

[MS-CONFPRO]: Centralized Conference Control Protocol: Provisioning Specification

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.
- **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.

Preliminary Documentation. This Open Specification provides documentation for past and current releases and/or for the pre-release (beta) version of this technology. This Open Specification is final documentation for past or current releases as specifically noted in the document, as applicable; it is preliminary documentation for the pre-release (beta) versions. Microsoft will release final documentation in connection with the commercial release of the updated or new version of this technology. As the documentation may change between this preliminary version and the final version of this technology, there are risks in relying on preliminary documentation. To the extent that you incur additional development obligations or any other costs as a result of relying on this preliminary documentation, you do so at your own risk.

Revision Summary

Date	Revision History	Revision Class	Comments
04/04/2008	0.1		Initial version
04/25/2008	0.2		Revised and edited the technical content
06/27/2008	1.0		Revised and edited the technical content
08/15/2008	1.01		Revised and edited the technical content
12/12/2008	2.0		Revised and edited the technical content
02/13/2009	2.01		Revised and edited the technical content
03/13/2009	2.02		Revised and edited the technical content
07/13/2009	2.03	Major	Revised and edited the technical content
08/28/2009	2.04	Editorial	Revised and edited the technical content
11/06/2009	2.05	Editorial	Revised and edited the technical content
02/19/2010	2.06	Editorial	Revised and edited the technical content
03/31/2010	2.07	Major	Updated and revised the technical content
04/30/2010	2.08	Editorial	Revised and edited the technical content
06/07/2010	2.09	Editorial	Revised and edited the technical content
06/29/2010	2.10	Editorial	Changed language and formatting in the technical content.
07/23/2010	2.10	No change	No changes to the meaning, language, or formatting of the technical content.
09/27/2010	3.0	Major	Significantly changed the technical content.
11/15/2010	3.0	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	3.0	No change	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
03/18/2011	3.0	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	3.0	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	4.0	Major	Significantly changed the technical content.
04/11/2012	4.0	No change	No changes to the meaning, language, or formatting of the technical content.

Preliminary

Table of Contents

1 Introduction	7
1.1 Glossary	7
1.2 References	8
1.2.1 Normative References	8
1.2.2 Informative References	8
1.3 Protocol Overview (Synopsis)	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	10
2 Messages	11
2.1 Transport	11
2.2 Message Syntax	11
2.2.1 Conference Data Model	12
2.2.1.1 conference-info Element	13
2.2.1.2 conference-description Element	14
2.2.1.3 conference-state Element	16
2.2.1.4 users Element	16
2.2.1.5 conference-view Element	17
2.2.2 Requests	19
2.2.2.1 addConference Request	20
2.2.2.2 modifyConference Request	21
2.2.2.3 deleteConference Request	21
2.2.2.4 getConference Request	21
2.2.2.5 getConferences Request	22
2.2.2.6 getEncryptionKey Request	22
2.2.2.7 getAvailableMcuTypes Request	22
2.2.2.8 getConferencingCapabilities Request	23
2.2.3 Responses	23
2.2.3.1 addConference Response	23
2.2.3.2 modifyConference Response	25
2.2.3.3 deleteConference Response	26
2.2.3.4 getConference Response	26
2.2.3.5 getConferences Response	27
2.2.3.6 getEncryptionKey Response	28
2.2.3.7 getAvailableMcuTypes Response	28
2.2.3.8 getConferencingCapabilities Response	29
3 Protocol Details	34
3.1 Client Details	34
3.1.1 Abstract Data Model	34
3.1.2 Timers	34
3.1.3 Initialization	34
3.1.4 Higher-Layer Triggered Events	34
3.1.4.1 Create a Conference	34
3.1.4.2 Modify a Conference	35
3.1.4.3 Delete a Conference	35

3.1.4.4	Query for a List of Conferences	35
3.1.4.5	Query for the Details of a Conference.....	35
3.1.4.6	Obtain an Encryption key.....	36
3.1.4.7	Obtain a List of MCU-Types.....	36
3.1.4.8	Obtain Focus Factory Capabilities.....	36
3.1.5	Message Processing Events and Sequencing Rules.....	36
3.1.5.1	Processing a SIP Response.....	36
3.1.5.2	Processing a Response	37
3.1.5.3	Processing an addConference Response.....	37
3.1.5.4	Processing a modifyConference Response	37
3.1.5.5	Processing a deleteConference Response	37
3.1.5.6	Processing a getConference Response	38
3.1.5.7	Processing a getConferences Response.....	38
3.1.5.8	Processing a getEncryptionKey Response.....	38
3.1.5.9	Processing a getAvailableMcuTypes Response	38
3.1.5.10	Processing a getConferencingCapabilities Response.....	38
3.1.6	Timer Events	39
3.1.7	Other Local Events	39
3.1.7.1	Send a Request	39
3.1.7.2	Obtain a Focus Factory Certificate for Encryption	39
3.1.7.3	Create a Self-signed Certificate	39
3.2	Focus Factory Details	39
3.2.1	Abstract Data Model	39
3.2.2	Timers	40
3.2.3	Initialization	40
3.2.4	Higher-Layer Triggered Events.....	40
3.2.5	Message Processing Events and Sequencing Rules.....	40
3.2.5.1	Processing a SIP Request.....	40
3.2.5.2	Processing an addConference Request.....	40
3.2.5.3	Processing a modifyConference Request.....	41
3.2.5.4	Processing a deleteConference Request	41
3.2.5.5	Processing a getConference Request	42
3.2.5.6	Processing a getConferences Request.....	42
3.2.5.7	Processing a getEncryptionKey Request.....	42
3.2.5.8	Processing a getAvailableMcuTypes Request.....	42
3.2.5.9	Processing a getConferencingCapabilities Request	42
3.2.6	Timer Events	43
3.2.6.1	Conference Expired.....	43
3.2.7	Other Local Events	43
4	Protocol Examples.....	44
4.1	Create a Conference.....	44
4.2	Modify a Conference.....	45
4.3	Delete a Conference.....	49
4.4	Query for a List of Conferences.....	50
4.5	Query for the Details of a Conference.....	52
4.6	Obtaining an Encryption Key	53
4.7	Obtaining a List of MCU-Types.....	54
4.8	Obtaining Focus Factory Capabilities	55
4.9	Delegation Request.....	57
5	Security.....	59
5.1	Security Considerations for Implementers.....	59

5.2 Index of Security Parameters	59
6 Appendix A: Product Behavior	60
7 Change Tracking.....	64
8 Index	65

Preliminary

1 Introduction

This document specifies Centralized Conference Control Protocol Provisioning, which is an extension of the Centralized Conference Control protocol. This protocol extension is used by the protocol client application of a conference organizer to communicate with a managing component to create a conference and subsequently query, modify, and delete the conference.

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 RFC 2119. Sections 1.5 and 1.9 are also normative but cannot contain those terms. All other sections and examples in this specification are informative.

1.1 Glossary

The following terms are defined in [\[MS-GLOS\]](#):

anonymous user
certificate
Coordinated Universal Time (UTC)
Cryptographic Application Programming Interface (CAPI) or CryptoAPI
fully qualified domain name (FQDN)
Hypertext Transfer Protocol (HTTP)
private key
public key
Transmission Control Protocol (TCP)
UTF-8
Voice over IP (VoIP)
X.509
XML

The following terms are defined in [\[MS-OFCGLOS\]](#):

200 OK
Audio/Video Multipoint Control Unit (AVMCU)
base64 encoding
conference
conference URI (conference-URI)
Conference-Id
Content-Type header
federated user
focus
Focus Factory
in-band provisioning
lobby
MCU-Type
Multipoint Control Unit (MCU)
organizer
participant
public switched telephone network (PSTN)
Request-URI
SERVICE
Session Initiation Protocol (SIP)
SIP message
SIP request
SIP response

SIP response code
Transport Layer Security (TLS)
Uniform Resource Identifier (URI)

The following terms are specific to this document:

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

1.2 References

References to Microsoft Open Specifications documentation do not include a publishing year because links are to the latest version of the documents, which are updated frequently. References to other documents include a publishing year when one is available.

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. Please check the archive site, <http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624>, as an additional source.

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

[MS-SIPAE] Microsoft Corporation, "[Session Initiation Protocol \(SIP\) Authentication 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>

[RFC3852] Housley, R., "Cryptographic Message Syntax (CMS)", RFC 3852, July 2004, <http://www.ietf.org/rfc/rfc3852.txt>

1.2.2 Informative References

[MS-GLOS] Microsoft Corporation, "[Windows Protocols Master Glossary](#)".

[MS-LCID] Microsoft Corporation, "[Windows Language Code Identifier \(LCID\) Reference](#)".

[MS-OFCGLOS] Microsoft Corporation, "[Microsoft Office Master Glossary](#)".

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

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

[XML10] World Wide Web Consortium, "Extensible Markup Language (XML) 1.0 (Third Edition)", February 2004, <http://www.w3.org/TR/REC-xml>

1.3 Protocol Overview (Synopsis)

Organizers intending to invite **participants (2)** into a **conference** first create the conference at a **Focus Factory**. The Focus Factory provides a **conference URI (conference-URI)** that is subsequently used to join the conference through the **focus**. This protocol specifies the mechanism

by which a protocol client application interacts with the Focus Factory to create a new conference and additionally performs the following actions within the context of a specific organizer:

- Query for a list of created conferences that have not expired.
- Query for the details of a created conference.
- Modify the details of an existing conference.
- Delete an existing conference.
- Perform ancillary operations necessary to create a conference.

The distribution of the conference-URI to a participant (2) is outside the scope of this protocol.

1.4 Relationship to Other Protocols

The conferencing architecture described in [\[MS-CONFBAS\]](#) section 1.3.1 depends on this protocol for provisioning conferences.

The **SERVICE SIP request** and its corresponding **SIP response** are used for transmitting **SIP messages**.

The message syntax for the content of the SIP messages is described in [\[XML10\]](#).

1.5 Prerequisites/Preconditions

To send messages to the Focus Factory using **SIP**, protocol client applications first obtain the Focus Factory **Uniform Resource Identifier (URI)**. This URI can be obtained by using the **in-band provisioning** mechanism described in [\[MS-SIPREGE\]](#) section 2.2.2 and [\[MS-SIPREGE\]](#) section 3.3.

This protocol assumes that SIP requests are authenticated as described in [\[MS-SIPAE\]](#).

1.6 Applicability Statement

This protocol is meant to be used in conjunction with the conferencing architecture described in [\[MS-CONFBAS\]](#) section 1.3.1.

1.7 Versioning and Capability Negotiation

This protocol covers versioning issues in the following areas:

- **Supported Transports:** This protocol can be implemented using SIP on top of **Transmission Control Protocol (TCP)**, and **Transport Layer Security (TLS)**, as discussed in section [2.1](#).
- **Capability Negotiation:** This protocol supports capability negotiation through the `getConferencingCapabilities` request specified in section [3.1.4.8](#). A protocol client application can also determine that it is interfacing with a Focus Factory that supports its version by issuing a test request to create a conference, as specified in section [3.1.4.1](#).

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

Preliminary

2 Messages

2.1 Transport

This protocol does not introduce a new transport to exchange messages. It is capable of being used with any transport used by SIP. SIP messages can be transported over TCP or TLS.

2.2 Message Syntax

SIP messages used by this protocol can be categorized into two types – requests and responses. A request is sent from a protocol client application to the Focus Factory to perform an action and a response is sent from the Focus Factory to the protocol client application as a result of receiving a request.

Each request is specified as the body of a SERVICE SIP request. Each response is specified as the body of a SERVICE SIP response. The body MUST be well-formed **XML** and MUST be encoded using **UTF-8**. The XML MUST conform to the XML Schemas specified in [\[MS-CONFAS\]](#) section 7 and [\[MS-CONFAS\]](#) section 8. Any element or attribute appearing in the XML that is not specified by this protocol MUST be ignored by all processing entities, as long as such elements or attributes continue to keep the XML conformant with the specified XML Schemas. Any entity appearing in the referenced XML schemas, but not explicitly mentioned in this protocol, MUST NOT be used. The hierarchy of XML elements is specified for each message. The element hierarchy specifies parent elements, child elements, and the order in which the elements appear. An element is specified exactly once unless tagged as (?), (+), or (*).

- (?) indicates an element that can appear zero or 1 time.
- (+) indicates an element that can appear 1 or more times.
- (*) indicates an element that can appear zero or more times.

SERVICE SIP requests and SIP responses have the following SIP fields set as indicated:

- The **Content-Type header** MUST contain "application/cccp+xml".
- The **From** header MUST contain the Uniform Resource Identifier (URI) of the organizer or the URI of the organizer's delegate.
- If the **From** Header has the URI of the organizer's delegate, there SHOULD<1> be a **p-session-on-behalf-of** header that equals the organizer's URI. The authorization logic when delegation is used is specified in [\[MS-SIPAE\]](#).
- The **To** header MUST contain the Focus Factory URI.
- The **Request-URI** MUST contain the Focus Factory URI.
- A SIP request MUST NOT contain any **Require** headers.
- A SIP request MUST contain a body.
- A SIP response MUST contain a body unless the **SIP response code** indicates a failure, in which case the body is optional.

Requests and responses operate on a common conference Data Model as specified in section [2.2.1](#). Requests use the message syntax defined in section [2.2.2](#) and responses use the message syntax defined in section [2.2.3](#).

2.2.1 Conference Data Model

[\[MS-CONFAS\]](#) section 2.2.2 specifies an XML-based data model for describing conferences. This section provides an element hierarchy that specifies only the fields of that data model that the messages in this protocol operate on. The subsequent description specifies the interpretation of those fields as used in this protocol. Normative definitions for these XML elements are specified in [\[MS-CONFAS\]](#) section 7 and [\[MS-CONFAS\]](#) section 8.

All elements MUST belong to the namespace "urn:ietf:params:xml:ns:conference-info" unless explicitly stated to be in a different namespace or tagged as "(E)", indicating that they belong to an extension namespace. All attributes MUST belong to no namespace unless indicated as belonging to the extension namespace. The extension namespace MUST use the "http://schemas.microsoft.com/rtc/2005/08/confinfoextensions" URI.

Requests and responses can further restrict this data model. Any such restrictions are specified by their message syntaxes as needed.

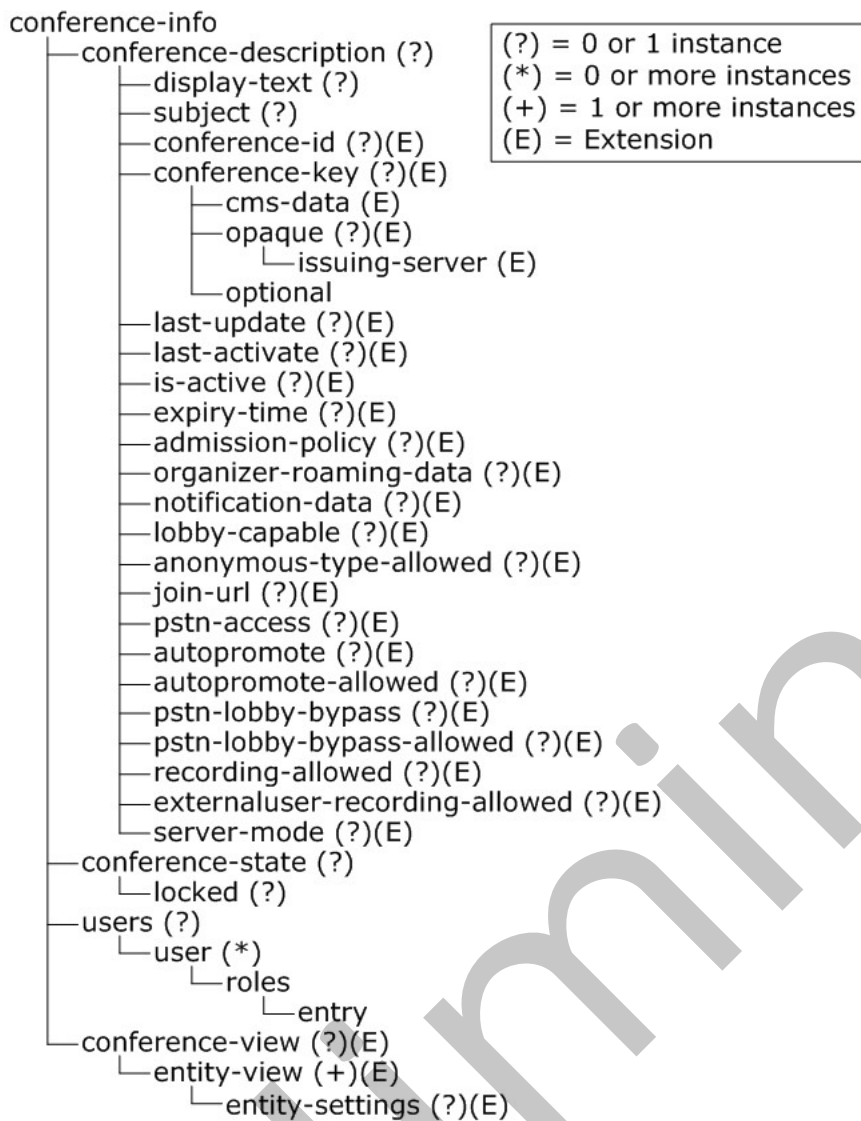


Figure 1: Conference entity hierarchy

2.2.1.1 conference-info Element

The **conference-info** element is the root element and MUST contain the following attribute:

- **entity:** This attribute MUST specify the conference-URI after the conference is created and MUST be set to an empty string before creation.

