

4.1 addUser Dial-In

In the following example, a typical call flow sequence for an **addUser** dial-in appears. In the example, Bacon has already created, joined, and subscribed to the conference. Adams has already joined and subscribed to the conference and has obtained the MCU service URI for **applicationsharing**. The dial-in flow begins with Adams joining the application sharing conference modality by sending a SIP INVITE message to the service URI.

Bacon is the sharer and Adams is the viewer in the following examples.

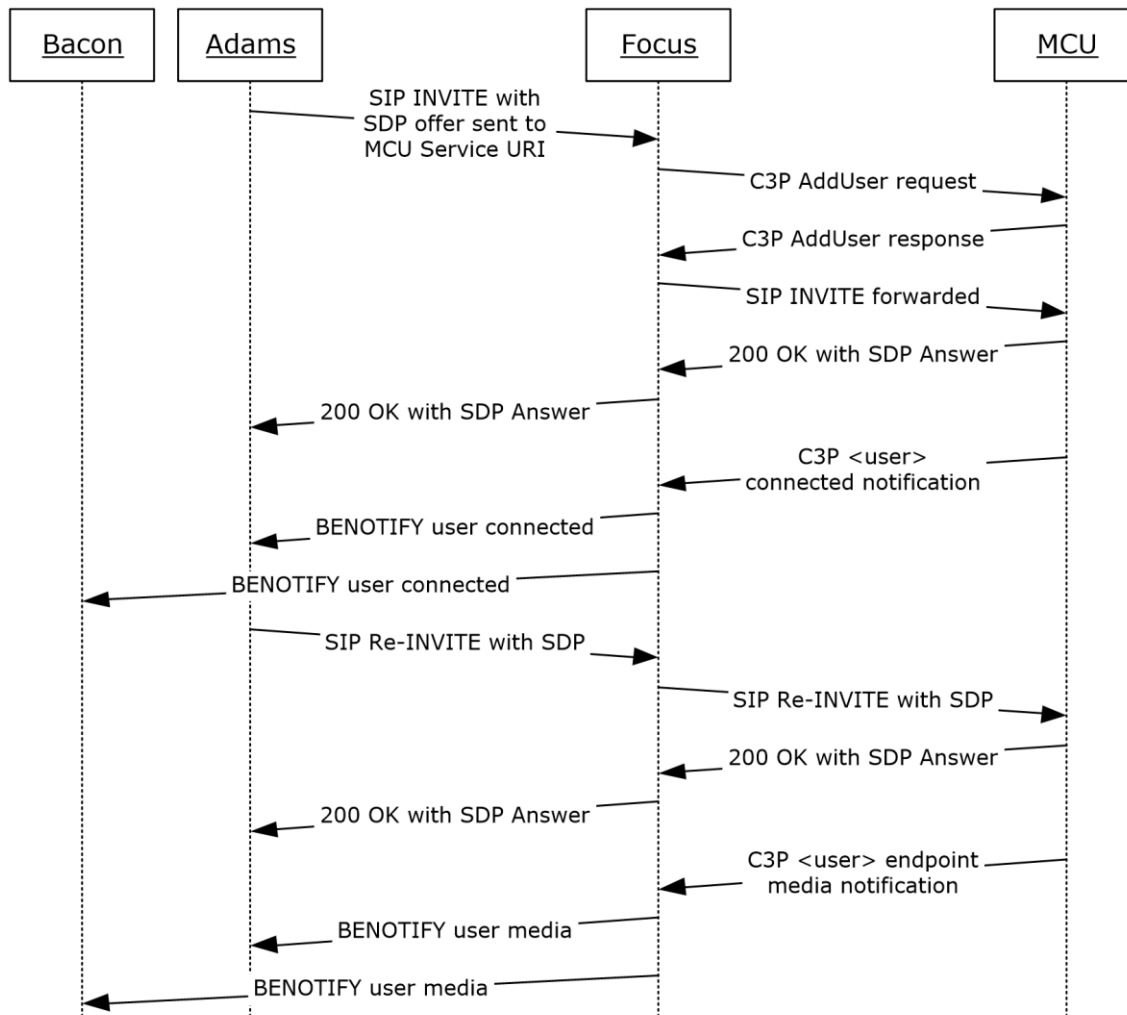


Figure 1: addUser dial-in call flow sequence

The following notification shows the conference roster state before Adams dials in to the conference.

```
<?xml version="1.0" encoding="utf-8"?>
<notify xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
notificationId="7#sip:bacon@fabrikam.com;gruu;opaque=app:conf:focus:id:
37787CBB1D3C444C8E29BDC4A79709A1" C3PVersion="1"
```

```

from="https://server.fabrikam.com:444/LiveServer/ASMCU/"
to="https://server.Fabrikam.com:444/LiveServer/Focus"
xmlns="urn:ietf:params:xml:ns:cccp">
  <conference-info version="5"
entity="sip:bacon@fabrikam.com;gruu;opaque=app:conf:focus:id:
37787CBB1D3C444C8E29BDC4A79709A1" xmlns="urn:ietf:params:xml:ns:conference-info">
  <users>
    <user entity="sip:bacon@fabrikam.com">
      <endpoint p7:endpoint-uri="sip:bacon@fabrikam.com;opaque=user:epid:c_3X-
18a F6zpxqZjfV1fwAA;gruu" entity="{CF3ACBF0-701C-4E3D-8EBA-49EDAD407F93}" p7:session-
type="applicationsharing"
xmlns:p7="http://schemas.microsoft.com/rtc/2005/08/confinfoextensions">
        <status>connected</status>
        <joining-method>dialed-in</joining-method>
        <media id="1">
          <type>applicationsharing</type>
          <status>sendonly</status>
          <separator xmlns="urn:ietf:params:xml:ns:conference-info-separator" />
          <p7:media-state>connected</p7:media-state>
          <p7:session-id>1</p7:session-id>
        </media>
        <p7:roles>
          <entry>presenter</entry>
        </p7:roles>
        <p7:authMethod>enterprise</p7:authMethod>
        <p7:accessMethod>internal</p7:accessMethod>
      </endpoint>
    </user>
  </users>
  <conference-view p5:state="full" xmlns:p5="urn:ietf:params:xml:ns:conference-info"
xmlns="http://schemas.microsoft.com/rtc/2005/08/confinfoextensions">
    <entity-view p5:state="full"
entity="sip:bacon@fabrikam.com;gruu;opaque=app:conf:applicationsharing:id:
37787CBB1D3C444C8E29BDC4A79709A1">
      <entity-capabilities>
        <separator xmlns="urn:ietf:params:xml:ns:conference-info-separator"/>
        <capabilities
xmlns="http://schemas.microsoft.com/rtc/2005/08/asconfinfoextensions">
          <supports-application-desktop-sharing>desktop</supports-application-desktop-
sharing>
        </capabilities>
      </entity-capabilities>
      <entity-state>
        <locked>>false</locked>
        <media>
          <p5:entry label="applicationsharing">
            <p5:type>applicationsharing</p5:type>
          </p5:entry>
        </media>
        <separator xmlns="urn:ietf:params:xml:ns:conference-info-separator" />
        <session-ids
xmlns="http://schemas.microsoft.com/rtc/2005/08/asconfinfoextensions">
          <session-id>1</session-id>
        </session-ids>
      </entity-state>
    </entity-view>
  </conference-view>
  <separator xmlns="urn:ietf:params:xml:ns:conference-info-separator" />
  <trusted-entities xmlns="http://schemas.microsoft.com/rtc/2005/08/confinfoextensions"
/>
</conference-info>
</notify>

