

[MS-OXWSCONV]:

Conversations Web Service Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

Preliminary Documentation. This Open Specification provides documentation for past and current releases and/or for the pre-release 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 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.

Preliminary

Revision Summary

Date	Revision History	Revision Class	Comments
5/5/2010	0.1.0	Major	Initial Release.
8/4/2010	0.2	Minor	Clarified the meaning of the technical content.
11/3/2010	0.2	No change	No changes to the meaning, language, or formatting of the technical content.
3/18/2011	1.0	Major	Significantly changed the technical content.
8/5/2011	2.0	Major	Significantly changed the technical content.
10/7/2011	2.0	No Change	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	3.0	Major	Significantly changed the technical content.
4/27/2012	3.0	No Change	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	4.0	Major	Significantly changed the technical content.
10/8/2012	5.0	Major	Significantly changed the technical content.
2/11/2013	5.0	No Change	No changes to the meaning, language, or formatting of the technical content.
7/26/2013	5.1	Minor	Clarified the meaning of the technical content.
11/18/2013	5.2	Minor	Clarified the meaning of the technical content.
2/10/2014	5.2	No Change	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	5.3	Minor	Clarified the meaning of the technical content.
7/31/2014	6.0	Major	Significantly changed the technical content.
10/30/2014	6.1	Minor	Clarified the meaning of the technical content.
5/26/2015	7.0	Major	Significantly changed the technical content.

Table of Contents

1	Introduction	6
1.1	Glossary	6
1.2	References	7
1.2.1	Normative References	7
1.2.2	Informative References	8
1.3	Overview	8
1.4	Relationship to Other Protocols	8
1.5	Prerequisites/Preconditions	9
1.6	Applicability Statement	9
1.7	Versioning and Capability Negotiation	9
1.8	Vendor-Extensible Fields	9
1.9	Standards Assignments.....	9
2	Messages.....	10
2.1	Transport.....	10
2.2	Common Message Syntax	10
2.2.1	Namespaces	10
2.2.2	Messages.....	10
2.2.3	Elements	11
2.2.4	Complex Types.....	11
2.2.5	Simple Types	11
2.2.6	Attributes	11
2.2.7	Groups	11
2.2.8	Attribute Groups.....	11
3	Protocol Details.....	12
3.1	ExchangeServicePortType Server Details.....	12
3.1.1	Abstract Data Model.....	12
3.1.2	Timers	12
3.1.3	Initialization.....	12
3.1.4	Message Processing Events and Sequencing Rules	12
3.1.4.1	FindConversation	12
3.1.4.1.1	Messages	13
3.1.4.1.1.1	FindConversationSoapIn	13
3.1.4.1.1.2	FindConversationSoapOut	14
3.1.4.1.2	Elements	14
3.1.4.1.2.1	FindConversation	14
3.1.4.1.2.2	FindConversationResponse	15
3.1.4.1.3	Complex Types	15
3.1.4.1.3.1	ArrayOfConversationsType	15
3.1.4.1.3.2	ArrayOfItemClassType	16
3.1.4.1.3.3	ConversationResponseShapeType	16
3.1.4.1.3.4	ConversationType	17
3.1.4.1.3.5	FindConversationType	20
3.1.4.1.3.6	FindConversationResponseMessageType	22
3.1.4.1.4	Simple Types	23
3.1.4.1.4.1	ConversationQueryTraversalType	23
3.1.4.1.4.2	FlagStatusType.....	24
3.1.4.1.4.3	ViewFilterType.....	24
3.1.4.2	GetConversationItems	25
3.1.4.2.1	Messages	26
3.1.4.2.1.1	GetConversationItemsSoapIn	26
3.1.4.2.1.2	GetConversationItemsSoapOut	26
3.1.4.2.2	Elements	27

3.1.4.2.2.1	GetConversationItems	27
3.1.4.2.2.2	GetConversationItemsResponse	27
3.1.4.2.3	Complex Types	28
3.1.4.2.3.1	ArrayOfConversationNodesType	28
3.1.4.2.3.2	ArrayOfConversationRequestsType	28
3.1.4.2.3.3	ConversationNodeType	29
3.1.4.2.3.4	ConversationResponseType	29
3.1.4.2.3.5	GetConversationItemsType	30
3.1.4.2.3.6	GetConversationItemsResponseMessageType	31
3.1.4.2.3.7	GetConversationItemsResponseType	32
3.1.4.2.3.8	ConversationRequestType	32
3.1.4.2.4	Simple Types	32
3.1.4.2.4.1	ConversationNodeSortOrder	32
3.1.4.3	ApplyConversationAction	33
3.1.4.3.1	Messages	34
3.1.4.3.1.1	ApplyConversationActionSoapIn	34
3.1.4.3.1.2	ApplyConversationActionSoapOut	34
3.1.4.3.2	Elements	35
3.1.4.3.2.1	ApplyConversationAction	35
3.1.4.3.2.2	ApplyConversationActionResponse	35
3.1.4.3.3	Complex Types	36
3.1.4.3.3.1	ApplyConversationActionResponseType	36
3.1.4.3.3.2	ApplyConversationActionType	36
3.1.4.3.3.3	ConversationActionType	37
3.1.4.3.3.4	NonEmptyArrayOfApplyConversationAction	39
3.1.4.3.4	Simple Types	39
3.1.4.3.4.1	ConversationActionTypeType	39
3.1.5	Timer Events	40
3.1.6	Other Local Events	40
4	Protocol Examples	41
5	Security	43
5.1	Security Considerations for Implementers	43
5.2	Index of Security Parameters	43
6	Appendix A: Full WSDL	44
7	Appendix B: Full XML Schema	47
7.1	Messages Schema	47
7.2	Types Schema	49
8	Appendix C: Product Behavior	53
9	Change Tracking	55
10	Index	57