The **conference-info** element can contain the following attributes:

- **state:** This attribute MUST be either "full" or "partial". The value "full" indicates that the XML describing the conference contains all available conference data. The value "partial" indicates that the XML describing the conference only specifies a subset of all available conference data. When

not specified, the value of this attribute MUST default to "full". The available conference data includes the entire current state of the conference as specified by the conference data model.

- **version:** This attribute MUST be set to a monotonically increasing non-negative integer starting at "1". Its value MUST be set to "1" when the conference is first created and MUST increase by 1 every time the conference is modified further. When not specified, the value of this attribute defaults to "1".

2.2.1.2 conference-description Element

The **conference-description** element specifies some basic properties for the conference using the following child elements.

display-text: This element is used to specify a description for the conference.

subject: This element is used to specify a subject of the conference.

conference-id: Every conference MUST be identifiable by a string that is unique among all conferences belonging to an organizer. Constraints on the length and value space of this string are implementation-specific. Implementations SHOULD allow for a minimum of 8 and a maximum of 32 alpha-numeric characters from the ASCII set.

last-update: This element specifies the **Coordinated Universal Time (UTC)** date and time when the conference was created or last modified. The format of this value is defined by the XML Schema **dateTime** data type. The **dateTime** data type is defined in the internet standard XML specifications [\[XML10\]](#) and its related specifications. This element is not required for request, but MUST be specified in responses.

last-activate: This element specifies the **UTC** date and time when the conference was last activated in the Focus Factory. The format of this value is defined by the XML Schema **dateTime** data type. This element appears only in responses. The **dateTime** data type is defined in the internet standard XML specifications [\[XML10\]](#) and its related specifications.

is-active: This element indicates whether the conference is active in the Focus Factory at the time this value is set. The format of this value is defined by the XML schema Boolean data type where the value can be either true or false. This element appears only in responses. If "true", the conference is active, if "false", the conference is not active.

expiry-time: This element indicates the UTC time after which the conference can be deleted by the Focus Factory. An implementation SHOULD enforce a constraint on how long a conference is available, but the exact value of that upper bound is specific to the implementation. The format of this value is defined by the XML schema **dateTime** data type. When not specified, the default value is specific to the implementation. The **dateTime** data type is defined in the internet standard XML specifications [\[XML10\]](#) and its related specifications.

join-url: This element specifies the HTTP join URL for joining the meeting.

admission-policy: This element specifies the types of participants (2) that are allowed to join this conference and MUST contain one of the following values:

- "closedAuthenticated": All participants (2) MUST be on the pre-set roster that is provided at conference-creation time. Please see section [2.2.1.4](#) for more details on how to specify the preset roster.
- "openAuthenticated": All participants (2) MUST have an enterprise identity that is verified by the local enterprise or **federated users**.

- "anonymous": All participants (2) with a verified enterprise identity or any user with a valid anonymous passcode are permitted into the conference.

organizer-roaming-data: This element MUST contain well-formed XML. The root element of this XML MUST NOT belong to the namespace of the containing **organizer-roaming-data** element. This XML is meant for protocol client applications to store arbitrary contextual data for the conference with the Focus Factory. The server doesn't run any business logic based on this data. It is purely used by the organizer to store and roam data. An implementation SHOULD enforce constraints on the size of this XML. Limits are defined by the implementation, but SHOULD allow at least 4096 bytes (4 kilobytes).

notification-data: This element MUST contain well-formed XML. The root element of this XML MUST NOT belong to the namespace of the containing **notification-data** element. This XML is meant for protocol client applications to store arbitrary contextual data for the conference with the Focus Factory so that this data can later be provided to any participant (2) joining the conference. The server doesn't run any business logic based on this data. It is purely used by the organizer to store and roam data to other participants in the meeting. An implementation SHOULD enforce constraints on the size of this XML. Limits are defined by the implementation, but SHOULD allow at least 4096 bytes (4 kilobytes).

conference-key: This element specifies a case-sensitive string that MUST be presented by **anonymous users** joining the conference. The constraints on the length and value space of this string are implementation-specific. Implementations SHOULD allow a range from 8 up to 16 characters for the **conference-key** with value space restricted to printable ASCII characters. This string always appears in an encrypted format in any messages. The encrypted **conference-key** is specified by the **cms-data**, **opaque**, and **optional** child elements:

- **cms-data:** The value of this element MUST be an encrypted **conference-key** that is formatted by using the Cryptographic Message Syntax (CMS) and specified as a string that is encoded using **base64 encoding**. The CMS message SHOULD be specified with the data content type. For more details about CMS and the data content type, see [\[RFC3852\]](#).
- **opaque:** This element contains information that is specific to a Focus Factory and MUST NOT be interpreted by a client application. It MUST be obtained from the Focus Factory and MUST be specified in any messages exactly as it was received. The **opaque** element is an implementation-defined value specifying which encryption key the Focus Factory needs to use when decrypting **cms-data**. This element contains the following child element:
 - **issuing-server:** A string value that specifies the **fully qualified domain name (FQDN) (2)** of the server issuing the **public key** for encrypting the **conference-key**. Implementations SHOULD determine the **certificate (1)** to decrypt with solely from the **issuing-server**. If this is not possible, the implementation MUST introduce some other means of disambiguating the certificate (1) to use, such as an additional implementation-specific child element under **opaque**.
- **optional:** This element is a Boolean that indicates whether the conference PIN is optional to authorize audio-only users joining from the **PSTN.<2>** If "true", the conference PIN is optional else if "false" it is not optional. The conference PIN details are beyond the scope of this specification. It is used only by audio-only users joining from the PSTN and not used by com2007, com2007 R2.
- The **pstn-lobby-bypass** is a Boolean value. If "true", all the users, including external users, joining over the phone can join the conference without going to the lobby. If "false", they go to the lobby and wait for the organizer or presenter to bring them into the meeting<**3**>. The term "external user" is defined in [\[MS-SIPRE\]](#)

- The **autopromote** element is an unsigned integer. The server interprets this value to indicate who needs to be promoted to presenters upon being admitted to the conference. People in the lobby, once moved into the meeting, also follow these rules. The currently defined bits are [<4>](#):
 - None: 0x00000000 (as default)
 - Everyone: 0x80000000 (bit 31)
 - Company (Authenticated users): 0x00008000 (bit 15)
- The **server-mode** element specifies an integer that sets the compatibility mode for the conference [<5>](#). The value MUST be 13 or 14. If not specified, implementations SHOULD assume this value to be 13. The effect of the **server-mode** on conference behavior is specified in [\[MS-CONFBAS\]](#) section 2.2.2.3. A summary follows.
 - Server-mode 13
 - The created conference does not support the lobby experience. Meeting participants (2) that are placed in the "on-hold" state will leave the conference. More details on lobby and the "on-hold" state is specified in [\[MS-CONFBAS\]](#).
 - Application sharing is allowed by all participants (2), regardless of role.
 - Federated users are treated as authenticated users when applying admission-policy.
 - All users, except for the conference organizer and pre-invited authenticated presenters, are placed on-hold if they join while the conference is locked.
 - Server-mode 14
 - The created conference supports the lobby experience. Lobby-aware meeting participants (2) placed in an "on-hold" state will wait to be admitted. More details on lobby and the "on-hold" state is specified in [\[MS-CONFBAS\]](#).
 - Application sharing is allowed only by users with the "presenter" role.
 - Federated users are treated as anonymous users when applying admission-policy.
 - All users except for the conference organizer, are placed in an on-hold state if they join while the conference is locked.

2.2.1.3 conference-state Element

The **conference-state** element can have a **locked** element, which is a Boolean value. If set to "true", all the participants (2), both attendees and presenters, in the meeting go to the **lobby** when they join the meeting. Once the organizer has joined, the participants (2) can be explicitly moved from the lobby to the meeting by presenters. [<6>](#) If the value of **locked** element is set to "false" the meeting participants do not land in to the "lobby" as they enter the meeting. For more details of the conference-state element please see [\[MS-CONFBAS\]](#)

2.2.1.4 users Element

The **users** element specifies a list of participants (2) and their roles in the conference. When a **users** element is specified, it MUST contain zero or more **user** child elements. The list of users MUST be definitive if the **admission-policy** element is "closedAuthenticated" because a user is not allowed to join the conference if not specified in this list. The organizer is implicitly included in this list by the Focus Factory and SHOULD NOT be specified in the message.

A **user** element MUST specify the following attribute:

- **entity:** The SIP URI of the participant (2) being invited. This value MUST be unique among all **user** elements specified.

A **user** element MUST contain exactly one **roles** child element.

A **roles** element MUST contain exactly one **entry** child element:

- **entry:** The role to be assigned to the corresponding user. It MUST be either "presenter" or "attendee". For more details on the "presenter"/"attendee" privilege please see [\[MS-CONFAS\]](#).

2.2.1.5 conference-view Element

The **conference-view** element specifies a list of **MCU-Types** to be allocated for the conference, as well as initial bootstrap information to be provided for each **Multipoint Control Unit (MCU)**.

Each MCU-Type is specified by an **entity-view** child element. An **entity-view** element MUST contain the following attribute:

- **entity:** The MCU-Type to be made available for the conference. The value MUST be one of the MCU-Types configured for the Focus Factory. For information about retrieving the configured MCU-Types, see section [2.2.2.7](#) and [2.2.2.8](#). This value MUST be unique among all **entity-view** elements specified.

An **entity-view** element can specify the following child element:

- **entity-settings:** This element MUST contain an XML fragment. Each top-level element of the XML fragment MUST NOT belong to the namespace of the containing **entity-settings** element. This XML fragment is stored by the Focus Factory and passed to the MCU, of the corresponding MCU-Type, when being bootstrapped for the conference by the focus. It is only passed back to clients as part of the **getConference** response from the Focus Factory. For more information on this response, see section [2.2.3.4](#). Implementations SHOULD enforce a limit on the size of the **entity-settings** fragment. The size limit is defined by the implementation, but all implementations MUST accept **entity-settings** of at least 2048 bytes, in UTF-8 representation, to accommodate the formats for "audio-video", "meeting", and "phone-conf" MCU-Types defined later.

This protocol specifies the XML fragments for "audio-video", "meeting", and "phone-conf" MCU-Types.

An XML fragment MUST NOT be specified for the "chat" MCU-Type because this MCU-Type does not accept any bootstrap information.

For an "audio-video" MCU-Type, the XML fragment consists of one **settings** child element in the namespace "http://schemas.microsoft.com/rtc/2005/08/avconfinfoextensions". All child elements of this **settings** element MUST also belong to the same namespace. This **settings** element MUST use the following element hierarchy:

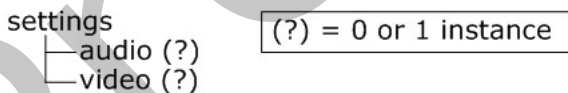


Figure 2: audio-video MCU-Type settings element hierarchy

- **audio:** This element MUST not specify any value. The existence of this element indicates that audio will be used in the conference. When not specified, the default is specific to the implementation.
- **video:** This element MUST not specify any value. The existence of this element indicates that video will be used in the conference. When not specified, the default is specific to the implementation.

For a "meeting" MCU-Type, the XML fragment consists of one **settings** child element in the namespace "http://schemas.microsoft.com/rtc/2005/08/dataconfinfoextensions". All child elements of this **settings** element MUST belong to the same namespace. This **settings** element MUST use the following element hierarchy:

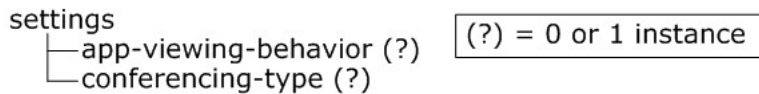


Figure 3: meeting MCU-Type settings element hierarchy

- **app-viewing-behavior:** This element contains a string value that specifies what form of application and desktop sharing to use for a conference. When a value is not specified, the default value is specific to the implementation. Valid values are as follows:
 - "disabled": Any form of application and desktop sharing is disabled for the conference.
 - "enableWithoutSharingControl": Application and desktop sharing are enabled without granting another participant (2) any remote interactivity with the shared region.
 - "enableWithSharingOfOnlyASingleApplication": Application sharing is enabled but desktop sharing is not allowed. Another participant (2) can be allowed to remotely interact with the shared region.
 - "enableWithFullSharing": Application and desktop sharing are enabled with the ability for other participants (2) to remotely interact with the shared region.
- **conferencing-type:** A string value indicating the type of conferencing the MCU is being used for. Values are "presentation" or "collaboration". When not specified, the default value is specific to the implementation.

For a "phone-conf" MCU-Type, the XML fragment consists of one **settings** child element in the namespace "http://schemas.microsoft.com/rtc/2005/08/acpconfinfoextensions". All child elements of this **settings** element MUST also belong to the same namespace. This **settings** element MUST use the following element hierarchy:

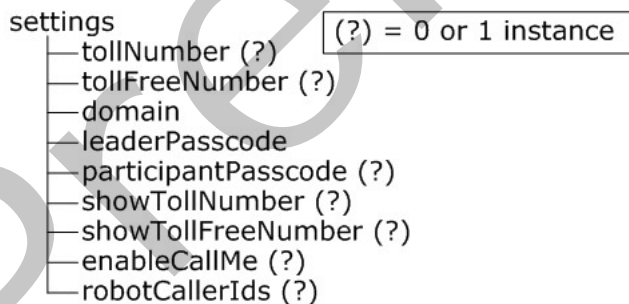


Figure 4: phone-conf MCU-Type settings element hierarchy

- **tollNumber:** A string value representing the telephone number for the phone conferencing provider. **tollNumber** or **tollFreeNumber** MUST be specified. The format of this string is specific to the phone conferencing provider being used.
- **tollFreeNumber:** A string value representing the toll-free telephone number for the phone conferencing provider. **tollNumber** or **tollFreeNumber** MUST be specified. The format of this string is specific to the phone conferencing provider being used.
- **domain:** A string value identifying the Internet domain of a phone conferencing provider.
- **leaderPasscode:** A string value representing the passcode to be entered by the organizer of the phone conference. The format of this string is specific to the phone conferencing provider being used.
- **participantPasscode:** A string value representing the passcode to be entered by participants (2) dialing in to the phone conference. The format of this string is specific to the phone conferencing provider being used.
- **showTollNumber:** Indicates whether **tollNumber** is published to everyone in the conference. The format of this value conforms to the XML Schema Boolean data type where the value can be either true or false.
- **showTollFreeNumber:** Indicates whether **tollFreeNumber** is published to everyone in the conference. The format of this value conforms to the XML Schema Boolean data type where the value can be either true or false.
- **enableCallMe:** Indicates whether a participant (2) in the conference is allowed to request the phone conferencing provider to make an outbound phone call to that participant's phone number. The format of this value conforms to the XML Schema Boolean data type where the value can be either true or false.
- **robotCallerIds:** A list of telephone numbers. A phone conferencing provider uses this list to identify callers that are not human and are considered a "robot". An example of a robot is an **Audio/Video Multipoint Control Unit (AVMCU)** that calls in to the phone conferencing provider for bridging audio between callers to the phone conferencing system and participants (2) connected to the AVMCU.