```

The following sample message shows the INVITE message sent from Adams's protocol client.

```

INVITE sip:server.Fabrikam.com:5065;transport=tls SIP/2.0
FROM: "adams"<sip:adams@fabrikam.com>;tag=8d240414e3;epid=24e8722df9

```

```
TO:
<sip:bacon@fabrikam.com;gruu;opaque=app:conf:applicationsharing:id:37787CBB1D3C444C8E29BD
C4A79709A1>
CSEQ: 1 INVITE
CALL-ID: 34f4e3f2f1b5404793ea9544ee987ddd
...Other SIP Headers...
v=0
o=- 0 0 IN IP4 11.0.0.25
s=session
c=IN IP4 11.0.0.25
b=CT:99980
t=0 0
m=applicationsharing 19575 TCP/RTP/SAVP 127
a=ice-ufraq:chA32g
a=ice-pwd:l6B6JV9K10xar3iMvfCT5aY5eerRdXd
a=candidate:1 1 TCP-PASS 2120613887 11.0.0.25 17323 typ host
a=candidate:1 2 TCP-PASS 2120613374 11.0.0.25 17323 typ host
a=candidate:2 1 TCP-ACT 2121006591 11.0.0.25 19575 typ host
a=candidate:2 2 TCP-ACT 2121006078 11.0.0.25 19575 typ host
a=candidate:3 1 TCP-PASS 2120612863 192.168.0.245 27564 typ host
a=candidate:3 2 TCP-PASS 2120612350 192.168.0.245 27564 typ host
a=candidate:4 1 TCP-ACT 2121005567 192.168.0.245 10678 typ host
a=candidate:4 2 TCP-ACT 2121005054 192.168.0.245 10678 typ host
a=setup:active
a=connection:new
a=maxptime:200
a=rtcp:19575
a=mid:1
a=rtpmap:127 x-data/90000
a=encryption:rejected
a=x-applicationsharing-role:viewer
a=x-applicationsharing-media-type:rdp
```

The following is an example of the SDP answer the MCU sends in response.

```
SIP/2.0 200 OK
FROM: "adams"<sip:adams@fabrikam.com>;tag=8d240414e3;epid=24e8722df9
TO:
<sip:bacon@fabrikam.com;gruu;opaque=app:conf:applicationsharing:id:37787CBB1D3C444C8E29BD
C4A79709A1>;tag=805cb33db;epid=5398C874B2
CSEQ: 1 INVITE
CALL-ID: 34f4e3f2f1b5404793ea9544ee987ddd
...Other SIP Headers...
Session-Expires: 600;refresher=uac
...Other SIP Headers...
v=0
o=- 0 0 IN IP4 192.168.0.240
s=session
c=IN IP4 192.168.0.240
b=CT:1000
t=0 0
m=applicationsharing 52973 TCP/RTP/SAVP 127
c=IN IP4 11.0.0.25
a=rtpmap:127 x-data/90000
a=mid:1
a=connection:existing
a=setup:active
a=rtcp:52973
a=ice-ufraq://9dwQ
a=ice-pwd:sSowfxlgODx9Nb1Ur5/gm+h8GUxySBmt
a=candidate:1 1 tcp-pass 2120613887 192.168.0.240 59052 typ host raddr 192.168.0.240
rport 59052
a=candidate:1 2 tcp-pass 2120613374 192.168.0.240 59052 typ host raddr 192.168.0.240
rport 59052
```

```

a=candidate:2 1 tcp-act 2121006591 192.168.0.240 49848 typ host raddr 192.168.0.240 rport
49848
a=candidate:2 2 tcp-act 2121006078 192.168.0.240 49848 typ host raddr 192.168.0.240 rport
49848
a=candidate:3 1 tcp-pass 6555135 11.0.0.25 52973 typ relay raddr 11.0.0.25 rport 52973
a=candidate:3 2 tcp-pass 6555134 11.0.0.25 52973 typ relay raddr 11.0.0.25 rport 52973
a=candidate:4 1 tcp-act 7076607 11.0.0.25 52973 typ relay raddr 11.0.0.25 rport 52973
a=candidate:4 2 tcp-act 7076094 11.0.0.25 52973 typ relay raddr 11.0.0.25 rport 52973
a=label:applicationsharing
a=x-applicationsharing-session-id:1
a=x-applicationsharing-role:sharer
a=x-applicationsharing-media-type:rdp