1 Introduction

The Conversations Web Service Protocol enables a protocol client to manage items that can be organized in a conversation. The typical scenarios covered by this protocol involve finding items in a conversation and applying actions to items in a conversation.

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

1.1 Glossary

The following terms are specific to this document:

conversation: A single representation of a send/response series of email messages. A conversation appears in the Inbox as one unit and allows the user to view and read the series of related email messages in a single effort.

endpoint: A communication port that is exposed by an application server for a specific shared service and to which messages can be addressed.

Hypertext Transfer Protocol (HTTP): An application-level protocol for distributed, collaborative, hypermedia information systems (text, graphic images, sound, video, and other multimedia files) on the World Wide Web.

Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS): An extension of **HTTP** that securely encrypts and decrypts webpage requests.

Information Rights Management (IRM): A technology that provides persistent protection to digital data by using encryption, certificates (1), and authentication (2). Authorized recipients or users acquire a license to gain access to the protected files according to the rights or business rules that are set by the content owner.

mailbox: A message store that contains email, calendar items, and other Message objects for a single recipient.

SOAP: A lightweight protocol for exchanging structured information in a decentralized, distributed environment. **SOAP** uses **XML** technologies to define an extensible messaging framework, which provides a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation-specific semantics. SOAP 1.2 supersedes SOAP 1.1. See [\[SOAP1.2-1/2003\]](#).

SOAP action: The HTTP request header field used to indicate the intent of the **SOAP** request, using a URI value. See [\[SOAP1.1\]](#) section 6.1.1 for more information.

SOAP body: A container for the payload data being delivered by a **SOAP message** to its recipient. See [\[SOAP1.2-1/2007\]](#) section 5.3 for more information.

SOAP header: A mechanism for implementing extensions to a **SOAP message** in a decentralized manner without prior agreement between the communicating parties. See [\[SOAP1.2-1/2007\]](#) section 5.2 for more information.

SOAP message: An **XML** document consisting of a mandatory SOAP envelope, an optional **SOAP header**, and a mandatory **SOAP body**. See [\[SOAP1.2-1/2007\]](#) section 5 for more information.

Uniform Resource Locator (URL): A string of characters in a standardized format that identifies a document or resource on the World Wide Web. The format is as specified in [\[RFC1738\]](#).

Web Services Description Language (WSDL): An XML format for describing network services as a set of endpoints that operate on messages that contain either document-oriented or procedure-oriented information. The operations and messages are described abstractly and are bound to a concrete network protocol and message format in order to define an endpoint. Related concrete endpoints are combined into abstract endpoints, which describe a network service. WSDL is extensible, which allows the description of endpoints and their messages regardless of the message formats or network protocols that are used.

WSDL message: An abstract, typed definition of the data that is communicated during a WSDL operation [\[WSDL\]](#). Also, an element that describes the data being exchanged between web service providers and clients.

WSDL port type: A named set of logically-related, abstract **Web Services Description Language (WSDL)** operations and messages.

XML: The Extensible Markup Language, as described in [\[XML1.0\]](#).

XML namespace: A collection of names that is used to identify elements, types, and attributes in XML documents identified in a URI reference [\[RFC3986\]](#). A combination of XML namespace and local name allows XML documents to use elements, types, and attributes that have the same names but come from different sources. For more information, see [\[XMLNS-2ED\]](#).

XML schema: A description of a type of XML document that is typically expressed in terms of constraints on the structure and content of documents of that type, in addition to the basic syntax constraints that are imposed by **XML** itself. An XML schema provides a view of a document type at a relatively high level of abstraction.

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

1.2 References

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

1.2.1 Normative References

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

[MS-OXDCLI] Microsoft Corporation, "[Autodiscover Publishing and Lookup Protocol](#)".