The **robotCallerIds** element can contain any number of **entry** child elements. Each **entry** child element MUST belong to the same namespace as **robotCallerIds**.

- **entry:** A string value representing a telephone number. The format of this string is specific to the phone conferencing provider being used.

2.2.2 Requests

A request MUST use the **request** root element. The **request** element and all child elements MUST belong to the namespace "urn:ietf:params:xml:ns:cccp" unless specified as belonging to an extension namespace. All attributes MUST belong to no namespace unless specified as belonging to the extension namespace. Elements and attributes belonging to the extension namespace MUST belong to the namespace "http://schemas.microsoft.com/rtc/2005/08/confinfoextensions".

The **request** element MUST contain the following attributes:

- **requestId:** MUST be set by the message sender to a non-negative integer that is unique among all requests from that sender.

- **from:** Specifies the source of the request and MUST be the same value as the organizer URI.
- **to:** Specifies the target of the request and MUST be the same value as the SIP URI contained in the **To** header of the enclosing SIP request.

The **request** element can contain the following attribute:

- **C3PVersion:** MUST be set to "1".

The **request** element MUST contain exactly one child element. The child element identifies the type of request. The different types of requests are subsequently defined.

2.2.2.1 addConference Request

An **addConference** request is used for creating a conference and is identified by a **request** element containing an **addConference** child element. The **addConference** child element MUST use the following element hierarchy:



Figure 5: addConference request element hierarchy

The syntax for the **conference-info** element is defined in section [2.2.1](#). The following additional restrictions apply:

- The **entity** attribute of **conference-info** MUST be an empty string.
- The **state** attribute of **conference-info** MUST be set to "full", if specified.
- The **version** attribute MUST be set to "1", if specified.
- The **conference-id** element MUST be specified.
- The **conference-key** element
 - MUST be specified when the **admission-policy** element is "anonymous" and the value of **server-mode** is "13".
 - SHOULD NOT be specified when the **admission-policy** element is not "anonymous"
 - SHOULD NOT be specified if the **server-mode** value is "14", even if the **admission-policy** element is "anonymous" [<7>](#).
- The **pstn-access** element, if specified, indicates that public switched telephone network (PSTN) bridging is enabled [<8>](#). PSTN bridging joins **Voice over IP (VoIP)** and PSTN dial-in audio channels so that all participants (2) in a meeting can hear each other. More details on the PSTN bridging is beyond the scope of this specification.

The **autopromote**

- element CAN be specified.
- The **server-mode** element CAN be specified.
- Static conferences are pre-created meetings on the server for which preset defaults are decided by the administrator. To create a static conference, send an **addConference** request. However,

unlike regular meetings, no properties of the meeting are specified. Only the **conference-id** is given, along with the attribute **static**, which is a Boolean value in the **conference-info** element that tells the Focus Factory to make it a static meeting if "true" <9>. If set to "false" a non-static meeting is created.

2.2.2.2 modifyConference Request

A **modifyConference** request is used to modify the details of an existing conference and is identified by a **request** element containing a **modifyConference** child element. The **modifyConference** child element MUST use the following element hierarchy:

```
modifyConference
└─conference-info
```

Figure 6: modifyConference request element hierarchy

The syntax for the **conference-info** element is exactly as specified for the **addConference** request with the following additional restrictions:

- The version attribute MUST be set to the current version of the conference.
- The effective **server-mode** value for the request MUST equal the current **server-mode** value for the conference. If not explicitly specified, the **server-mode** value SHOULD be assumed to be the same default assumed in section [2.2.2.1](#).

Note that static conferences cannot be modified because they are created on the server. The only operations that are allowed on static conferences are querying whether a static conference exists, creating a static conference, and deleting a static conference.

2.2.2.3 deleteConference Request

A **deleteConference** request is used to delete an existing conference and is identified by a **request** element containing a **deleteConference** child element. The **deleteConference** child element MUST use the following element hierarchy. To delete a static meeting, the client MUST send a **deleteConference** with static bit set.

```
deleteConference
└─conferenceKeys
```

Figure 7: deleteConference request element hierarchy

The **conferenceKeys** element indicates the conference to be deleted. This element MUST specify the following attributes:

- **conference-id**: This attribute is in the extension namespace and MUST be set to the **Conference-Id** of the conference to be deleted.
- **Static**: This attribute is an optional Boolean attribute that MUST be set to "true" to delete static conferences. For non-static conferences, the **static** attribute does not need to be specified.

2.2.2.4 getConference Request

A **getConference** request is used to obtain the details of an existing conference and is identified by a **request** element containing a **getConference** child element. The **getConference** child element MUST use the following element hierarchy

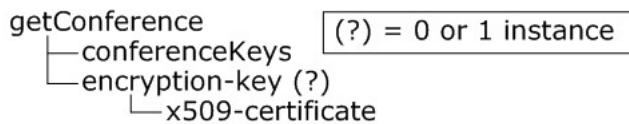


Figure 8: getConference request element hierarchy

The **conferenceKeys** element indicates the conference whose details are to be obtained. This element MUST specify the following attribute:

- **conference-id:** This attribute is in the extension namespace and MUST be set to the Conference-Id of the conference whose details are to be retrieved.

The **encryption-key** element, when specified, provides an **X.509** certificate (1) containing a public key. This element MUST specify the following child element:

- **x509-certificate:** The value of this element MUST be a string encoded with base64 encoding. The string MUST represent an X.509 certificate (1), MUST have a public key and MUST NOT have a **private key**.

2.2.2.5 getConferences Request

A **getConferences** request is used to obtain a list of conferences created by an organizer and is identified by a **request** element containing a **getConferences** child element. The **getConferences** element MUST have no child elements.

To locate the static conference, clients send a **getConferences** command with the **static** attribute, which is a Boolean value, set to "true". The **static** attribute is optional, with its default value being "false".

If the user sets static value as "true", the response of the command also has the **static** flag set to "true", to reassure clients that the server is returning static conferences. [<10>](#)

If the **getConferences** request has **static** as "false", the response of the command also has the **static** flag set to "false", to reassure clients that the server is returning non-static conferences. [<11>](#) .

2.2.2.6 getEncryptionKey Request

A **getEncryptionKey** request is used to obtain a certificate (1) from the Focus Factory containing a public key. The public key is used for encrypting sensitive information to be sent to the Focus Factory. Encrypting sensitive information is necessary to ensure that any logging being performed by intermediate servers that exist between the protocol client application and the Focus Factory will not be able to decipher the sensitive information. The **conference-key** is an example of a sensitive piece of information that needs to be transmitted in an encrypted fashion to the Focus Factory. The **conference-key** is considered sensitive information because obtaining this value could enable a potentially malicious recipient to join the associated conference as an anonymous user.

A **getEncryptionKey** request is identified by a **request** element containing a **getEncryptionKey** child element. The **getEncryptionKey** element MUST have no child elements.

2.2.2.7 getAvailableMcuTypes Request

A **getAvailableMcuTypes** request is used to obtain the MCU-Types available for use in a conference and is identified by a **request** element containing a **getAvailableMcuTypes** child

element. The **getAvailableMcuTypes** element MUST have no child elements. The **getAvailableMcuTypes** element SHOULD include the **server-mode** attribute <12>.

The **server-mode** attribute is an integer that affects which MCU-Types appear in the response. The value MUST be 13 or 14. If not specified, implementations SHOULD assume that this value is 13.

2.2.2.8 getConferencingCapabilities Request

This section follows the product behavior described in product behavior note <13>.

A **getConferencingCapabilities** request is used to obtain information about the capabilities available for use in a conference, as well as available MCU-Types. It is identified by a request element containing a **getConferencingCapabilities** child element. The **getConferencingCapabilities** element MUST have no child elements. The **getConferencingCapabilities** element SHOULD include the **server-mode** attribute <14>.

The **server-mode** attribute is an integer that affects which MCU-Types appear in the response. The value MUST be 13 or 14. If not specified, implementations SHOULD assume that this value is 13.

2.2.3 Responses

A response that is not a 400 SIP response MUST use the **response** root element. The **response** root element and all its child elements MUST belong to the namespace "urn:ietf:params:xml:ns:cccp" unless specified as belonging to an extension namespace. All attributes MUST belong to no namespace unless specified as belonging to the extension namespace. Elements and attributes belonging to an extension namespace MUST belong to the namespace "http://schemas.microsoft.com/rtc/2005/08/confinfoextensions", "http://schemas.microsoft.com/rtc/2008/12/confinfoextensions", or "http://schemas.microsoft.com/rtc/2005/08/cccpextensions".

The **response** element MUST contain the following attributes:

- **requestId**: MUST be set to the value obtained from the **requestId** attribute of the corresponding request.
- **from**: Specifies the source of the response and MUST be set to the value obtained from the **to** attribute of the corresponding request.
- **to**: Specifies the target of the response and MUST be set to the value obtained from the **from** attribute of the corresponding request.
- **code**: Indicates whether the corresponding request succeeded or failed and MUST be the string value "success" or "failure".

The **response** element can contain the following attribute:

- **C3PVersion**: MUST be set to "1".

The **response** element MUST contain exactly one child element. The child element identifies the type of response. The different types of responses are subsequently defined.

2.2.3.1 addConference Response

An **addConference** response indicates the status of the corresponding **addConference** request and is identified by a **response** element containing an **addConference** child element. The **addConference** child element MUST use the following element hierarchy:

addConference
└─conference-info (?) (?) = 0 or 1 instance

Figure 9: addConference response element hierarchy

The **addConference** element contains the following attribute:

- **Reason:** When the code attribute of the enclosing **response** element is "success", this attribute SHOULD NOT be used. When the code attribute of the enclosing **response** element is "failure", this attribute indicates the cause of the failure. The value of this attribute SHOULD be one of the following strings to specify the problem found in the corresponding request:
 - "anonymousUsersNotAllowed": The organizer is not authorized to create conferences with the admission policy set to "anonymous".
 - "conferenceExistsAlready": The specified **conference-id** element already exists for this organizer.
 - "entitySettingsTooLarge": The size of the XML in the **entity-settings** element exceeds the size allowed by the Focus Factory. The maximum size of the XML in the **entity-settings** element is implementation specific.
 - "federatedUsersNotAllowed": The organizer is not authorized to create conferences with federated users or the federation is not enabled.
 - "invalidAdmissionPolicy": The **admission-policy** element is missing.
 - "invalidConferenceId": The **conference-id** element does not follow the specified message syntax.
 - "invalidEncryptionKeyUsed": The certificate (1) used for encrypting the conference key is stale.
 - "invalidExpiryTime": The **expiry-time** element does not follow the specified message syntax.
 - "invalidPasscode": The conference-key could not be decrypted or does not follow the specified message syntax. This reason attribute is also returned when the conference-key element is not specified. [<15>](#)
 - "invalidRole": The **roles** element does not follow the specified message syntax.
 - "invalidUserEntity": The **entity** attribute of the **user** element does not follow the specified message syntax.
 - "invalidVersion": The value of the **version** attribute of the **conference-info** element is not equal to the current version of the conference.
 - "maxConferencesExceeded": The organizer has exceeded the Focus Factory quota for the maximum number of conferences allowed. The maximum number of conferences is implementation specific.
 - "mcuTypeNotAvailable": The **mcuType** value specified by the **entity** attribute of each **entity-view** element in the **conference-view** element is not available. **mcuType** is specified in detail in [\[MS-CONFBAS\]](#).
 - "notificationDataTooLarge": The size of the XML in the **notification-data** element exceeds the size allowed by the Focus Factory. The maximum size of the XML in the **notification-data** element is implementation specific.

- "organizerRoamingDataTooLarge": The size of the XML in the **organizer-roaming-data** element exceeds the size allowed by the Focus Factory. The maximum size of the XML in the **organizer-roaming-data** element is implementation specific.
- "maxStaticMeetingsExceeded": The number of static meetings created exceeds the size allowed. Note that there can only be one static meeting per user.
- "invalidStaticMeetingRequest": The static meeting request does not follow the specified message syntax.
- "invalidAutopromoteValue": The value for **autopromote** is not supported by the server.
- "pstnLobbyBypassNotAllowed": The user is not allowed to create conferences where PSTN users can bypass the lobby.
- "pstnBridgeNotEnabled": The user attempted to create a conference with **pstn-access** specified but is not allowed to create PSTN conferences.
- "pstnMeetingIdCannotBeSpecified": The **pstn-access** element does not follow the specified message syntax as specified in [MS-CONFAS].
- "otherFailure": The cause of the failure is not clearly established. This reason MUST be used when a failure is not caused by one of the other reasons outlined here (for example, if it is caused by memory exhaustion on the server). It SHOULD NOT be used if another reason describes the failure.

The **conference-info** element MUST be specified when the **code** attribute of the enclosing **response** element is "success" and MUST NOT be specified when the **code** attribute is "failure". The syntax for the **conference-info** element is defined in section [2.2.1](#). The following additional restrictions apply to the **conference-info** element:

- The entity attribute MUST be the conference-URI.
- The state attribute MUST be specified and set to "partial".
- The version attribute MUST be specified and set to "1".
- The conference-description child elements SHOULD be specified. Versions prior to LYNC2010 did not specify any child elements.

2.2.3.2 modifyConference Response

A **modifyConference** response indicates the status of the corresponding **modifyConference** request and is identified by a **response** element containing a **modifyConference** child element. The **modifyConference** child element MUST use the following element hierarchy:

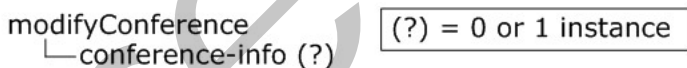


Figure 10: modifyConference response element hierarchy

The syntax for the **modifyConference** element and its **conference-info** child element is exactly as defined for the **addConference** response with the additional restriction that the **version** attribute MUST be set to the current version of the conference. Also the **reason** attribute SHOULD be the string "conferenceDoesNotExist" when the conference being modified does not exist at the Focus Factory.

2.2.3.3 deleteConference Response

A **deleteConference** response indicates the status of the corresponding **deleteConference** request and is identified by a **response** element containing a **deleteConference** child element. The **deleteConference** element SHOULD have no child elements.

The **deleteConference** element can contain the following attribute:

- **reason**: When the **code** attribute of the enclosing **response** element is "success", this attribute SHOULD NOT be used. When the **code** attribute of the enclosing **response** element is "failure", this attribute indicates the cause of the failure. The value of this attribute SHOULD be one of the following strings to specify the problem found in the corresponding request:
 - "conferenceDoesNotExist": The conference being deleted does not exist at the Focus Factory.
 - "staticFlagDoesntMatch": An attempt was made to delete a static meeting without setting the **static** attribute to "true" or an attempt was made to delete a non-static meeting with the **static** attribute set to "true".
 - "otherFailure": The cause of the failure is not clearly established. This reason MUST be used when a failure is not caused by one of the other reasons outlined here (for example, if it is caused by memory exhaustion on the server). It SHOULD NOT be used if another reason describes the failure.