```

The following notification indicating that Adams's SIP dialog is connected is sent to all users subscribed to conference notifications.

```

<?xml version="1.0" encoding="utf-8"?>
<notify xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
notificationId="7#sip:bacon@fabrikam.com;gruu;opaque=app:conf:focus:id:37787CBB1D3C444C8E
29BDC4A79709A1" C3PVersion="1" from="https://server.Fabrikam.com:444/LiveServer/ASMCU/"
to="https://server.Fabrikam.com:444/LiveServer/Focus"
xmlns="urn:ietf:params:xml:ns:cccp">
  <conference-info version="5" state="partial"
entity="sip:bacon@fabrikam.com;gruu;opaque=app:conf:focus:id:37787CBB1D3C444C8E29BDC4A797
09A1" xmlns="urn:ietf:params:xml:ns:conference-info">
    <users state="partial">
      <user entity="sip:adams@fabrikam.com">
        <endpoint p7:endpoint-
uri="sip:bacon@fabrikam.com;opaque=user:epid:QDSEVSAXT1qbmoMIGmndEAAA;gruu"
entity="{B1B278AD-BB72-4FF2-A9E1-C94D76419FED}" p7:session-type="applicationsharing"
xmlns:p7="http://schemas.microsoft.com/rtc/2005/08/confinfoextensions">
          <status>connected</status>
          <joining-method>dialed-in</joining-method>
          <p7:roles>
            <entry>attende</entry>
          </p7:roles>
          <p7:authMethod>enterprise</p7:authMethod>
          <p7:accessMethod>internal</p7:accessMethod>
        </endpoint>
      </user>
    </users>
    <separator xmlns="urn:ietf:params:xml:ns:conference-info-separator" />
  </conference-info>
</notify>

```

The protocol client sends a re-INVITE with SDP with the chosen candidates similar to the following example.

```

INVITE sip:server.Fabrikam.com:5065;transport=TLS SIP/2.0
FROM: <sip:adams@fabrikam.com>;tag=8d240414e3;epid=24e8722df9
TO:
<sip:bacon@fabrikam.com;gruu;opaque=app:conf:applicationsharing:id:37787CBB1D3C444C8E29BD
C4A79709A1>;tag=805cb33db;epid=5398C874B2
CSEQ: 2 INVITE
CALL-ID: 34f4e3f2f1b5404793ea9544ee987ddd
...Other SIP Headers...
v=0
o=- 0 0 IN IP4 192.168.0.245
s=session
c=IN IP4 192.168.0.245
b=CT:99980
t=0 0
m=applicationsharing 21461 TCP/RTP/SAVP 127

```

```
a=ice-ufrag:chA32g
a=ice-pwd:l6B6JV9K1Oxar3iMvfCT5aY5eerRdXd
a=candidate:7 1 TCP-ACT 2121005567 192.168.0.245 21461 typ host
a=candidate:7 2 TCP-ACT 2121005054 192.168.0.245 21461 typ host
a=remote-candidates:1 192.168.0.240 59052 2 192.168.0.240 59052
a=setup:active
a=connection:existing
a=maxptime:200
a=rtcp:21461
a=mid:1
a=rtpmap:127 x-data/90000
a=x-applicationsharing-role:viewer
a=x-applicationsharing-media-type:rdp
```

An SDP answer with the chosen media candidates is sent in a 200 OK.

```
SIP/2.0 200 OK
FROM: <sip:adams@fabrikam.com>;tag=8d240414e3;epid=24e8722df9
TO:
<sip:bacon@fabrikam.com;gruu;opaque=app:conf:applicationsharing:id:37787CBB1D3C444C8E29BD
C4A79709A1>;epid=5398C874B2;tag=805cb33db
CSEQ: 2 INVITE
CALL-ID: 34f4e3f2f1b5404793ea9544ee987ddd
...Other SIP Headers...
Session-Expires: 600;refresher=uac
...Other SIP Headers...
v=0
o=- 0 1 IN IP4 192.168.0.240
s=session
c=IN IP4 192.168.0.240
b=CT:1000
t=0 0
m=applicationsharing 59052 TCP/RTP/SAVP 127
c=IN IP4 192.168.0.240
a=rtpmap:127 x-data/90000
a=mid:1
a=connection:existing
a=setup:passive
a=rtcp:59052
a=ice-ufrag://9dwQ
a=ice-pwd:sSowfxlgODx9Nb1Ur5/gm+h8GUxySBmt
a=candidate:1 1 tcp-pass 2120613887 192.168.0.240 59052 typ host raddr 192.168.0.240
rport 59052
a=candidate:1 2 tcp-pass 2120613374 192.168.0.240 59052 typ host raddr 192.168.0.240
rport 59052
a=remote-candidates:1 192.168.0.245 21461 2 192.168.0.245 21461
a=label:applicationsharing
a=x-applicationsharing-session-id:1
a=x-applicationsharing-role:sharer
a=x-applicationsharing-media-type:rdp
```

At this point, the SDP has fully negotiated, and the following notification is sent with **media-state** with the value "joining".

```
<?xml version="1.0" encoding="utf-8"?>
<notify xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
notificationId="9#sip:bacon@fabrikam.com;gruu;opaque=app:conf:focus:id:37787CBB1D3C444C8E
29BDC4A79709A1" C3PVersion="1" from="https://server.Fabrikam.com:444/LiveServer/ASMCU/"
to="https://server.Fabrikam.com:444/LiveServer/Focus"
xmlns="urn:ietf:params:xml:ns:cccp">
  <conference-info version="7" state="partial"
entity="sip:bacon@fabrikam.com;gruu;opaque=app:conf:focus:id:37787CBB1D3C444C8E29BDC4A797
09A1" xmlns="urn:ietf:params:xml:ns:conference-info">
```

```

    <users state="partial">
      <user entity="sip:adams@fabrikam.com">
        <endpoint p7:endpoint-
uri="sip:bacon@fabrikam.com;opaque=user:epid:QDSEVsAXT1qbmoMIGmndEAAA;gruu"
entity="{B1B278AD-BB72-4FF2-A9E1-C94D76419FED}" p7:session-type="applicationsharing"
xmlns:p7="http://schemas.microsoft.com/rtc/2005/08/confinfoextensions">
          <status>connected</status>
          <joining-method>dialed-in</joining-method>
          <media id="2">
            <type>applicationsharing</type>
            <status>recvonly</status>
            <separator xmlns="urn:ietf:params:xml:ns:conference-info-separator" />
            <p7:media-state>joining</p7:media-state>
            <p7:session-id>1</p7:session-id>
          </media>
          <p7:roles>
            <entry>attendeer</entry>
          </p7:roles>
          <p7:authMethod>enterprise</p7:authMethod>
          <p7:accessMethod>internal</p7:accessMethod>
        </endpoint>
      </user>
    </users>
    <separator xmlns="urn:ietf:params:xml:ns:conference-info-separator" />
  </conference-info>
</notify>