2.2.3.4 getConference Response

A **getConference** response indicates the status of the corresponding **getConference** request and returns the requested details. A **getConference** response is identified by a **response** element containing a **getConference** child element. The **getConference** child element MUST use the following element hierarchy:

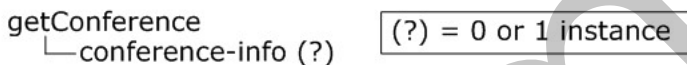


Figure 11: getConference response element hierarchy

The **getConference** element can contain the following attribute:

- **reason**: When the **code** attribute of the enclosing **response** element is "success", this attribute SHOULD NOT be used. When the code attribute of the enclosing **response** element is "failure", this attribute indicates the cause of the failure. The value of this attribute SHOULD be one of the following strings to specify the problem found in the corresponding request:
 - "conferenceDoesNotExist": The conference does not exist at the Focus Factory.
 - "invalidEncryptionKey": The specified **encryption-key** cannot be used.
 - "otherFailure": The cause of the failure is not clearly established. This reason MUST be used when a failure is not caused by one of the other reasons outlined here (for example, if it is caused by memory exhaustion on the server). It SHOULD NOT be used if another reason describes the failure.

The **conference-info** element MUST be specified when the **code** attribute of the enclosing **response** element is "success" and MUST NOT be specified when the **code** attribute is "failure". The syntax for the **conference-info** element is defined in section [2.2.1](#). The following additional restrictions apply to the **conference-info** element:

- The **entity** attribute MUST be set to the conference-URI.
- The **state** attribute MUST be specified and set to "full" if the **conference-key** does not exist or the **conference-key** exists and the **encryption-key** element was specified in the corresponding request. The **state** attribute MUST be set to "partial" otherwise.
- The **version** attribute MUST be specified and set to the current version of the conference.
- The **conference-key** element of **conference-description** SHOULD be specified when the state attribute is "full" and conference-key exists.
- The **last-activate** element SHOULD be specified if the conference has ever been activated by a participant (2). It SHOULD NOT be specified, otherwise.
- The **is-active** element SHOULD be specified when the value is "true". It SHOULD NOT be specified otherwise.
- The **organizer-roaming-data** and **notification-data** elements MUST be specified if those elements were present in the conference creation request. They MUST NOT be specified otherwise.

2.2.3.5 getConferences Response

A **getConferences** response indicates the status of the corresponding **getConferences** request and returns the list of conferences created by the organizer that sent the request. A **getConferences** response is identified by a **response** element containing a **getConferences** child element. The **getConferences** child element MUST use the following element hierarchy:



Figure 12: getConferences response element hierarchy

The **getConferences** element can contain the following attribute:

- **reason:** When the **code** attribute of the enclosing **response** element is "success", this attribute SHOULD NOT be used. When the **code** attribute of the enclosing **response** element is "failure", this attribute indicates the cause of the failure. The value of this attribute SHOULD be one of the following strings to specify the problem found in the corresponding request:
 - "otherFailure": The cause of the failure is not clearly established. This reason MUST be used when a failure is not caused by one of the other reasons outlined here (for example, if it is caused by memory exhaustion on the server). It SHOULD NOT be used if another reason describes the failure.

The **conferences** element contains **conference-info** child elements, one for each conference. The syntax for each **conference-info** element is defined in section [2.2.1](#). The following additional restrictions apply to the **conference-info** element:

- The **entity** attribute MUST be set to the conference-URI.
- The **state** attribute MUST be specified and set to "partial".
- The **version** attribute MUST be specified and set to the current version of the conference.

- The **conference-id**, **admission-policy** elements of **conference-description** MUST be specified.

2.2.3.6 getEncryptionKey Response

A **getEncryptionKey** response indicates the status of the corresponding **getEncryptionKey** request and returns the Focus Factory certificate (1) containing a public key for encryption. A **getEncryptionKey** response is identified by a **response** element containing a **getEncryptionKey** child element. The **getEncryptionKey** child element MUST use the following element hierarchy:

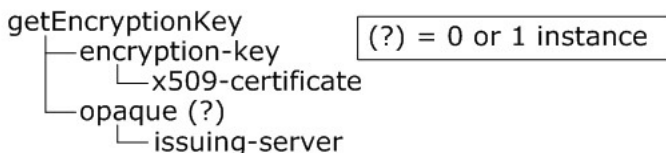


Figure 13: getEncryptionKey response element hierarchy

The **getEncryptionKey** element can contain the following attribute:

- **reason:** When the **code** attribute of the enclosing **response** element is "success", this attribute SHOULD NOT be used. When the **code** attribute of the enclosing **response** element is "failure", this attribute indicates the cause of the failure. The value of this attribute SHOULD be one of the following strings to specify the problem found in the corresponding request:
 - "otherFailure": The cause of the failure is not clearly established. This reason MUST be used when a failure is not caused by one of the other reasons outlined here (for example, if it is caused by memory exhaustion on the server). It SHOULD NOT be used if another reason describes the failure.

The **getEncryptionKey** element MUST contain an **encryption-key** child element and can contain an **opaque** element, both of which are in the extension namespace.

The **encryption-key** element provides an X.509 certificate (1) containing a public key. This element MUST specify the following child element:

- **x509-certificate:** The value of this element MUST be a string encoded with base64 encoding. The string MUST represent an X.509 certificate (1) with only a public key and no private key.

The **opaque** element contains information that is specific to a Focus Factory and MUST NOT be interpreted by a protocol client application. This element contains the following child element:

- **issuing-server:** This element contains a string value that specifies the FQDN (2) of the server issuing the public key for encrypting the conference key.

2.2.3.7 getAvailableMcuTypes Response

A **getAvailableMcuTypes** response indicates the status of the corresponding **getAvailableMcuTypes** request and returns the list of MCU-Types available at the Focus Factory. A **getAvailableMcuTypes** response is identified by a **response** element containing a **getAvailableMcuTypes** child element. The **getAvailableMcuTypes** child element MUST use the following element hierarchy:

```

getAvailableMcuTypes
├─ mcu-types
│   └─ mcuType (*)

```

(*) = 0 or more instances

Figure 14: getAvailableMcuTypes response element hierarchy

The **getAvailableMcuTypes** element can contain the following attribute:

- **reason:** When the **code** attribute of the enclosing **response** element is "success", this attribute SHOULD NOT be used. When the **code** attribute of the enclosing **response** element is "failure", this attribute indicates the cause of the failure. The value of this attribute SHOULD be one of the following strings to specify the problem found in the corresponding request:
 - "otherFailure": The cause of the failure is not clearly established. This reason MUST be used when a failure is not caused by one of the other reasons outlined here (for example, if it is caused by memory exhaustion on the server). It SHOULD NOT be used if another reason describes the failure.

The **getAvailableMcuTypes** element MUST contain an **mcu-types** child element. An **mcuTypes** element contains zero or more **mcuType** elements.

mcuType: A string value that represents a type of MCU that can be used in a conference. The **mcuType** has the same value space as the **purpose** subelement of the **conf-uris**. [MS-CONFAS] section 2.2.2.4 specifies the allowed values. If the **server-mode** value for the request is 14, this list MUST NOT include the **mcuType** "meeting"<16>. If the **server-mode** value for the request is 13, this list MUST NOT include the **mcuType** "data-conf".

2.2.3.8 getConferencingCapabilities Response

This section follows the product behavior described in product behavior note<17>.

A **getConferencingCapabilities** response indicates the status of the corresponding **getConferencingCapabilities** request and returns conferencing capabilities for the Focus Factory along with a list of MCU-Types available at the Focus Factory. A **getConferencingCapabilities** response is identified by a **response** element containing a **getConferencingCapabilities** child element. The **getConferencingCapabilities** child element MUST use the following element hierarchy.

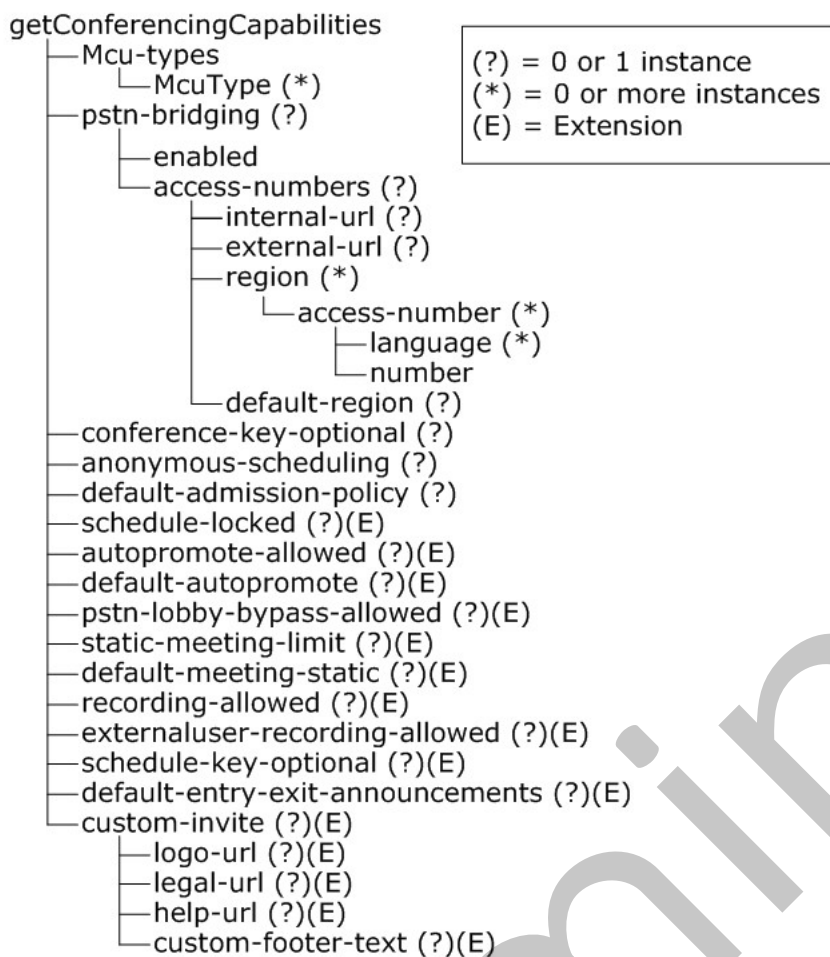


Figure 15: getConferencingCapabilities element hierarchy

The **getConferencingCapabilities** element can contain **reason** and **capability-version** attributes.

- **reason:** When the **code** attribute of the enclosing **response** element is "success", this attribute SHOULD NOT be used. When the **code** attribute of the enclosing **response** element is "failure", this attribute indicates the cause of the failure. The value of this attribute SHOULD be one of the following strings to specify the problem found in the corresponding request:
 - "otherFailure": The cause of the failure is not clearly established. This reason MUST be used when a failure is not caused by one of the other reasons outlined here (for example, if it is caused by memory exhaustion on the server). It SHOULD NOT be used if another reason describes the failure.
- **capability-version:** A non-negative integer value specifying the set of capabilities advertised in this response. When the **code** attribute of the enclosing **response** element is "success", this attribute SHOULD be used. A Focus Factory compliant to this revision of this protocol SHOULD specify 0 when this attribute is used. The **capability-version** is not used when the **code** attribute of the enclosing **response** element is not "success".

The **getConferencingCapabilities** element MUST contain an **mcu-types** child element and an **anonymous-scheduling** element and can contain all of the other child elements. An **mcu-types** element contains zero or more **mcuType** elements.

- **mcuType**: A string value that represents a type of MCU that can be used in a conference. The **mcuType** element has the same value space as the **purpose** subelement of the **conf-uris** element. For specification of the allowed values, see [\[MS-CONFAS\]](#) section 2.2.2.4. If the **server-mode** value for the request is 14, this list MUST NOT include the **mcuType** "meeting". If the **server-mode** value for the request is 13, this list MUST NOT include the **mcuType** "data-conf".

anonymous-scheduling: A Boolean value that represents whether the Focus Factory can create conferences with the "anonymous" value for **admission-policy**. For details of the **admission-policy**, see section [2.2.1.2](#). If the value is **true**, the Focus Factory MUST allow "anonymous" as a value for **admission-policy**. If **false**, the Focus Factory MUST NOT allow "anonymous" as a value for **admission-policy**.

conference-key-optional: This Boolean value indicates whether the client can create conferences with the **optional** element of the **conference-key** enabled. If **true**, the conference-key is optional, else if **false**, it is not optional.

default-admission-policy: This value specifies an admission-policy that clients SHOULD use as the default when creating a new conference. It can be overridden by user input.

schedule-locked: A Boolean value that, if **true**, indicates that the client can schedule locked meetings. [<18>](#) If **false**, indicates that the client cannot schedule locked meetings.

schedule-key-optional: A Boolean value that, if **true**, indicates that the client can schedule meetings without a **conference-key** specified. If **false**, indicates that the client needs to specify a **conference-key** when scheduling a conference. [<19>](#)

recording-allowed: A Boolean value that represents whether recording is allowed for the conference. [<20>](#) If **true**, recording is allowed, else if **false**, it is not allowed.

externaluser-recording-allowed: A Boolean value that, if **false**, indicates that an external user cannot record during the conference. If **true**, indicates that an external user can record during the conference. This value is applicable only if **recording-allowed** is **true**. [<21>](#) The term "external user" is defined in [MS-SIPRE].

autopromote-allowed: An unsigned integer that indicates the supported levels of people who can be automatically escalated to presenter when they join the meeting. This value is a bitmask. Bits with value 1 in the binary representation of the number indicate classes of participants (2) who are escalated in this manner. If a particular bit is set, all less significant bits defined SHOULD also be set. Bits not defined by this specification are reserved for future use and MUST be 0. The allowed masks are as follows: [<22>](#)

- None: 0x00000000 (as default)
- Everyone: 0x80000000 (bit 31)
- Company (Authenticated users): 0x00008000 (bit 15)

default-autopromote: An unsigned integer that indicates what the default autopromote value is if it is not specified in **addConference** or **modifyConference**. The bit mask values are the same as those specified for **autopromote-allowed**. [<23>](#)

pstn-lobby-bypass-allowed: A Boolean that indicates whether the server supports PSTN users bypassing the lobby when they join the meeting. If the value is **true**, the client can enable this bypass. If **false**, the client cannot enable this bypass. <24>

static-meeting-limit: This is an integer value that indicates the number of static meetings the user can have. The maximum value is 1. <25>

default-meeting-static: This Boolean value indicates whether the default meeting type is static. If this value is **true**, the client uses the same preexisting static meeting coordinates in all meeting invites. If this value is **false**, by default, the client creates a new meeting. How to create a conference is described in section 3.1.4.1. <26>

default-entry-exit-announcements: This Boolean value indicates the recommended default that the client uses for the entry-exit global announcements feature when creating a meeting. If this value is **true**, the client will, by default, turn on the entry-exit announcement feature when scheduling meetings. If **false**, the client will turn off the entry-exit announcement feature. End-user or client can always choose a non-default value for this feature. <27>

The **pstn-bridging** element MUST contain the following child elements:

- **enabled:** A Boolean value that indicates whether the server supports PSTN bridging. If **true**, the PSTN bridging is supported, else if **false**, it is not supported.
- **access-numbers:** A container for links and access numbers related to dial-in from the PSTN. It SHOULD contain the following child elements:
 - **internal-url:** A **Hypertext Transfer Protocol (HTTP)** link to a location where clients can get more information about the PSTN bridging numbers available to them if they are in the corporate network.
 - **external-url:** An HTTP link to a location where clients can get more information about the PSTN bridging numbers available to them if they are outside of the corporate network.
 - **region:** A group of access numbers applicable in the same geographic region. A region MUST contain the name attribute, which is a string giving a name for the region. A region MUST contain the following child elements:
 - **access-number:** An individual access number for PSTN access. An access-number MUST contain the following elements:
 - **language:** A representation of a language that the access number can interact with users in. The first language listed SHOULD indicate the default language used by the access number. A language MUST have a tag attribute that gives the identity of the language, as a value in the XML schema type for language. It MUST also have an **lcid** attribute, which gives the language as an unsigned integer, in the value space defined by [\[MS-LCID\]](#).
 - **number:** A string value giving a human readable phone number at which the access number can be reached.
 - **default-region:** A string value giving the client a hint about which region of access numbers to emphasize in a user interface. This value SHOULD equal the **name** of the **region** that most closely matches the user's usual physical locale.

custom-invite: The custom-invite element MAY contain the following child elements: <28>

- **logo-url:** An HTTP link to a location where clients can get a logo to render for the customized invite. [<29>](#)
- **legal-url:** An HTTP link that clients can render for legal information for the customized invite. [<30>](#)
- **help-url:** An HTTP link that clients can render for help information for the customized invite. [<31>](#)
- **custom-footer-text:** A string value that clients can use to render a custom footer for the customized invite. [<32>](#)

Preliminary

3 Protocol Details

3.1 Client Details

This section documents protocol client application details.