```

And finally, once the RDP negotiation completes over the RTP channel, a notification that **media-state** is "connected" is sent.

```

<?xml version="1.0" encoding="utf-8"?>
<notify xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
notificationId="10#sip:bacon@fabrikam.com;gruu;opaque=app:conf:focus:id:37787CBB1D3C444C8
E29BDC4A79709A1" C3PVersion="1" from="https://server.Fabrikam.com:444/LiveServer/ASMCU/"
to="https://server.Fabrikam.com:444/LiveServer/Focus"
xmlns="urn:ietf:params:xml:ns:cccp">
  <conference-info version="8" state="partial"
entity="sip:bacon@fabrikam.com;gruu;opaque=app:conf:focus:id:37787CBB1D3C444C8E29BDC4A797
09A1" xmlns="urn:ietf:params:xml:ns:conference-info">
    <users state="partial">
      <user entity="sip:adams@fabrikam.com">
        <endpoint p7:endpoint-
uri="sip:adams@fabrikam.com;opaque=user:epid:QDSEVsAXT1qbmoMIGmndEAAA;gruu"
entity="{B1B278AD-BB72-4FF2-A9E1-C94D76419FED}" p7:session-type="applicationsharing"
xmlns:p7="http://schemas.microsoft.com/rtc/2005/08/confinfoextensions">
          <status>connected</status>
          <joining-method>dialed-in</joining-method>
          <media id="2">
            <type>applicationsharing</type>
            <status>recvonly</status>
            <separator xmlns="urn:ietf:params:xml:ns:conference-info-separator" />
            <p7:media-state>connected</p7:media-state>
            <p7:session-id>1</p7:session-id>
          </media>
          <p7:roles>
            <entry>attendeer</entry>
          </p7:roles>
          <p7:authMethod>enterprise</p7:authMethod>
          <p7:accessMethod>internal</p7:accessMethod>
        </endpoint>
      </user>
    </users>
    <separator xmlns="urn:ietf:params:xml:ns:conference-info-separator" />
  </conference-info>
</notify>

```

4.2 Sharer Switch

In the example, Bacon is sharing in the conference and Adams is viewing. Adams then goes on to renegotiate and become a sharer. The following example shows a typical call-flow sequence for a sharer switch.

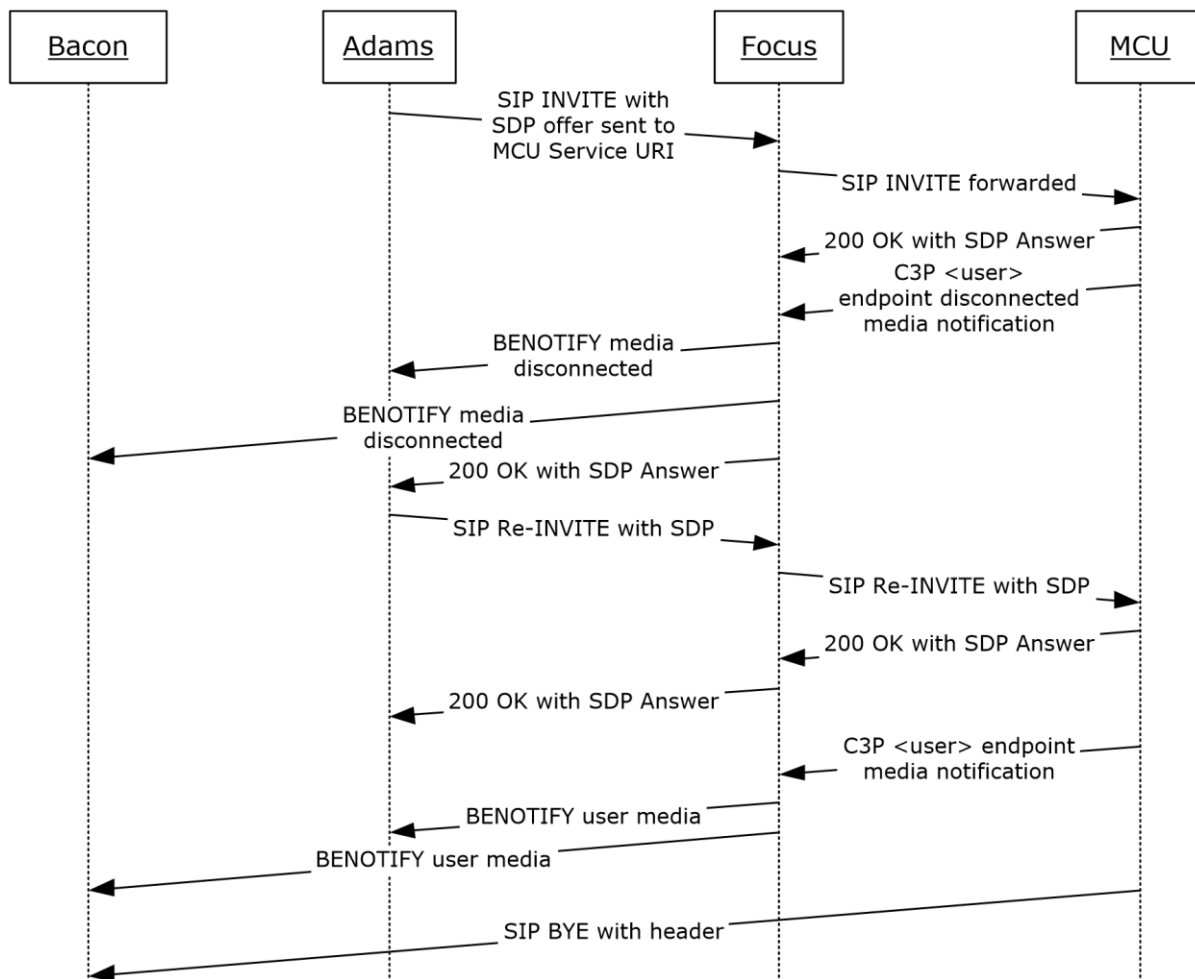


Figure 2: Switch sharer call flow sequence

The new SDP offer to renegotiate is sent on the existing SIP dialog. The SDP offer sent by Adams contains a new **media-type** "m=applicationsharing" line indicating that it will be a sharer in the session. The original "m=applicationsharing" line has the value "port=0", indicating that it will be disconnected. The new "m=applicationsharing" line contains information about the new media that will be negotiated.

```

INVITE sip:server.Fabrikam.com:5065;transport=Tls SIP/2.0
From: <sip:adams@fabrikam.com>;tag=63e63da53e;epid=7e108179c5
To:
<sip:adams@fabrikam.com;gruu;opaque=app:conf:applicationsharing:id:DAA7C2F6E6303A4584E703
180714F7AF>;tag=f62043201b;epid=8C5B7B76C2
CSeq: 3 INVITE
Call-ID: d076780f514a47ce8d78c7bc3f2be347
...Other SIP Headers...
o=- 0 0 IN IP4 192.168.0.238
s=session
  
```



```

c=IN IP4 192.168.0.238
b=CT:99980
t=0 0
m=applicationsharing 0 RTP/SAVP 34
m=applicationsharing 14029 TCP/RTP/SAVP 127
a=ice-ufrag:ePx6ew
a=ice-pwd:TjmChfRDStxmmmovuIP141G8iIifiJPQW
a=candidate:1 1 TCP-PASS 2120613887 192.168.0.238 11133 typ host
a=candidate:1 2 TCP-PASS 2120613374 192.168.0.238 11133 typ host
a=candidate:2 1 TCP-ACT 2121006591 192.168.0.238 14029 typ host
a=candidate:2 2 TCP-ACT 2121006078 192.168.0.238 14029 typ host
a=cryptoscale:1 client AES_CM 128 HMAC SHA1 80
inline:+wFmy3sHXabzGAbxqgHosW4Rcmk7jEJ5rxlQB/51|2^31|1:1
a=crypto:2 AES_CM 128 HMAC_SHA1 80
inline:zEtyPrpYvpM9f6DFljUM776pr5bV07B2dzGsmHJx|2^31|1:1
a=setup:active
a=connection:new
a=maxptime:200
a=rtcp:14029
a=mid:2
a=rtpmap:127 x-data/90000
a=encryption:required
a=x-applicationsharing-role:sharer
a=x-applicationsharing-media-type:rdp

```

The MCU disconnects the original media connection and sends out the notification.

```

<?xml version="1.0" encoding="utf-8"?>
<notify xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
notificationId="13#sip:adams@fabrikam.com;gruu;opaque=app:conf:focus:id:DAA7C2F6E6303A458
4E703180714F7AF" C3PVersion="1" from="https://server.Fabrikam.com:444/LiveServer/ASMCU"
to="https://server.Fabrikam.com:444/LiveServer/Focus"
xmlns="urn:ietf:params:xml:ns:cccp">
  <conference-info version="11" state="partial"
entity="sip:adams@fabrikam.com;gruu;opaque=app:conf:focus:id:DAA7C2F6E6303A4584E703180714
F7AF" xmlns="urn:ietf:params:xml:ns:conference-info">
    <users state="partial">
      <user entity="sip:adams@fabrikam.com">
        <endpoint p7:endpoint-
uri="sip:haas@fabrikam.com;opaque=user:epid:2H2uT4mdmlWYztVQwEKDKQAA;gruu"
entity="{D8D9C858-265B-4BD8-AA94-578403B4A674}" p7:session-type="applicationsharing"
xmlns:p7="http://schemas.microsoft.com/rtc/2005/08/confinfoextensions">
          <status>connected</status>
          <joining-method>dialed-in</joining-method>
          <p7:roles>
            <entry>attendee</entry>
          </p7:roles>
          <p7:authMethod>enterprise</p7:authMethod>
          <p7:accessMethod>internal</p7:accessMethod>
        </endpoint>
      </user>
    </users>
    <separator xmlns="urn:ietf:params:xml:ns:conference-info-separator" />
  </conference-info>
</notify>

```

The MCU then responds to the INVITE with a 200 OK and an SDP answer for the new media connection.

```

SIP/2.0 200 OK
From: <sip:adams@fabrikam.com>;tag=63e63da53e;epid=7e108179c5

```

To:
<sip:adams@fabrikam.com;gruu;opaque=app:conf:applicationsharing:id:DAA7C2F6E6303A4584E703180714F7AF>;epid=8C5B7B76C2;tag=f62043201b
CSeq: 3 INVITE
Call-ID: d076780f514a47ce8d78c7bc3f2be347
...Other SIP Headers...
Session-Expires: 600;refresher=uac
...Other SIP Headers...
Message-Body: v=0
o=- 0 2 IN IP4 192.168.0.240
s=session
c=IN IP4 192.168.0.240
b=CT:1000
t=0 0
m=applicationsharing 0 RTP/SAVP 127
a=rtpmap:127 x-data/90000
a=mid:1
m=applicationsharing 50589 TCP/RTP/SAVP 127
c=IN IP4 11.0.0.25
a=rtpmap:127 x-data/90000
a=mid:2
a=connection:existing
a=setup:active
a=rtcp:50589
a=ice-ufrag:5gh0jw
a=ice-pwd:nmjPs4GgOd4LVmjQZ3zUKgXb4bWNejWC
a=candidate:1 1 tcp-pass 2120613887 192.168.0.240 64479 typ host raddr 192.168.0.240 rport 64479
a=candidate:1 2 tcp-pass 2120613374 192.168.0.240 64479 typ host raddr 192.168.0.240 rport 64479
a=candidate:2 1 tcp-act 2121006591 192.168.0.240 51892 typ host raddr 192.168.0.240 rport 51892
a=candidate:2 2 tcp-act 2121006078 192.168.0.240 51892 typ host raddr 192.168.0.240 rport 51892
a=candidate:3 1 tcp-pass 6555135 11.0.0.25 50589 typ relay raddr 11.0.0.25 rport 50589
a=candidate:3 2 tcp-pass 6555134 11.0.0.25 50589 typ relay raddr 11.0.0.25 rport 50589
a=candidate:4 1 tcp-act 7076607 11.0.0.25 50589 typ relay raddr 11.0.0.25 rport 50589
a=candidate:4 2 tcp-act 7076094 11.0.0.25 50589 typ relay raddr 11.0.0.25 rport 50589
a=crypto:2 AES_CM_128_HMAC_SHA1_80
inline:7Kyeimk6riyYFeH3F9GmhnnKG5iTftPfoJWp1MWB|2^31|1:1
a=label:applicationsharing
a=x-applicationsharing-role:viewer
a=x-applicationsharing-media-type:rdp

Then the re-INVITE is sent from the new sharer protocol client.