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.

Outstanding Request Queue (ORQ): A list of request IDs, one for each request sent to the Focus Factory for which a response has not yet been received.

3.1.2 Timers

Request Timer: A type of timer, an instance of which is associated with each request ID in the ORQ. It SHOULD go off 32 seconds after the request is placed in the ORQ.

3.1.3 Initialization

Before any requests can be sent to the Focus Factory an implementation MUST obtain the organizer SIP URI and Focus Factory URI because those values are necessary for constructing requests.

3.1.4 Higher-Layer Triggered Events

3.1.4.1 Create a Conference

When an organizer intends to create a conference, the protocol client application MUST use the following steps:

1. Create a unique Conference-Id for this organizer.
2. If a **conference-key** is provided, the protocol client application MUST:
 1. Obtain a certificate (1) for encryption from the Focus Factory, as specified in section [3.1.7.2](#). Also obtain the **opaque** element if specified by the Focus Factory.
 2. Use the certificate (1) to create a Cryptographic Message Syntax string encoded with base64 encoding containing the encrypted conference key. For more details about the Cryptographic Message Syntax, see [\[RFC3852\]](#).
3. Send an **addConference** request using the steps specified in section [3.1.7.1](#) and wait for an **addConference** response.

When an organizer intends to create a static conference, the protocol client application MUST use the following steps:

1. Check if the static conference is already created by doing a GetConferences call with the static flag set to "true".

2. If the static conference exists, use the conference-id returned in the response to do a `GetConference` call to get the details of the static meeting.
3. If the static meeting does not exist, do an **addConference** call with **static** set to "true" to create a static meeting.

3.1.4.2 Modify a Conference

When an organizer intends to modify a conference, the protocol client application MUST use the following steps:

1. Obtain all current details of the conference using the steps in section [3.1.4.5](#).
2. Obtain the current version of the conference from the current conference details.
3. Obtain the current **server-mode** of the conference from the current conference details [<33>](#).
4. Ensure that the version of the conference specified by the higher layer equals the current version. If it does not, stop processing, provide the higher layer the current conference details, and have the higher layer reissue the request. Version mismatches can happen when the organizer uses the protocol client application from multiple locations and all locations have not synchronized to the latest version.
5. Perform the same steps as those used to create a conference in section [3.1.4.1](#) with the following exceptions:
6. Send a **modifyConference** request instead of an **addConference** request.
7. Set the **version** attribute of **conference-info** to the value obtained in Step 2.
8. Set the **server-mode** element of the conference-description to the value obtained in step [3<34>](#).
9. Wait for a **modifyConference** response.

3.1.4.3 Delete a Conference

When an organizer intends to delete a conference, the protocol client application MUST send a **deleteConference** request using the steps specified in section [3.1.7.1](#) and wait for a **deleteConference** response.

To delete a static conference, make sure that the optional **static** attribute is set to "true" in the delete request.

3.1.4.4 Query for a List of Conferences

To obtain a list of the conferences of an organizer, the protocol client application MUST send a **getConferences** request using the steps specified in section [3.1.7.1](#) and wait for a **getConferences** response.

To obtain information about the static conference of an organizer, the protocol client MUST send a **getConferences** request with the **static** flag set to "true" and wait for the response.

3.1.4.5 Query for the Details of a Conference

To obtain the details of an existing conference of an organizer, the protocol client application MUST use the following steps:

1. Create a self-signed certificate (1) as a string encoded with base64 encoding containing the certificate (1) formatted as specified in section [3.1.7.3](#), using the Cryptographic Message Syntax as specified in [\[RFC3852\]](#).
2. Create an appropriate **getConference** request.
3. Assign the self-signed certificate (1) to the **x509-certificate** child element of the **encryption-key** element.
4. Send a **getConference** request using the steps specified in section [3.1.7.1](#) and wait for a **getConference** response.

Optionally, if the organizer does not need to obtain the **conference-key**, a **getConference** request can be issued without passing a self-signed certificate (1), thus avoiding step 1 and step 3.

3.1.4.6 Obtain an Encryption key

When an organizer intends to get the encryption key of the conference, the protocol client application MUST send a **getEncryptionKey** request using the steps specified in section [3.1.7.1](#) and wait for a **getEncryptionKey** response.

3.1.4.7 Obtain a List of MCU-Types

When creating a conference, an organizer can explicitly specify the MCU-Types to be made available for the conference and, optionally, any MCU-specific settings. Before such a conference can be created, the client application MUST first retrieve the list of MCU-Types available from the Focus Factory. This list is obtained by sending a **getAvailableMcuTypes** request using the steps specified in section [3.1.7.1](#) and waiting for a **getAvailableMcuTypes** response, or by sending a **getConferencingCapabilities** request [<35>](#) using the steps specified in section [3.1.7.1](#) and waiting for a **getConferencingCapabilities** response.

The application SHOULD indicate the **server-mode** it intends to schedule a conference in, so that appropriate MCU-Types are returned [<36>](#).

3.1.4.8 Obtain Focus Factory Capabilities

This section follows the product behavior described in product behavior note [<37>](#).

When creating a conference, the protocol client application can discover the capabilities of the Focus Factory. The capabilities of the Focus Factory are obtained by sending a **getConferencingCapabilities** request using the steps specified in section [3.1.7.1](#) and waiting for a **getConferencingCapabilities** response.

The application SHOULD indicate the **server-mode** it intends to schedule a conference in, so that appropriate MCU-Types are returned [<38>](#).

3.1.5 Message Processing Events and Sequencing Rules

3.1.5.1 Processing a SIP Response

A **200 OK** SIP response code indicates that the request was successfully processed by the Focus Factory. Any other response code indicates that the Focus Factory failed to process the request. When the response code is 200 OK, the body of the SIP response MUST be set. For any other response code, the body of the SIP response can be set. If the body is not set, return the failure SIP response code to the higher layer. When the body is set it MUST be one of the messages specified in section [2.2.3](#). The subsequent section specifies how such a response is processed.

3.1.5.2 Processing a Response

A protocol client application SHOULD use the following steps when processing a response:

1. Retrieve the body of the SIP response and parse to verify that it is well-formed XML.
2. Validate that the root element of the XML conforms to the response message syntax specified in section [2.2.3](#).
3. Retrieve the value of the **requestId** attribute from the **response** element. Verify and remove the specified **requestId** from the ORQ.
4. Identify the type of response by comparing the child element of the **response** element to the list of valid responses specified in section [2.2.3](#) and validate that the XML conforms to the message syntax for that response type.

If any of the preceding steps fail, stop further processing and notify the higher layer that the corresponding SIP request failed.

Record whether the corresponding request succeeded or failed by reading the **code** attribute of the **response** element.

The processing of each type of response is specified in subsequent sections of this protocol.

3.1.5.3 Processing an addConference Response

A protocol client application SHOULD use the following when processing an **addConference** response:

- If the response indicated failure, obtain the cause by reading the **reason** attribute from the **addConference** element, stop further processing, and indicate failure and return the reason to the higher layer.
- If the response indicated success, read the conference-URI from the **entity** attribute of the **addConference** element, return it to the higher layer, and indicate that the corresponding request succeeded.

3.1.5.4 Processing a modifyConference Response

A protocol client application SHOULD use the exact same steps as those specified in section [3.1.5.3](#) for processing an **addConference** response, with the exception that it references the **modifyConference** element instead of the **addConference** element wherever specified.

3.1.5.5 Processing a deleteConference Response

A protocol client application SHOULD use the following when processing a **deleteConference** response:

- If the response indicated failure, obtain the cause by reading the **reason** attribute from the **deleteConference** element, stop further processing, and indicate failure and return the reason to the higher layer.
- If the response indicated success, return success to the higher layer.

3.1.5.6 Processing a getConference Response

A protocol client application SHOULD use the following when processing a **getConference** response:

- If the response indicated failure, obtain the cause by reading the **reason** attribute from the **getConference** element, stop further processing, and indicate failure and return the reason to the higher layer.
- If the response indicated success, if a self-signed certificate (1) was specified in the corresponding request and an encrypted **conference-key** is specified in the response, obtain the decrypted string by using the locally stored private key for the self-signed certificate (1) sent in the corresponding **getConference** request. Return success to the higher layer along with the details retrieved from the **getConference** response.

3.1.5.7 Processing a getConferences Response

A protocol client application SHOULD use the following when processing a **getConferences** response:

- If the response indicated failure, obtain the cause by reading the **reason** attribute from the **getConferences** element, stop further processing, and indicate failure and return the reason to the higher layer.
- If the response indicated success, return success to the higher layer, along with the list of conferences obtained from the **getConferences** response.

3.1.5.8 Processing a getEncryptionKey Response

A protocol client application SHOULD use the following when processing a **getEncryptionKey** response:

- If the response indicated failure, obtain the cause by reading the **reason** attribute from the **getEncryptionKey** element, stop further processing, and indicate failure and return the reason to the higher layer.
- If the response indicated success, return success to the higher layer along with the values retrieved from the **encryption-key** element of the **getEncryptionKey** response.

3.1.5.9 Processing a getAvailableMcuTypes Response

A protocol client application SHOULD use the following processing a **getAvailableMcuTypes** response:

- If the response indicated failure, obtain the cause by reading the **reason** attribute from the **getAvailableMcuTypes** element, stop further processing, and indicate failure and return the reason to the higher layer.
- If the response indicated success, return success to the higher layer along with the list of MCU-Types obtained from the **mcu-types** element of the **getAvailableMcuTypes** response.

3.1.5.10 Processing a getConferencingCapabilities Response

This section follows the product behavior described in product behavior note [<39>](#).

A protocol client application SHOULD use the following when processing a **getConferencingCapabilities** response:

- If the response indicated failure, obtain the cause by reading the **reason** attribute from the **getConferencingCapabilities** element, stop further processing, and indicate failure and return the reason to the higher layer.
- If the response indicated success, return success to the higher layer, along with the anonymous scheduling capability flag obtained from the **anonymous-scheduling** child element of the **getConferencingCapabilities** element, and the list of MCU-Types obtained from the **mcu-types** child element of the **getConferencingCapabilities** element.

3.1.6 Timer Events

When a Request Timer goes off, if no SIP response for the associated request ID was received in that time interval, the corresponding request ID SHOULD be removed from the ORQ and the higher layer SHOULD be notified that the corresponding request timed out. If a SIP response was received for the associated request ID, do nothing.

3.1.7 Other Local Events

3.1.7.1 Send a Request

To send a request, the following steps SHOULD be followed:

1. Create an appropriate request using the message syntax specified for that request in section [2.2.2](#).
2. Send the request.
3. Store the **requestId** from the corresponding request in the ORQ.

3.1.7.2 Obtain a Focus Factory Certificate for Encryption

To obtain the Focus Factory certificate (1) for encryption, the protocol client application MUST send a **getEncryptionKey** request using the steps specified in section [3.1.7.1](#) and wait for a **getEncryptionKey** response.

3.1.7.3 Create a Self-signed Certificate

The protocol client application SHOULD use any available **Cryptographic Application Programming Interface (CAPI) or CryptoAPI** on the local computer to create a self-signed certificate (1). The returned certificate (1) MUST contain only the public key. The private key MUST be stored safely for later decryption.

3.2 Focus Factory Details

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.

Conference Table: Stores a list of conferences indexed by the pair (organizer, Conference-Id). For each conference, it stores all fields specified in the Conference Data Model from section [2.2.1](#).

MCU Table: Stores a list of MCUs that are indexed by their MCU-Types. For each MCU, it stores which **server-mode** values the MCU is expected to be used with.

Focus Factory Certificate: A certificate (1) used for encrypting data sent from the protocol client application to the Focus Factory.

3.2.2 Timers

Conference Expiration Timer: A timer that SHOULD fire periodically at an implementation-specific interval.

3.2.3 Initialization

Initialize the Focus Factory certificate (1) to a certificate (1) configured for use by the Focus Factory.

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Message Processing Events and Sequencing Rules

3.2.5.1 Processing a SIP Request

The Focus Factory SHOULD take the following steps when processing any SIP request:

1. Retrieve the body of the SIP request and parse it to verify that it is well-formed XML.
2. Validate that the root element of the XML conforms to the request message syntax specified in section [2.2.2](#).
3. Identify the type of request by comparing the child element of the **request** element to the list of valid requests specified in section [2.2.2](#) and validate that the XML conforms to the message syntax for that request type.

If any of the steps in the previous procedure fail, stop further processing and return a 400 SIP response code with no body.

The processing of each of type of request is specified in subsequent sections of the protocol. In the following section, a success response means the response with "success" code attribute, and a failure response is the response with "failure" code attribute.

3.2.5.2 Processing an addConference Request

The Focus Factory SHOULD use the following steps when processing an **addConference** request:

1. Authorize the user for creating a conference with the specified admission policy. How to authorize a user is implementation specific.
2. If the **conference-key** element is specified and if the value specified in the containing **issuing-server** element is not for the server currently processing this request, verify that the specified server is a valid server and forward the request to that server.
3. If the **conference-key** element is specified, decrypt the specified conference key using the private key of the Focus Factory certificate (1).

4. If a **conference-view** element is present, validate the **mcuType** value specified in each **entity-view** element against the list of available MCU-Types.
5. Authorize the **entity** attribute of each **user** element to verify that the organizer is allowed to invite these participants (2). The procedure for authorizing a user is implementation specific.
6. Verify that the size of the XML contained within **notification-data**, **organizer-roaming-data**, and **entity-settings** does not exceed the configured maximum storage size.
7. Verify that the organizer has not reached his quota of the maximum number of conferences the organizer is allowed to have at any point in time.
8. Validate that the specified Conference-Id is not already in use by an existing conference for the sending organizer.
9. Create an entry in the Conference Table with **version** "1" and all other details retrieved from the **addConference** request, and send a "success" **addConference** response.

If any of the steps in the previous procedure fail, stop processing and send a failure **addConference** response with an appropriate **reason** attribute. Values for the reason attribute are specified in section [2.2.3.1](#).

3.2.5.3 Processing a modifyConference Request

The Focus Factory SHOULD use the same steps as those specified in section [3.2.5.2](#) for processing an **addConference** request, with the following exceptions:

- Send a **modifyConference** response anywhere an **addConference** response is mentioned.
- In the previous procedure, change step 8 to validate that a conference with the specified Conference-Id exists for the sending organizer, to verify that the value of the **version** attribute from the **conference-info** element equals the value stored in the Conference Table for the specified conference, and to verify that the value of the **server-mode** element from the conference-description element equals the value stored in the Conference Table for the specified conference.
- In the previous procedure, change step 9 to replace the existing entry in the Conference Table with an entry with **version** one higher than the existing version and all other details taken from the **modifyConference** request, and send a "success" **modifyConference** response.

Values for the reason attribute in the response are specified in section [2.2.3.2](#).

3.2.5.4 Processing a deleteConference Request

The Focus Factory SHOULD use the following steps when processing a **deleteConference** request:

1. Validate that a conference with the specified Conference-Id exists for the sending organizer.
2. If the conference is currently active, communicate with the focus to deactivate the conference.
3. Remove the conference entry for the specified organizer and Conference-Id.
4. Send a success **deleteConference** response.

If any of the previous steps fail, stop processing and send a "failure" **deleteConference** response with an appropriate **reason** attribute. Values for the reason attribute are specified in section [2.2.3.3](#).