INVITE sip:server.Fabrikam.com:5065;transport=TLS SIP/2.0
From: <sip:adams@fabrikam.com>;tag=63e63da53e;epid=7e108179c5
To:
<sip:adams@fabrikam.com;gruu;opaque=app:conf:applicationsharing:id:DAA7C2F6E6303A4584E703180714F7AF>;tag=f62043201b;epid=8C5B7B76C2
CSeq: 4 INVITE
Call-ID: d076780f514a47ce8d78c7bc3f2be347
...Other SIP Headers...
o=- 0 0 IN IP4 192.168.0.238
s=session
c=IN IP4 192.168.0.238
b=CT:99980
t=0 0
m=applicationsharing 0 RTP/SAVP 34
m=applicationsharing 5688 TCF/RTP/SAVP 127
a=ice-ufrag:ePx6ew
a=ice-pwd:TjmChfRDStxmmmovU1P141G8iIifiJPQW
a=candidate:2 1 TCP-ACT 2121006591 192.168.0.238 5688 typ host
a=candidate:2 2 TCP-ACT 2121006078 192.168.0.238 5688 typ host
a=crypto:2 AES_CM_128_HMAC_SHA1_80
inline:zEtyPrpYvpM9f6DF1jUM776pr5bV07B2dzGSmHJx|2^31|1:1

```
a=remote-candidates:1 192.168.0.240 64479 2 192.168.0.240 64479
a=setup:active
a=connection:existing
a=maxptime:200
a=rtcp:5688
a=mid:2
a=rtpmap:127 x-data/90000
a=x-applicationsharing-role:sharer
a=x-applicationsharing-media-type:rdp
```

The MCU responds with a 200 OK and the final SDP answer.

```
SIP/2.0 200 OK
From: <sip:adams@fabrikam.com>;tag=63e63da53e;epid=7e108179c5
To:
<sip:adams@fabrikam.com;gruu;opaque=app:conf:applicationsharing:id:DAA7C2F6E6303A4584E703
180714F7AF>;epid=8C5B7B76C2;tag=f62043201b
CSeq: 4 INVITE
Call-ID: d076780f514a47ce8d78c7bc3f2be347
...Other SIP Headers...
Session-Expires: 600;refresher=uac
...Other SIP Headers...
o=- 0 3 IN IP4 192.168.0.240
s=session
c=IN IP4 192.168.0.240
b=CT:1000
t=0 0
m=applicationsharing 0 RTP/SAVP 127
a=rtpmap:127 x-data/90000
a=mid:1
m=applicationsharing 64479 TCP/RTP/SAVP 127
c=IN IP4 192.168.0.240
a=rtpmap:127 x-data/90000
a=mid:2
a=connection:existing
a=setup:passive
a=rtcp:64479
a=ice-frag:5gh0jw
a=ice-pwd:nmjPs4GgOd4LVmjQZ3zUKgXb4bWNejWC
a=candidate:1 1 tcp-pass 2120613887 192.168.0.240 64479 typ host raddr 192.168.0.240
rport 64479
a=candidate:1 2 tcp-pass 2120613374 192.168.0.240 64479 typ host raddr 192.168.0.240
rport 64479
a=remote-candidates:1 192.168.0.238 5688 2 192.168.0.238 5688
a=crypto:2 AES_CM 128_HMAC_SHA1_80
inline:7Kyeimk6riyYFeH3F9GmhnnKG5iTftPfoJWpLMWB|2^31|1:1
a=label:applicationsharing
a=x-applicationsharing-role:viewer
a=x-applicationsharing-media-type:rdp
```

The MCU then sends a notification indicating that the new sharer's **media-state** is "joining".

```
<?xml version="1.0" encoding="utf-8"?>
<notify xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
notificationId="17#sip:adams@fabrikam.com;gruu;opaque=app:conf:focus:id:DAA7C2F6E6303A458
4E703180714F7AF" C3PVersion="1" from="https://server.Fabrikam.com:444/LiveServer/ASMCU"
to="https://server.Fabrikam.com:444/LiveServer/Focus"
xmlns="urn:ietf:params:xml:ns:cccp">
  <conference-info version="15" state="partial"
entity="sip:adams@fabrikam.com;gruu;opaque=app:conf:focus:id:DAA7C2F6E6303A4584E703180714
F7AF" xmlns="urn:ietf:params:xml:ns:conference-info">
    <users state="partial">
      <user entity="sip:adams@fabrikam.com">
```

```

    <endpoint p7:endpoint-
uri="sip:haas@fabrikam.com;opaque=user:epid:2H2uT4mdmlWYztVQwEKDKQAA;gruu"
entity="{D8D9C858-265B-4BD8-AA94-578403B4A674}" p7:session-type="applicationsharing"
xmlns:p7="http://schemas.microsoft.com/rtc/2005/08/confinfoextensions">
  <status>connected</status>
  <joining-method>dialled-in</joining-method>
  <media id="4">
    <type>applicationsharing</type>
    <status>sendonly</status>
    <separator xmlns="urn:ietf:params:xml:ns:conference-info-separator" />
    <p7:media-state>joining</p7:media-state>
    <p7:session-id>1</p7:session-id>
  </media>
  <p7:roles>
    <entry>attendeer</entry>
  </p7:roles>
  <p7:authMethod>enterprise</p7:authMethod>
  <p7:accessMethod>internal</p7:accessMethod>
</endpoint>
</user>
</users>
<separator xmlns="urn:ietf:params:xml:ns:conference-info-separator" />
</conference-info>
</notify>

```

The MCU sends a notification indicating that the old sharer's **media** element has been removed. This notification might not be present if the user is deleted before its media disconnects. In that case, the protocol client only receives the following notification that the user was deleted.

```

<?xml version="1.0" encoding="utf-8"?>
<notify xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
notificationId="15#sip:adams@fabrikam.com;gruu;opaque=app:conf:focus:id:DAA7C2F6E6303A458
4E703180714F7AF" C3PVersion="1" from="https://server.Fabrikam.com:444/LiveServer/ASMCU"
to="https://server.Fabrikam.com:444/LiveServer/Focus"
xmlns="urn:ietf:params:xml:ns:cccp">
  <conference-info version="13" state="partial"
entity="sip:adams@fabrikam.com;gruu;opaque=app:conf:focus:id:DAA7C2F6E6303A4584E703180714
F7AF" xmlns="urn:ietf:params:xml:ns:conference-info">
    <users state="partial">
      <user entity="sip:bacon@fabrikam.com">
        <endpoint p7:endpoint-uri="sip:adams@fabrikam.com;opaque=user:epid:cZ3vlGz54F-
0PtUJu9gQtwAA;gruu" entity="{25B88C75-A55D-4852-B36E-91C74046DA6D}" p7:session-
type="applicationsharing"
xmlns:p7="http://schemas.microsoft.com/rtc/2005/08/confinfoextensions">
          <status>connected</status>
          <joining-method>dialled-in</joining-method>
          <p7:roles>
            <entry>presenter</entry>
          </p7:roles>
          <p7:authMethod>enterprise</p7:authMethod>
          <p7:accessMethod>internal</p7:accessMethod>
        </endpoint>
      </user>
    </users>
    <separator xmlns="urn:ietf:params:xml:ns:conference-info-separator" />
  </conference-info>
</notify>

```

The user deleted notification is sent for the old sharer.

```

<?xml version="1.0" encoding="utf-8"?>
<notify xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"

```

```

notificationId="16#sip:adams@fabrikam.com;gruu;opaque=app:conf:focus:id:DAA7C2F6E6303A458
4E703180714F7AF" C3PVersion="1" from="https://server.Fabrikam.com:444/LiveServer/ASMCU"
to="https://server.Fabrikam.com:444/LiveServer/Focus"
xmlns="urn:ietf:params:xml:ns:cccp">
  <conference-info version="14" state="partial"
entity="sip:adams@fabrikam.com;gruu;opaque=app:conf:focus:id:DAA7C2F6E6303A4584E703180714
F7AF" xmlns="urn:ietf:params:xml:ns:conference-info">
    <users state="partial">
      <user state="deleted" entity="sip:bacon@fabrikam.com" />
    </users>
    <separator xmlns="urn:ietf:params:xml:ns:conference-info-separator" />
  </conference-info>
</notify>

```

The media connected notification is sent for the new sharer. This notification can be sent before the preceding user deleted notification because the two operations do not happen serially.

```

<?xml version="1.0" encoding="utf-8"?>
<notify xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
notificationId="17#sip:adams@fabrikam.com;gruu;opaque=app:conf:focus:id:DAA7C2F6E6303A458
4E703180714F7AF" C3PVersion="1" from="https://server.Fabrikam.com:444/LiveServer/ASMCU"
to="https://server.Fabrikam.com:444/LiveServer/Focus"
xmlns="urn:ietf:params:xml:ns:cccp">
  <conference-info version="15" state="partial"
entity="sip:adams@fabrikam.com;gruu;opaque=app:conf:focus:id:DAA7C2F6E6303A4584E703180714
F7AF" xmlns="urn:ietf:params:xml:ns:conference-info">
    <users state="partial">
      <user entity="sip:adams@fabrikam.com">
        <endpoint p7:endpoint-
uri="sip:haas@fabrikam.com;opaque=user:epid:2H2uT4mdmlWYztVQwEKDKQAA;gruu"
entity="{D8D9C858-265B-4BD8-AA94-578403B4A674}" p7:session-type="applicationsharing"
xmlns:p7="http://schemas.microsoft.com/rtc/2005/08/confinfoextensions">
          <status>connected</status>
          <joining-method>dialed-in</joining-method>
          <media id="4">
            <type>applicationsharing</type>
            <status>sendonly</status>
            <separator xmlns="urn:ietf:params:xml:ns:conference-info-separator" />
            <p7:media-state>connected</p7:media-state>
            <p7:session-id>1</p7:session-id>
          </media>
          <p7:roles>
            <entry>attendee</entry>
          </p7:roles>
          <p7:authMethod>enterprise</p7:authMethod>
          <p7:accessMethod>internal</p7:accessMethod>
        </endpoint>
      </user>
    </users>
    <separator xmlns="urn:ietf:params:xml:ns:conference-info-separator" />
  </conference-info>
</notify>

```

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

6 Appendix A: Full XML Schema

6.1 asconfinfoextensions Namespace Schema

Schema reference specified in [\[MS-CONFAS\]](#) section 6.3.8.

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 released service packs.

- 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

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product 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.

<1> [Section 3.2.5.1.6](#): Office Communicator 2007 R2, Office Communications Server 2007 R2: This behavior is not implemented.

<2> [Section 3.2.5.1.6](#): Office Communicator 2007 R2, Office Communications Server 2007 R2: This behavior is not implemented.

<3> [Section 3.2.5.1.6](#): Office Communicator 2007 R2, Office Communications Server 2007 R2: This behavior is not implemented.

<4> [Section 3.2.5.1.6](#): Office Communicator 2007 R2, Office Communications Server 2007 R2: This behavior is not implemented.

<5> [Section 3.2.5.1.6](#): Office Communicator 2007 R2, Office Communications Server 2007 R2: This behavior is not implemented.

<6> [Section 3.2.5.1.6](#): Office Communicator 2007 R2, Office Communications Server 2007 R2: This behavior is not implemented. Lync 2010 can interoperate with Office Communications Server 2007 R2 and Microsoft Office Communications Server 2007 which do not send a C3P notification to the client indicating the SDP capabilities.

<7> [Section 3.3.5.7.1](#): Office Communicator 2007 R2, Office Communications Server 2007 R2: This behavior is not implemented.

<8> [Section 3.3.5.7.3](#): Office Communicator 2007 R2, Office Communications Server 2007 R2, Lync 2010, Lync Server 2010: This behavior is not implemented.