3.2.5.5 Processing a **getConference** Request

The Focus Factory SHOULD use the following steps when processing a **getConference** request:

1. Validate that a conference with the specified Conference-Id exists for the sending organizer.
2. Retrieve the details for the specified organizer and Conference-Id from the Conference Table to populate a **getConference** response. If an **encryption-key** is specified, and a **conference-key** exists for the conference, encrypt the **conference-key** using the public key from the specified **x509-certificate** and include it in the **getConference** response.
3. Send a "success" **getConference** response.

If any of the previous steps fail, stop processing and send a "failure" **getConference** response with an appropriate **reason** attribute. Values for the reason attribute are specified in section [2.2.3.4](#).

3.2.5.6 Processing a **getConferences** Request

The Focus Factory SHOULD retrieve the list of conferences and associated conference data for the specified organizer from the Conference Table to populate and send a "success" **getConferences** response.

If the previous procedure fails, stop processing and send a "failure" **getConferences** response with an appropriate **reason** attribute. Values for the reason attribute are specified in section [2.2.3.5](#).

3.2.5.7 Processing a **getEncryptionKey** Request

The Focus Factory SHOULD use the Focus Factory certificate (1) to construct and send a **getEncryptionKey** response.

If the previous procedure fails, stop processing and send a failure **getEncryptionKey** response with an appropriate **reason** attribute. Values for the reason attribute are specified in section [2.2.3.6](#).

3.2.5.8 Processing a **getAvailableMcuTypes** Request

The Focus Factory SHOULD retrieve the list of MCU-Types available for serving conferences to populate and send a **getAvailableMcuTypes** response.

If the previous procedure fails, stop processing and send a "failure" **getAvailableMcuTypes** response with an appropriate **reason** attribute. Values for the reason attribute are specified in section [2.2.3.7](#).

If the **server-mode** attribute of the request is 14, the returned list MUST NOT contain the value "meeting" [<40>](#). If the **server-mode** attribute of the request is 13, the returned list MUST NOT contain the value "data-conf".

3.2.5.9 Processing a **getConferencingCapabilities** Request

The Focus Factory SHOULD [<41>](#) retrieve the list of MCU types available for serving conferences and the anonymous scheduling capability, and use this information to populate and send a **getConferencingCapabilities** response.

If the previous procedure fails, stop processing and send a "failure" **getConferencingCapabilities** response with an appropriate **reason** attribute. Values for the reason attribute are specified in section [2.2.3.8](#).

If the **server-mode** attribute of the request is 14, the returned list MUST NOT contain the value "meeting"<42>. If the **server-mode** attribute of the request is 13, the returned list MUST NOT contain the value "data-conf".

3.2.6 Timer Events

3.2.6.1 Conference Expired

When the Conference Expiration Timer fires, the Focus Factory SHOULD find any conference in the Conference Table with an expiry-time that has passed and, if the conference is not active, delete the entry for that conference from the Conference Table.

3.2.7 Other Local Events

None.

4 Protocol Examples

4.1 Create a Conference

The following example is an **addConference** request from a protocol client application to the Focus Factory.

```
SERVICE sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory SIP/2.0
From: sip:alice@contoso.com;tag=f7588dc66124429ab736;epid=1
To: sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory;tag=ccb81c3509
Call-ID: 72d6a36a36784cf58e7e7ab1a51deca2
CSeq: 1 SERVICE
Content-Type: application/cccp+xml
...Other SIP headers...
<request xmlns="urn:ietf:params:xml:ns:cccp"
xmlns:mscp="http://schemas.microsoft.com/rtc/2005/08/cccpextensions" C3PVersion="1"
to="sip:mhakim2@clientsemain.rtmp.selfhost.corp.microsoft.com;gruu;opaque=app:conf:focusfacto
ry" from="sip:mhakim2@clientsemain.rtmp.selfhost.corp.microsoft.com" requestId="68537848">
  <addConference>
    <ci:conference-info xmlns:ci="urn:ietf:params:xml:ns:conference-info" entity=""
xmlns:msci="http://schemas.microsoft.com/rtc/2005/08/confinfoextensions">
      <ci:conference-description>
        <msci:conference-id>TPDD8VYG</msci:conference-id>
        <msci:admission-policy>openAuthenticated</msci:admission-policy>
        <cis:separator xmlns:cis="urn:ietf:params:xml:ns:conference-info-separator">
          </cis:separator>
        <msci:pstn-access>
          </msci:pstn-access>
        <cis:separator xmlns:cis="urn:ietf:params:xml:ns:conference-info-separator">
          </cis:separator>
        <msci:autopromote>2147483648</msci:autopromote>
        <msci:pstn-lobby-bypass>true</msci:pstn-lobby-bypass>
        <msci:server-mode>14</msci:server-mode>
      </ci:conference-description>
      <ci:conference-state>
        <locked xmlns="urn:ietf:params:xml:ns:conference-info">false</locked>
      </ci:conference-state>
      <msci:conference-view>
        <msci:entity-view entity="chat" />
        <msci:entity-view entity="audio-video" />
        <msci:entity-view entity="applicationsharing" />
        <msci:entity-view entity="data-conf" />
      </msci:conference-view>
    </ci:conference-info>
  </addConference>
</request>
```

The following example is a successful **addConference** response from a Focus Factory to a protocol client application.

```
SIP/2.0 200 OK
From: sip:alice@contoso.com;tag=f7588dc66124429ab736;epid=1
To: sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory;tag=ccb81c3509
Call-ID: 72d6a36a36784cf58e7e7ab1a51deca2
CSeq: 1 SERVICE
Content-Type: application/cccp+xml
...Other SIP headers...
```

```

<response C3PVersion="1"
  requestId="1"
  from="sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory"
  to="sip:alice@contoso.com"
  code="success"
  xmlns="urn:ietf:params:xml:ns:cccp">
  <addConference>
    <conference-info
      entity="sip:alice@contoso.com;gruu;opaque=app:conf:focus:id:SDFBSD12"
      state="partial"
      version="1"
      xmlns="urn:ietf:params:xml:ns:conference-info"/>
    </addConference>
  </response>

```

The following example is a "failure" **addConference** response from a Focus Factory to a protocol client application.

```

SIP/2.0 403 anonymousUsersNotAllowed
From: sip:alice@contoso.com;tag=f7588dc66124429ab736;epid=1
To: sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory;tag=ccb81c3509
Call-ID: 72d6a36a36784cf58e7e7ab1a51deca2
CSeq: 1 SERVICE
Content-Type: application/cccp+xml
...Other SIP headers...

<response C3PVersion="1"
  requestId="1"
  from="sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory"
  to="sip:alice@contoso.com"
  code="failure"
  xmlns="urn:ietf:params:xml:ns:cccp">
  <addConference reason="anonymousUsersNotAllowed"/>
</response>

```

The following example is an **addConference** request to create a static meeting.

```

<addConference>
  <ci:conference-info xmlns:ci="urn:ietf:params:xml:ns:conference-info" entity=""
  xmlns:msci="http://schemas.microsoft.com/rtc/2005/08/confinfoextensions"
  static="true">
    <ci:conference-description>
      <msci:conference-id>WJ7UVUYO</msci:conference-id>
    </ci:conference-description>
  </conference-info>
</addConference>

```

4.2 Modify a Conference

The following example is a **modifyConference** request from a client application to the Focus Factory. [<43>](#)

```

SERVICE sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory SIP/2.0
From: sip:alice@contoso.com;tag=f7588dc66124429ab736;epid=1

```

To: sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory;tag=ccb81c3509
Call-ID: 4de149885e5a4c598e64520a708c4739
CSeq: 2 SERVICE
Content-Type: application/cccp+xml
...Other SIP headers...

```
<request C3PVersion="1"
  requestId="2"
  from="sip:alice@contoso.com"
  to="sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory"
  xmlns="urn:ietf:params:xml:ns:cccp"
  xmlns:msci="http://schemas.microsoft.com/rtc/2005/08/confinfoextensions">
  <modifyConference>
    <conference-info entity=""
      version="1"
      xmlns="urn:ietf:params:xml:ns:conference-info">
      <conference-description>
        <subject>Conference subject 2</subject>
        <msci:conference-id>SDFBSD12</msci:conference-id>
        <msci:expiry-time>2006-01-10T17:11:47Z</msci:expiry-time>
        <msci:admission-policy>openAuthenticated</msci:admission-policy>
        <msci:organizer-roaming-data>
          <AnyOrganizerXml/>
        </msci:organizer-roaming-data>
        <msci:notification-data>
          <AnyParticipantXml/>
        </msci:notification-data>
      </conference-description>
      <users>
        <user entity="sip:bob@contoso.com">
          <roles>
            <entry>presenter</entry>
          </roles>
        </user>
      </users>
      <msci:conference-view>
        <msci:entity-view entity="chat"/>
        <msci:entity-view entity="meeting">
          <msci:entity-settings>
            <settings
              xmlns="http://schemas.microsoft.com/rtc/2005/08/dataconfinfoextensions">
              <app-viewing-behavior>enableWithFullSharing</app-viewing-behavior>
              <conferencing-type>collaboration</conferencing-type>
            </settings>
          </msci:entity-settings>
        </msci:entity-view>
      </msci:conference-view>
    </conference-info>
  </modifyConference>
</request>
```

The following example is a **modifyConference** request from a client application to the Focus Factory. [<44>](#)

```
SERVICE sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory SIP/2.0
From: sip:alice@contoso.com;tag=f7588dc66124429ab736;epid=1
To: sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory;tag=ccb81c3509
```

Call-ID: 4de149885e5a4c598e64520a708c4739
CSeq: 2 SERVICE
Content-Type: application/cccp+xml
...Other SIP headers...

```
<request C3PVersion="1"
  requestId="2"
  from="sip:alice@contoso.com"
  to="sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory"
  xmlns="urn:ietf:params:xml:ns:cccp"
  xmlns:msci="http://schemas.microsoft.com/rtc/2005/08/confinfoextensions">
  <modifyConference>
    <ci:conference-info xmlns:ci="urn:ietf:params:xml:ns:conference-info" entity=""
      xmlns:msci="http://schemas.microsoft.com/rtc/2005/08/confinfoextensions" version="2"
      static="false">
      <ci:conference-description>
        <ci:subject>test</ci:subject>
        <msci:conference-id>WJ7UVUYO</msci:conference-id>
        <msci:expiry-time>2010-03-01T15:00:00Z</msci:expiry-time>
        <msci:admission-policy>closedAuthenticated</msci:admission-policy>
        <msci:notification-data>
          <conference-origin>scheduled</conference-origin>
        </msci:notification-data>
        <cis:separator xmlns:cis="urn:ietf:params:xml:ns:conference-info-separator">
        </cis:separator>
        <msci:pstn-access>
        </msci:pstn-access>
        <cis:separator xmlns:cis="urn:ietf:params:xml:ns:conference-info-separator">
        </cis:separator>
        <msci:autopromote>0</msci:autopromote>
        <msci:pstn-lobby-bypass>false</msci:pstn-lobby-bypass>
      </ci:conference-description>
      <ci:conference-state>
        <locked xmlns="urn:ietf:params:xml:ns:conference-info">false</locked>
      </ci:conference-state>
      <ci:users>
        <ci:user entity="sip:mhakin@fabrikam.com">
          <ci:roles>
            <ci:entry>presenter</ci:entry>
          </ci:roles>
        </ci:user>
      </ci:users>
      <msci:conference-view>
        <msci:entity-view entity="chat" />
        <msci:entity-view entity="audio-video">
          <msci:entity-settings>
            <msci:mediaFiltersRules>
              <msci:mayModifyOwnFilters>
                <msci:role>attendee</msci:role>
                <msci:value>false</msci:value>
              </msci:mayModifyOwnFilters>
              <msci:initialFilters>
                <msci:role>attendee</msci:role>
                <msci:ingressFilter>block</msci:ingressFilter>
              </msci:initialFilters>
            </msci:mediaFiltersRules>
          </msci:entity-settings>
        </msci:entity-view>
        <msci:entity-view entity="applicationsharing" />
      </msci:conference-view>
    </ci:conference-info>
  </modifyConference>
</request>
```

```

    <msci:entity-view entity="data-conf">
      <msci:entity-settings>
        <msdata:settings
xmlns:msdata="http://schemas.microsoft.com/rtc/2005/08/dataconfinfoextensions">
          <msdata:app-viewing-behavior>enableWithFullSharing</msdata:app-viewing-
behavior>
          <msdata:conferencing-type>collaboration</msdata:conferencing-type>
        </msdata:settings>
      </msci:entity-settings>
    </msci:entity-view>
  </msci:conference-view>
</ci:conference-info>
</modifyConference>
</request>

```

The following example is a successful **modifyConference** response from a Focus Factory to a client application.

```

SIP/2.0 200 OK
From: sip:alice@contoso.com;tag=f7588dc66124429ab736;epid=1
To: sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory;tag=ccb81c3509
Call-ID: 4de149885e5a4c598e64520a708c4739
CSeq: 2 SERVICE
Content-Type: application/cccp+xml
...Other SIP headers...

<response C3PVersion="1"
requestId="2"
from="sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory"
to="sip:alice@contoso.com"
code="success"
xmlns="urn:ietf:params:xml:ns:cccp">
  <modifyConference>
    <conference-info
      entity="sip:alice@contoso.com;gruu;opaque=app:conf:focus:id:SDFBSD12"
      state="partial"
      version="2"
      xmlns="urn:ietf:params:xml:ns:conference-info"/>
    </modifyConference>
  </response>

```

The following example is a "failure" **modifyConference** response from a Focus Factory to a protocol client application.

```

SIP/2.0 400 invalidConferenceId
From: sip:alice@contoso.com;tag=f7588dc66124429ab736;epid=1
To: sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory;tag=ccb81c3509
Call-ID: 4de149885e5a4c598e64520a708c4739
CSeq: 2 SERVICE
Content-Type: application/cccp+xml
...Other SIP headers...

<response C3PVersion="1"
requestId="2"
from="sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory"
to="sip:alice@contoso.com"

```



```

        code="failure"
        xmlns="urn:ietf:params:xml:ns:cccp">
      <modifyConference reason="invalidConferenceId"/>
    </response>

```

4.3 Delete a Conference

The following example is a **deleteConference** request from a protocol client application to the Focus Factory.

```

SERVICE sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory SIP/2.0
From: sip:alice@contoso.com;tag=f7588dc66124429ab736;epid=1
To: sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory;tag=ccb81c3509
Call-ID: 8cb3cfd3d8384f59adeb1b3e5c237d84
CSeq: 3 SERVICE
Content-Type: application/cccp+xml
...Other SIP headers...

<request C3PVersion="1"
  requestId="3"
  from="sip:alice@contoso.com"
  to="sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory"
  xmlns="urn:ietf:params:xml:ns:cccp"
  xmlns:misci="http://schemas.microsoft.com/rtc/2005/08/confinfoextensions">
  <deleteConference>
    <conferenceKeys misci:conference-id="BGSDFJ12"/>
  </deleteConference>
</request>

```

The following example is a successful **deleteConference** response from a Focus Factory to a protocol client application.

```

SIP/2.0 200 OK
From: sip:alice@contoso.com;tag=f7588dc66124429ab736;epid=1
To: sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory;tag=ccb81c3509
Call-ID: 8cb3cfd3d8384f59adeb1b3e5c237d84
CSeq: 3 SERVICE
Content-Type: application/cccp+xml
...Other SIP headers...

<response C3PVersion="1"
  requestId="3"
  from="sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory"
  to="sip:alice@contoso.com"
  code="success"
  xmlns="urn:ietf:params:xml:ns:cccp">
  <deleteConference/>
</response>

```

The following example is a failure **deleteConference** response from a Focus Factory to a protocol client application.

```

SIP/2.0 404 conferenceDoesNotExist
From: sip:alice@contoso.com;tag=f7588dc66124429ab736;epid=1

```

```
To: sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory;tag=ccb81c3509
Call-ID: 8cb3cfd3d8384f59adeb1b3e5c237d84
CSeq: 3 SERVICE
Content-Type: application/cccp+xml
...Other SIP headers...
```

```
<response C3PVersion="1"
  requestId="3"
  from="sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory"
  to="sip:alice@contoso.com"
  code="failure"
  xmlns="urn:ietf:params:xml:ns:cccp">
  <deleteConference reason="conferenceDoesNotExist"/>
</response>
```