8 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

9 Index

A

Abstract data model
 client ([section 3.1.1](#) 13, [section 3.2.1](#) 13)
 server ([section 3.1.1](#) 13, [section 3.3.1](#) 16)
 [correlation of media parameters](#) 16
[addUser dial-in example](#) 25
[Applicability](#) 9
application/conference-info+xml
 schema
 [asconfinfoextensions namespace](#) 38

C

C3P request/response Document Content message
 [addUser dial-in request document](#) 11
 [addUser dial-out request document](#) 11
[Capability negotiation](#) 9
[Change tracking](#) 40
Client
 abstract data model ([section 3.1.1](#) 13, [section 3.2.1](#) 13)
 higher-layer triggered events ([section 3.1.4](#) 13, [section 3.2.4](#) 13)
 initialization ([section 3.1.3](#) 13, [section 3.2.3](#) 13)
 [local events](#) 13
 [message processing](#) 13
 [RDP](#) 15
 [SIP dialog](#) 14
 [other local events](#) 16
 sequencing rules
 [RDP](#) 15
 [SIP dialog](#) 14
 timer events ([section 3.1.6](#) 13, [section 3.2.6](#) 15)
 timers ([section 3.1.2](#) 13, [section 3.2.2](#) 13)
[Correlation of media parameters - server](#) 16

D

Data model - abstract
 [client](#) 13
 [server](#) 16

E

Examples
 [addUser dial-in](#) 25
 [sharer switch](#) 31

F

[Fields - vendor-extensible](#) 9

G

[Glossary](#) 6

H

Higher-layer triggered events
 client ([section 3.1.4](#) 13, [section 3.2.4](#) 13)
 server ([section 3.1.4](#) 13, [section 3.3.4](#) 17)

I

[Implementer - security considerations](#) 37
[Index of security parameters](#) 37
[Informative references](#) 7
Initialization
 client ([section 3.1.3](#) 13, [section 3.2.3](#) 13)
 server ([section 3.1.3](#) 13, [section 3.3.3](#) 17)
[Introduction](#) 6

L

Local events
 [client](#) 13
 [server](#) 13

M

[MCU Conference Roster Document Format message](#)
 10
 [MCU conference-view element](#) 11
 [MCU endpoint element](#) 10
Message processing
 [client](#) 13
 [RDP](#) 15
 [SIP dialog](#) 14
 server ([section 3.1.5](#) 13, [section 3.3.5](#) 17)
 [addUser dial-in request](#) 17
 [RDP](#) 19
 [SIP dialog](#) 17
 user notifications
 [disconnecting media](#) 20
 [establishing connection](#) 19
 [policy enforcement](#) 23
 [switching sharer in a conference](#) 21
Messages
 C3P request/response Document Content
 [addUser dial-in request document](#) 11
 [addUser dial-out request document](#) 11
 [MCU Conference Roster Document Format](#) 10
 elements
 [MCU conference-view](#) 11
 [MCU endpoint](#) 10
 [syntax](#) 10
 [transport](#) 10

N

[Normative references](#) 7

O

Other local events
 [client](#) 16
 [server](#) 24
[Overview \(synopsis\)](#) 8

P

[Parameters - security index](#) 37
[Preconditions](#) 9

[Prerequisites](#) 9
[Product behavior](#) 39

R

RDP

client
[message processing](#) 15
[sequencing rules](#) 15
server
[message processing](#) 19
[sequencing rules](#) 19

References

[informative](#) 7
[normative](#) 7

[Relationship to other protocols](#) 9

S

Schema

application/conference-info+xml
[asconfinfoextensions namespace](#) 38

Security

[implementer considerations](#) 37
[parameter index](#) 37

Sequencing rules

client
[RDP](#) 15
[SIP dialog](#) 14
[server](#) 17
[addUser dial-in request](#) 17
[RDP](#) 19
[SIP dialog](#) 17
user notifications
[disconnecting media](#) 20
[establishing connection](#) 19
[policy enforcement](#) 23
[switching sharer in a conference](#) 21

Server

abstract data model ([section 3.1.1](#) 13, [section 3.3.1](#) 16)
[correlation of media parameters](#) 16
higher-layer triggered events ([section 3.1.4](#) 13, [section 3.3.4](#) 17)
initialization ([section 3.1.3](#) 13, [section 3.3.3](#) 17)
[local events](#) 13
message processing ([section 3.1.5](#) 13, [section 3.3.5](#) 17)
[addUser dial-in request](#) 17
[RDP](#) 19
[SIP dialog](#) 17
user notifications
[disconnecting media](#) 20
[establishing connection](#) 19
[policy enforcement](#) 23
[switching sharer in a conference](#) 21
[other local events](#) 24
[sequencing rules](#) 17
[addUser dial-in request](#) 17
[RDP](#) 19
[SIP dialog](#) 17
user notifications
[disconnecting media](#) 20
[establishing connection](#) 19
[policy enforcement](#) 23

[switching sharer in a conference](#) 21

timer events ([section 3.1.6](#) 13, [section 3.3.6](#) 24)
timers ([section 3.1.2](#) 13, [section 3.3.2](#) 17)

[Sharer switch example](#) 31

SIP dialog

client
[message processing](#) 14
[sequencing rules](#) 14
server
[message processing](#) 17
[sequencing rules](#) 17

[Standards assignments](#) 9

[Syntax](#) 10

T

Timer events

client ([section 3.1.6](#) 13, [section 3.2.6](#) 15)
server ([section 3.1.6](#) 13, [section 3.3.6](#) 24)

Timers

client ([section 3.1.2](#) 13, [section 3.2.2](#) 13)
server ([section 3.1.2](#) 13, [section 3.3.2](#) 17)

[Tracking changes](#) 40

[Transport](#) 10

Triggered events - higher layer

[client](#) 13
[server](#) 13

Triggered events - higher-layer

[client](#) 13
[server](#) 17

U

User notifications

disconnecting media ([section 3.3.5.5](#) 20, [section 3.3.5.6](#) 21)
[establishing connection](#) 19
[policy enforcement](#) 23

V

[Vendor-extensible fields](#) 9

[Versioning](#) 9