4.4 Query for a List of Conferences

The following example is a **getConferences** request from a protocol client application to the Focus Factory.

```
SERVICE sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory SIP/2.0
From: sip:alice@contoso.com;tag=f7588dc66124429ab736;epid=1
To: sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory;tag=ccb81c3509
Call-ID: 11615b2791ae491a9c121d1d554a3e67
CSeq: 4 SERVICE
Content-Type: application/cccp+xml
...Other SIP headers...
```

```
<request C3PVersion="1"
  requestId="4"
  from="sip:alice@contoso.com"
  to="sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory"
  xmlns="urn:ietf:params:xml:ns:cccp">
  <getConferences/>
</request>
```

The following example is a **getConferences** response from a Focus Factory to a protocol client application.

```
SIP/2.0 200 OK
From: sip:alice@contoso.com;tag=f7588dc66124429ab736;epid=1
To: sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory;tag=ccb81c3509
Call-ID: 11615b2791ae491a9c121d1d554a3e67
CSeq: 4 SERVICE
Content-Type: application/cccp+xml
...Other SIP headers...

<response C3PVersion="1"
  requestId="4"
  from="sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory"
  to="sip:alice@contoso.com"
  code="success"
  xmlns="urn:ietf:params:xml:ns:cccp"
  xmlns:misci="http://schemas.microsoft.com/rtc/2005/08/confinfoextensions">
  <getConferences>
  <conferences>
```

```

<conference-info
  entity="sip:alice@contoso.com;gruu;opaque=app:conf:focus:id:SDFBSD12"
  state="partial"
  xmlns="urn:ietf:params:xml:ns:conference-info">
  <conference-description>
    <subject>Conference subject 2</subject>
    <msci:conference-id>SDFBSD12</msci:conference-id>
    <msci:admission-policy>openAuthenticated</msci:admission-policy>
  </conference-description>
</conference-info>
<conference-info
  entity="sip:alice@contoso.com;gruu; opaque=app:conf:focus:id:JASDFD2A"
  state="partial"
  xmlns="urn:ietf:params:xml:ns:conference-info">
  <conference-description>
    <subject>Conference subject</subject>
    <msci:conference-id>JASDFD2A</msci:conference-id>
    <msci:admission-policy>anonymous</msci:admission-policy>
  </conference-description>
</conference-info>
</conferences>
</getConferences>
</response>

```

The following example is a **getConferences** request to get the static meeting. [<45>](#)

```

<request xmlns="urn:ietf:params:xml:ns:cccp"
  xmlns:mscp="http://schemas.microsoft.com/rtc/2005/08/cccpextensions" C3PVersion="1" to="sip:
  alice@contoso.com;gruu;opaque=app:conf:focus:factory" from=" alice@contoso.com"
  requestId="161651504">
  <getConferences static="true" />
</request>

```

The response to the preceding call is as follows.

```

<getConferences static="true">
  <conferences>
    <conference-info xmlns="urn:ietf:params:xml:ns:conference-info" entity="sip:
    alice@contoso.com;gruu;opaque=app:conf:focus:id:I5TQBR7J" state="partial" version="471">
      <conference-description>
        <conference-id
        xmlns="http://schemas.microsoft.com/rtc/2005/08/confinfoextensions">I5TQBR7J</conference-id>
        <admission-policy
        xmlns="http://schemas.microsoft.com/rtc/2005/08/confinfoextensions">openAuthenticated</admission-policy>
      </conference-description>
      <conference-state>
        <locked>>false</locked>
      </conference-state>
    </conference-info>
  </conferences>
</getConferences>

```

4.5 Query for the Details of a Conference

The following example is a **getConference** request from a protocol client application to the Focus Factory.

```
SERVICE sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory SIP/2.0
From: sip:alice@contoso.com;tag=f7588dc66124429ab736;epid=1
To: sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory;tag=ccb81c3509
Call-ID: 5a27d6b1a82f4668be39db975d6fe184
CSeq: 5 SERVICE
Content-Type: application/cccp+xml
...Other SIP headers...

<request C3PVersion="1"
  requestId="5"
  from="sip:alice@contoso.com"
  to="sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory"
  xmlns="urn:ietf:params:xml:ns:cccp"
  xmlns:misci="http://schemas.microsoft.com/rtc/2005/08/confinfoextensions">
  <getConference>
    <conferenceKeys misci:conference-id="SDFBSD12"/>
  <misci:encryption-key>
    <misci:x509-certificate>123213789BC234D</misci:x509-certificate>
  </misci:encryption-key>
  </getConference>
</request>
```

The following example is a **getConference** response from a Focus Factory to a protocol client application.

```
SIP/2.0 200 OK
From: sip:alice@contoso.com;tag=f7588dc66124429ab736;epid=1
To: sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory;tag=ccb81c3509
Call-ID: 5a27d6b1a82f4668be39db975d6fe184
CSeq: 5 SERVICE
Content-Type: application/cccp+xml
...Other SIP headers...

<response C3PVersion="1"
  requestId="5"
  from="sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory"
  to="sip:alice@contoso.com"
  code="success"
  xmlns="urn:ietf:params:xml:ns:cccp"
  xmlns:misci="http://schemas.microsoft.com/rtc/2005/08/confinfoextensions">
  <getConference>
    <conference-info entity=""
      version="2"
      xmlns="urn:ietf:params:xml:ns:conference-info">
    <conference-description>
      <subject>Conference subject 2</subject>
      <misci:conference-id>SDFBSD12</misci:conference-id>
      <misci:expiry-time>2006-01-10T17:11:47Z</misci:expiry-time>
      <misci:admission-policy>openAuthenticated</misci:admission-policy>
      <misci:organizer-roaming-data>
        <AnyOrganizerXml/>
      </misci:organizer-roaming-data>
  </getConference>
</response>
```

```

    <msci:notification-data>
      <AnyParticipantXml/>
    </msci:notification-data>
  </conference-description>
  <users>
    <user entity="sip:bob@contoso.com">
      <roles>
        <entry>presenter</entry>
      </roles>
    </user>
  </users>
  <msci:conference-view>
    <msci:entity-view entity="chat"/>
    <msci:entity-view entity="meeting">
      <msci:entity-settings>
        <settings
          xmlns="http://schemas.microsoft.com/rtc/2005/08/dataconfinfoextensions">
          <app-viewing-behavior>enableWithFullSharing</app-viewing-behavior>
          <conferencing-type>collaboration</conferencing-type>
        </settings>
      </msci:entity-settings>
    </msci:entity-view>
  </msci:conference-view>
</conference-info>
</getConference>
</response>

```

4.6 Obtaining an Encryption Key

The following example is a **getEncryptionKey** request from a protocol client application to the Focus Factory.

```

SERVICE sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory SIP/2.0
From: sip:alice@contoso.com;tag=f7588dc66124429ab736;epid=1
To: sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory;tag=ccb81c3509
Call-ID: e71edff2fd924af389cd334ba827049b
CSeq: 6 SERVICE
Content-Type: application/cccp+xml
...Other SIP headers...

<request C3PVersion="1"
  requestId="6"
  from="sip:alice@contoso.com"
  to="sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory"
  xmlns="urn:ietf:params:xml:ns:cccp">
  <getEncryptionKey/>
</request>

```

The following example is a **getEncryptionKey** response from a Focus Factory to a protocol client application.

```

SIP/2.0 200 OK
From: sip:alice@contoso.com;tag=f7588dc66124429ab736;epid=1
To: sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory;tag=ccb81c3509
Call-ID: e71edff2fd924af389cd334ba827049b
CSeq: 6 SERVICE

```

```

Content-Type: application/cccp+xml
...Other SIP headers...

<response C3PVersion="1"
    requestId="6"
    from="sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory"
    to="sip:alice@contoso.com"
    code="success"
    xmlns="urn:ietf:params:xml:ns:cccp"
    xmlns:misci="http://schemas.microsoft.com/rtc/2005/08/confinfoextensions">
    <getEncryptionKey>
<misci:encryption-key>
    <misci:x509-certificate>123213789BC234D</misci:x509-certificate>
</misci:encryption-key>
<misci:opaque>
    <misci:issuing-server>server.contoso.com</misci:issuing-server>
</misci:opaque>
    </getEncryptionKey>
</response>

```

4.7 Obtaining a List of MCU-Types

The following example is a **getAvailableMcuTypes** request from a protocol client application to the Focus Factory.

```

SERVICE sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory SIP/2.0
From: sip:alice@contoso.com;tag=f7588dc66124429ab736;epid=1
To: sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory;tag=ccb81c3509
Call-ID: 65a46bb0b6604f4b8bc3f47b3f225c46
CSeq: 7 SERVICE
Content-Type: application/cccp+xml
...Other SIP headers...

<request C3PVersion="1"
    requestId="7"
    from="sip:alice@contoso.com"
    to="sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory"
    xmlns="urn:ietf:params:xml:ns:cccp">
    <getAvailableMcuTypes/>
</request>

```

The following example is a **getAvailableMcuTypes** response from a Focus Factory to a protocol client application.

```

SIP/2.0 200 OK
From: sip:alice@contoso.com;tag=f7588dc66124429ab736;epid=1
To: sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory;tag=ccb81c3509
Call-ID: 65a46bb0b6604f4b8bc3f47b3f225c46
CSeq: 7 SERVICE
Content-Type: application/cccp+xml
...Other SIP headers...

<response C3PVersion="1"
    requestId="7"
    from="sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory"
    to="sip:alice@contoso.com"

```

```

        code="success"
        xmlns="urn:ietf:params:xml:ns:cccp">
<getAvailableMcuTypes>
  <mcu-types>
<mcu-type>audio-video</mcu-type>
<mcu-type>chat</mcu-type>
<mcu-type>meeting</mcu-type>
<mcu-type>phone-conf</mcu-type>
</mcu-types>
  </getAvailableMcuTypes>
</response>

```

4.8 Obtaining Focus Factory Capabilities

This section follows the product behavior described in product behavior note [46](#).

The following example is a **getConferencingCapabilities** request from a protocol client application to the Focus Factory.

```

SERVICE sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory SIP/2.0
From: sip:alice@contoso.com;tag=f7588dc66124429ab736;epid=1
To: sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory;tag=ccb81c3509
Call-ID: 65a46bb0b6604f4b8bc3f47b3f225c46
CSeq: 7 SERVICE
Content-Type: application/cccp+xml
...Other SIP headers...

<request requestId="1" from="sip:alice@contoso.com"
to="sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory"
xmlns="urn:ietf:params:xml:ns:cccp">
  <getConferencingCapabilities/>
</request>

```

The following example is a **getConferencingCapabilities** response from a Focus Factory to a protocol client application.

```

SIP/2.0 200 OK
From: sip:alice@contoso.com;tag=f7588dc66124429ab736;epid=1
To: sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory;tag=ccb81c3509
Call-ID: 65a46bb0b6604f4b8bc3f47b3f225c46
CSeq: 7 SERVICE
Content-Type: application/cccp+xml
...Other SIP headers...

<response C3PVersion="1" requestId="1"
from=sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory
to="sip:alice@contoso.com" code="success" xmlns="urn:ietf:params:xml:ns:cccp">
  <getConferencingCapabilities capability-version="0">
    <mcu-types>
      <mcuType>audiovideo</mcuType>
      <mcuType>meeting</mcuType>
      <mcuType>chat</mcuType>
      <mcuType>phone-conf</mcuType>
      <mcuType>applicationsharing</mcuType>
    </mcu-types>
    <anonymous-scheduling>true</anonymous-scheduling>
  </getConferencingCapabilities>
</response>

```

```
</getConferencingCapabilities></response>
```

The following example is a **getConferencingCapabilities** request from a protocol client application to the Focus Factory. [<47>](#)

```
SERVICE sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory SIP/2.0
From: sip:alice@contoso.com;tag=f7588dc66124429ab736;epid=1
To: sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory;tag=ccb81c3509
Call-ID: 65a46bb0b6604f4b8bc3f47b3f225c46
CSeq: 7 SERVICE
Content-Type: application/cccp+xml
...Other SIP headers...

<request requestId="1" from="sip:alice@contoso.com"
to="sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory"
xmlns="urn:ietf:params:xml:ns:cccp">
  <getConferencingCapabilities server-mode="14"/>
</request>
```

The following example is a **getConferencingCapabilities** response from a Focus Factory to a protocol client application. [<48>](#)

```
SIP/2.0 200 OK
From: sip:alice@contoso.com;tag=f7588dc66124429ab736;epid=1
To: sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory;tag=ccb81c3509
Call-ID: 65a46bb0b6604f4b8bc3f47b3f225c46
CSeq: 7 SERVICE
Content-Type: application/cccp+xml
...Other SIP headers...

<response C3PVersion="1" requestId="1"
from=sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory to="sip:alice@contoso.com"
code="success" xmlns="urn:ietf:params:xml:ns:cccp"
xmlns:mhci="http://schemas.microsoft.com/rtc/2005/08/confinfoextensions"
xmlns:mhci2="http://schemas.microsoft.com/rtc/2008/12/confinfoextensions"
xmlns:mhcp="http://schemas.microsoft.com/rtc/2005/08/cccpextensions">
  <getConferencingCapabilities capability-version="0">
    <mcu-types>
      <mcuType>applicationsharing</mcuType>
      <mcuType>audio-video</mcuType>
      <mcuType>data-conf</mcuType>
      <mcuType>chat</mcuType>
      <mcuType>phone-conf</mcuType>
    </mcu-types>
    <pstn-bridging>
      <enabled>true</enabled>
      <access-numbers>
        <mhci:internal-
url>https://tk5ucdfp101.exchange.corp.microsoft.com:443/Dialin</mhci:internal-url>
        <mhci:external-url>https://lslm84.meet.microsoft.com:443/Dialin</mhci:external-url>
        <mhci:region name="Redmond">
          <mhci:access-number>
            <mhci:language tag="en-US" lcid="1033" />
            <mhci:language tag="de-DE" lcid="1031" />
            <mhci:language tag="fr-FR" lcid="1036" />
            <mhci:number>425-707-0001</mhci:number>
          </mhci:access-number>

```



```

    <msci:access-number>
      <msci:language tag="en-US" lcid="1033" />
      <msci:number>425-707-0002</msci:number>
    </msci:access-number>
  </msci:access-number>
  <msci:access-number>
    <msci:language tag="fr-FR" lcid="1036" />
    <msci:language tag="en-US" lcid="1033" />
    <msci:language tag="de-DE" lcid="1031" />
    <msci:number>425-707-9999</msci:number>
  </msci:access-number>
</msci:region>
  <msci2:default-region>Redmond</msci2:default-region>
</access-numbers>
</pstn-bridging>
<conference-key-optional>true</conference-key-optional>
<anonymous-scheduling>true</anonymous-scheduling>
<default-admission-policy>anonymous</default-admission-policy>
<cis:separator />
<mscp:schedule-locked>true</mscp:schedule-locked>
<msci:autopromote-allowed>2147516416</msci:autopromote-allowed>
<mscp:default-autopromote>32768</mscp:default-autopromote>
<msci:pstn-lobby-bypass-allowed>true</msci:pstn-lobby-bypass-allowed>
<mscp:static-meeting-limit>1</mscp:static-meeting-limit>
<mscp:default-meeting-static>true</mscp:default-meeting-static>
<msci:recording-allowed>true</msci:recording-allowed>
<msci:externaluser-recording-allowed>false</msci:externaluser-recording-allowed>
<msci:default-entry-exit-announcements>false</msci:default-entry-exit-announcements>
</getConferencingCapabilities>

```

4.9 Delegation Request

The following example is an **addConference** request from a protocol client application to the Focus Factory using delegation. In this example, bob@contoso.com is sending an **addConference** request on behalf of alice@contoso.com. [<49>](#)

```

SERVICE sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory SIP/2.0
From: sip:bob@contoso.com;tag=f7588dc66124429ab736;epid=1
To: sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory;tag=ccb81c3509
p-session-on-behalf-of: sip:alice@contoso.com
Call-ID: 72d6a36a36784cf58e7e7ab1a51deca2
CSeq: 1 SERVICE
Content-Type: application/cccp+xml
...Other SIP headers...

<request C3PVersion="1"
  requestId="1"
  from="sip:alice@contoso.com"
  to="sip:alice@contoso.com;gruu;opaque=app:conf:focusfactory"
  xmlns="urn:ietf:params:xml:ns:cccp"
  xmlns:msci="http://schemas.microsoft.com/rtc/2005/08/confinfoextensions">
  <addConference>
    <conference-info entity=""
      xmlns="urn:ietf:params:xml:ns:conference-info">
      <conference-description>
        <subject>Conference subject</subject>
        <msci:conference-id>SDFBSD12</msci:conference-id>
        <msci:conference-key>

```

```

    <msci:cms-data>ABDEF1243</msci:cms-data>
    <msci:opaque>
      <msci:issuing-server>server.contoso.com</msci:issuing-server>
    </msci:opaque>
  </msci:conference-key>
  <msci:expiry-time>2006-01-10T17:11:47Z</msci:expiry-time>
  <msci:admission-policy>anonymous</msci:admission-policy>
  <msci:organizer-roaming-data>
    <AnyOrganizerXml/>
  </msci:organizer-roaming-data>
  <msci:notification-data>
    <AnyParticipantXml/>
  </msci:notification-data>
</conference-description>
<users>
  <user entity="sip:bob@contoso.com">
    <roles>
      <entry>presenter</entry>
    </roles>
  </user>
  <user entity="sip:jack@fabrikam.com">
    <roles>
      <entry>attendee</entry>
    </roles>
  </user>
</users>
<msci:conference-view>
  <msci:entity-view entity="phone-conf">
    <msci:entity-settings>
      <settings xmlns="
http://schemas.microsoft.com/rtc/2005/08/acpconfinfoextensions">
        <tollNumber>+1-4255550101</tollNumber>
        <tollFreeNumber>+1-8661111111</tollFreeNumber>
        <domain>conferencing.company.com</domain>
        <leaderPasscode>654321</leaderPasscode>
        <participantPasscode>70344101</participantPasscode>
        <showTollNumber>true</showTollNumber>
        <showTollFreeNumber>true</showTollFreeNumber>
        <enableCallMe>true</enableCallMe>
        <robotCallerIds>
          <entry>+1-4255550100</entry>
          <entry>+1-4255550101</entry>
          <entry>+1-4255550102</entry>
        </robotCallerIds>
      </settings>
    </msci:entity-settings>
  </msci:entity-view>
</msci:conference-view>
</conference-info>
</addConference>
</request>

```

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

Preliminary

6 Appendix A: 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
- Microsoft® Office Communications Server 2007 R2
- Microsoft® Office Communicator 2007
- Microsoft® Office Communicator 2007 R2
- Microsoft® Lync™ Server 2010
- Microsoft® Lync™ 2010
- Microsoft® Lync Server 15 Technical Preview
- Microsoft® Lync 15 Technical Preview

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 2.2:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported. Required for all other products.

[<2> Section 2.2.1.2:](#) Office Communications Server 2007, Office Communicator 2007: This behavior is not supported.

[<3> Section 2.2.1.2:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

[<4> Section 2.2.1.2:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

[<5> Section 2.2.1.2:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

[<6> Section 2.2.1.3:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

[<7> Section 2.2.2.1:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

[<8> Section 2.2.2.1:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

[<9> Section 2.2.2.1:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<10> [Section 2.2.2.5:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<11> [Section 2.2.2.5:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<12> [Section 2.2.2.7:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<13> [Section 2.2.2.8:](#) Office Communications Server 2007, Office Communicator 2007: This behavior is not supported.

<14> [Section 2.2.2.8:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<15> [Section 2.2.3.1:](#) When starting Lync Server 2010, success is returned in this case.

<16> [Section 2.2.3.7:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<17> [Section 2.2.3.8:](#) Office Communications Server 2007, Office Communicator 2007: This behavior is not supported.

<18> [Section 2.2.3.8:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported

<19> [Section 2.2.3.8:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported

<20> [Section 2.2.3.8:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported

<21> [Section 2.2.3.8:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported

<22> [Section 2.2.3.8:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported

<23> [Section 2.2.3.8:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported

<24> [Section 2.2.3.8:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported

<25> [Section 2.2.3.8:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported

<26> [Section 2.2.3.8:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported

<27> [Section 2.2.3.8:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported

<28> [Section 2.2.3.8:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2, Lync Server 2010, Lync 2010: This behavior is not supported

<29> [Section 2.2.3.8:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2, Lync Server 2010, Lync 2010: This behavior is not supported

<30> [Section 2.2.3.8:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2, Lync Server 2010, Lync 2010: This behavior is not supported

<31> [Section 2.2.3.8:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2, Lync Server 2010, Lync 2010: This behavior is not supported

<32> [Section 2.2.3.8:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2, Lync Server 2010, Lync 2010: This behavior is not supported

<33> [Section 3.1.4.2:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<34> [Section 3.1.4.2:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<35> [Section 3.1.4.7:](#) Office Communications Server 2007, Office Communicator 2007: This behavior is not supported.

<36> [Section 3.1.4.7:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<37> [Section 3.1.4.8:](#) Office Communications Server 2007, Office Communicator 2007: This behavior is not supported.

<38> [Section 3.1.4.8:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<39> [Section 3.1.5.10:](#) Office Communications Server 2007, Office Communicator 2007: This behavior is not supported.

<40> [Section 3.2.5.8:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<41> [Section 3.2.5.9:](#) Office Communications Server 2007, Office Communicator 2007: This behavior is not supported.

<42> [Section 3.2.5.9:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<43> [Section 4.2:](#) Supported in Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2 only.

<44> [Section 4.2:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<45> [Section 4.4:](#) Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<46> [Section 4.8:](#) Office Communications Server 2007, Office Communicator 2007: This behavior is not supported.

<47> [Section 4.8](#): Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<48> [Section 4.8](#): Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

<49> [Section 4.9](#): Office Communications Server 2007, Office Communicator 2007, Office Communications Server 2007 R2, Office Communicator 2007 R2: This behavior is not supported.

Preliminary

7 Change Tracking

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

Preliminary

8 Index

A

Abstract data model
[client](#) 34
[focus factory](#) 39
addConference
[example](#) 44
[request](#) 20
[processing](#) 40
response
[processing](#) 37
[Applicability](#) 9

C

[Capability negotiation](#) 9
[Change tracking](#) 64
Client
[abstract data model](#) 34
higher-layer triggered events
[create a conference](#) 34
delete a conference ([section 3.1.4.3](#) 35, [section 3.1.4.6](#) 36)
[modify a conference](#) 35
[obtain a list of MCU types](#) 36
[obtain focus factory capabilities](#) 36
[query for a list of conferences](#) 35
[query for the details of a conference](#) 35
[initialization](#) 34
message processing
[processing a deleteConference response](#) 37
[processing a getAvailableMcuTypes response](#) 38
[processing a getConference response](#) 38
[processing a getConferences response](#) 38
[processing a getConferencingCapabilities response](#) 38
[processing a getEncryptionKey response](#) 38
[processing a modifyConference response](#) 37
[processing a response](#) 37
[processing a SIP response](#) 36
[processing an addConference response](#) 37
other local events
[create a self-signed certificate](#) 39
[obtain a focus factory certificate](#) 39
[send a request](#) 39
[overview](#) 34
sequencing rules
[processing a deleteConference response](#) 37
[processing a getAvailableMcuTypes response](#) 38
[processing a getConference response](#) 38
[processing a getConferences response](#) 38
[processing a getConferencingCapabilities response](#) 38
[processing a getEncryptionKey response](#) 38
[processing a modifyConference response](#) 37
[processing a response](#) 37
[processing a SIP response](#) 36

[processing an addConference response](#) 37
[timer events](#) 39
[timers](#) 34
[Conference Data Model message](#) 12
[conference-description element](#) 14
[conference-info element](#) 13
[conference-view element](#) 17
[users element](#) 16
[Conference-description element](#) 14
[Conference-info element](#) 13
[Conference-view element](#) 17
Create a conference
[example](#) 44

D

Data model - abstract
[client](#) 34
[focus factory](#) 39
Delegation request
[example](#) 57
Delete a conference
[example](#) 49
deleteConference
[example](#) 49
[request](#) 21
[processing](#) 41
[response](#) 26
[processing](#) 37

E

Elements
[conference-description](#) 14
[conference-info](#) 13
[conference-view](#) 17
[users](#) 16
Examples
[create a conference](#) 44
[delegation request](#) 57
[delete a conference](#) 49
[modify a conference](#) 45
[obtain a list of MCU-types](#) 54
[obtain an encryption key](#) 53
[obtain focus factory capabilities](#) 55
[query for a list of conferences](#) 50
[query for the details of a conference](#) 52

F

[Fields - vendor-extensible](#) 9
Focus factory
[abstract data model](#) 39
[higher-layer triggered events](#) 40
[initialization](#) 40
[message processing](#) 40
[addConference request](#) 40
[deleteConference request](#) 41
[getAvailableMcuTypes request](#) 42

- [getConference request](#) 42
- [getConferences request](#) 42
- [getConferencingCapabilities request](#) 42
- [getEncryptionKey request](#) 42
- [modifyConference request](#) 41
- [SIP request](#) 40
- other local events 43
- sequencing rules 40
 - [addConference request](#) 40
 - [deleteConference request](#) 41
 - [getAvailableMcuTypes request](#) 42
 - [getConference request](#) 42
 - [getConferences request](#) 42
 - [getConferencingCapabilities request](#) 42
 - [getEncryptionKey request](#) 42
 - [modifyConference request](#) 41
 - [SIP request](#) 40
- timer events
 - [conference expired](#) 43
- [timers](#) 40

G

- getAvailableMcuTypes
 - [example](#) 54
 - [request](#) 22
 - [processing](#) 42
 - [response](#) 28
 - [processing](#) 38
- getConference
 - [example](#) 52
 - [request](#) 21
 - [processing](#) 42
 - [response](#) 26
 - [processing](#) 38
- getConferences
 - [example](#) 50
 - [request](#) 22
 - [processing](#) 42
 - [response](#) 27
 - [processing](#) 38
- getConferencingCapabilities
 - [example](#) 55
 - [request](#) 23
 - [processing](#) 42
 - [response](#) 29
 - [processing](#) 38
- getEncryptionKey
 - [example](#) 53
 - [request](#) 22
 - [processing](#) 42
 - [response](#) 28
 - [processing](#) 38
- [Glossary](#) 7

H

- Higher-layer triggered events
 - client
 - [create a conference](#) 34
 - delete a conference ([section 3.1.4.3](#) 35, [section 3.1.4.6](#) 36)

- [modify a conference](#) 35
- [obtain a list of MCU types](#) 36
- [obtain focus factory capabilities](#) 36
- [query for a list of conferences](#) 35
- [query for the details of a conference](#) 35
- [focus factory](#) 40

I

- [Implementer - security considerations](#) 59
- [Index of security parameters](#) 59
- [Informative references](#) 8
- Initialization
 - [client](#) 34
 - [focus factory](#) 40
- [Introduction](#) 7

M

- Message processing
 - client
 - [processing a deleteConference response](#) 37
 - [processing a getAvailableMcuTypes response](#) 38
 - [processing a getConference response](#) 38
 - [processing a getConferences response](#) 38
 - [processing a getConferencingCapabilities response](#) 38
 - [processing a getEncryptionKey response](#) 38
 - [processing a modifyConference response](#) 37
 - [processing a response](#) 37
 - [processing a SIP response](#) 36
 - [processing an addConference response](#) 37
 - [focus factory](#) 40
 - [addConference request](#) 40
 - [deleteConference request](#) 41
 - [getAvailableMcuTypes request](#) 42
 - [getConference request](#) 42
 - [getConferences request](#) 42
 - [getConferencingCapabilities request](#) 42
 - [getEncryptionKey request](#) 42
 - [modifyConference request](#) 41
 - [SIP request](#) 40
- [Messages](#) 11
 - [Conference Data Model](#) 12
 - [conference-description element](#) 14
 - [conference-info element](#) 13
 - [conference-view element](#) 17
 - [users element](#) 16
- [Requests](#) 19
 - [addConference](#) 20
 - [deleteConference](#) 21
 - [getAvailableMcuTypes](#) 22
 - [getConference](#) 21
 - [getConferences](#) 22
 - [getConferencingCapabilities](#) 23
 - [getEncryptionKey](#) 22
 - [modifyConference](#) 21
- [Responses](#) 23
 - [deleteConference](#) 26
 - [getAvailableMcuTypes](#) 28
 - [getConference](#) 26

[getConferences](#) 27
[getConferencingCapabilities](#) 29
[getEncryptionKey](#) 28
[modifyConference](#) 25
[transport](#) 11

Modify a conference
[example](#) 45

modifyConference
[example](#) 45
[request](#) 21
[processing](#) 41
[response](#) 25
[processing](#) 37

N

[Normative references](#) 8

O

Obtain a list of MCU-types
[example](#) 54

Obtain an encryption key
[example](#) 53

Obtain focus factory capabilities
[example](#) 55

Other local events
client
[create a self-signed certificate](#) 39
[obtain a focus factory certificate](#) 39
[send a request](#) 39
[focus factory](#) 43
[Overview \(synopsis\)](#) 8

P

[Parameters - security index](#) 59

[Preconditions](#) 9

[Prerequisites](#) 9

[Product behavior](#) 60

Q

Query for a list of conferences
[example](#) 50

Query for the details of a conference
[example](#) 52

R

[References](#) 8

[informative](#) 8

[normative](#) 8

[Relationship to other protocols](#) 9

[Requests message](#) 19

[addConference](#) 20

[deleteConference](#) 21

[getAvailableMcuTypes](#) 22

[getConference](#) 21

[getConferences](#) 22

[getConferencingCapabilities](#) 23

[getEncryptionKey](#) 22

[modifyConference](#) 21

[Responses message](#) 23

[deleteConference](#) 26

[getAvailableMcuTypes](#) 28

[getConference](#) 26

[getConferences](#) 27

[getConferencingCapabilities](#) 29

[getEncryptionKey](#) 28

[modifyConference](#) 25

S

Security

[implementer considerations](#) 59

[parameter index](#) 59

Sequencing rules

client

[processing a deleteConference response](#) 37

[processing a getAvailableMcuTypes response](#) 38

[processing a getConference response](#) 38

[processing a getConferences response](#) 38

[processing a getConferencingCapabilities response](#) 38

[processing a getEncryptionKey response](#) 38

[processing a modifyConference response](#) 37

[processing a response](#) 37

[processing a SIP response](#) 36

[processing an addConference response](#) 37

focus factory 40

[addConference request](#) 40

[deleteConference request](#) 41

[getAvailableMcuTypes request](#) 42

[getConference request](#) 42

[getConferences request](#) 42

[getConferencingCapabilities request](#) 42

[getEncryptionKey request](#) 42

[modifyConference request](#) 41

[SIP request](#) 40

[Standards assignments](#) 10

T

Timer events

[client](#) 39

focus factory

[conference expired](#) 43

Timers

[client](#) 34

[focus factory](#) 40

[Tracking changes](#) 64

[Transport](#) 11

Triggered events

client

[create a conference](#) 34

delete a conference ([section 3.1.4.3](#) 35, [section 3.1.4.6](#) 36)

[modify a conference](#) 35

[obtain a list of MCU types](#) 36

[obtain focus factory capabilities](#) 36

[query for a list of conferences](#) 35

[query for the details of a conference](#) 35

Triggered events - higher-layer
[focus factory](#) 40

U

[Users element](#) 16

V

[Vendor-extensible fields](#) 9

[Versioning](#) 9

Preliminary