[MS-OXWSADISC] Microsoft Corporation, "[Autodiscover Publishing and Lookup SOAP-Based Web Service Protocol](#)".

[MS-OXWSCDATA] Microsoft Corporation, "[Common Web Service Data Types](#)".

[MS-OXWSCORE] Microsoft Corporation, "[Core Items Web Service Protocol](#)".

[MS-OXWSEDISC] Microsoft Corporation, "[Electronic Discovery \(eDiscovery\) Web Service Protocol](#)".

[MS-OXWSFOLD] Microsoft Corporation, "[Folders and Folder Permissions Web Service Protocol](#)".

[MS-OXWSSRCH] Microsoft Corporation, "[Mailbox Search Web Service Protocol](#)".

[MS-OXWSURPT] Microsoft Corporation, "[Retention Tag Web Service Protocol](#)".

[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>

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000, <http://www.rfc-editor.org/rfc/rfc2818.txt>

[SOAP1.1] Box, D., Ehnebuske, D., Kakivaya, G., et al., "Simple Object Access Protocol (SOAP) 1.1", May 2000, <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>

[WSDL] Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S., "Web Services Description Language (WSDL) 1.1", W3C Note, March 2001, <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>

[XMLNS] Bray, T., Hollander, D., Layman, A., et al., Eds., "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation, December 2009, <http://www.w3.org/TR/2009/REC-xml-names-20091208/>

[XMLSCHEMA1] Thompson, H., Beech, D., Maloney, M., and Mendelsohn, N., Eds., "XML Schema Part 1: Structures", W3C Recommendation, May 2001, <http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/>

[XMLSCHEMA2] Biron, P.V., Ed. and Malhotra, A., Ed., "XML Schema Part 2: Datatypes", W3C Recommendation, May 2001, <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>

1.2.2 Informative References

[MS-OXPROTO] Microsoft Corporation, "[Exchange Server Protocols System Overview](#)".

[RFC2616] Fielding, R., Gettys, J., Mogul, J., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, <http://www.rfc-editor.org/rfc/rfc2616.txt>

1.3 Overview

Conversations are a way to organize related e-mail messages in a **mailbox** based on the subject of the messages. By accessing messages based on the message subject, servers and clients can manage more message items with fewer actions. The Conversations Web Service Protocol enables clients to find message items that are grouped into conversations and then manage the grouped items with a single operation.

1.4 Relationship to Other Protocols

A client that implements this protocol can use the Autodiscover Publishing and Lookup SOAP-Based Web Service Protocol, as described in [\[MS-OXWSADISC\]](#), or the Autodiscover Publishing and Lookup Protocol, as described in [\[MS-OXDCLI\]](#), to identify the target **endpoint** to use for each operation.

This protocol uses the SOAP Protocol, as described in [\[SOAP1.1\]](#), to specify the structure information exchanged between the client and the server. This protocol uses the **XML** Protocol, as described in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#), to describe the message content sent to and from the server.

The Conversations Web Service Protocol uses the SOAP Protocol, as described in [\[SOAP1.1\]](#), over **Hypertext Transfer Protocol (HTTP)**, as described in [\[RFC2616\]](#), or **Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)**, which is described in [\[RFC2818\]](#), as shown in the following layering diagram.

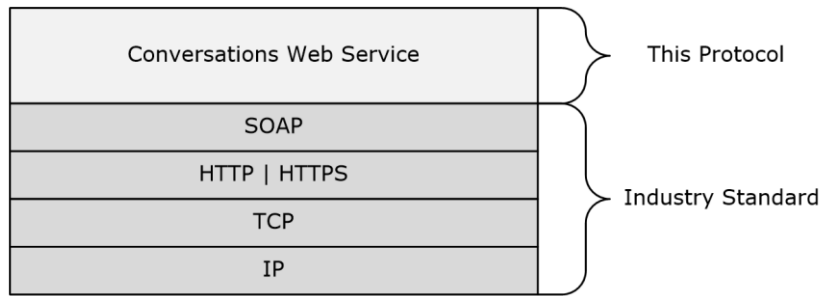


Figure 1: This protocol in relation to other protocols

For conceptual background information and overviews of the relationships and interactions between this and other protocols, see [\[MS-OXPROTO\]](#).

1.5 Prerequisites/Preconditions

The endpoint **URL** that is returned by either the Autodiscover Publishing Lookup SOAP-Based Web Service Protocol, as specified in [\[MS-OXWSADISC\]](#), or the Autodiscover Publishing and Lookup Protocol, as specified in [\[MS-OXDSCLI\]](#), is required to form the Hypertext Transfer Protocol (HTTP) request to the Web server that hosts this protocol. The operations that this protocol defines cannot be accessed unless the correct endpoint is identified in the HTTP Web requests that target this protocol.

1.6 Applicability Statement

The Conversations Web Service Protocol is applicable to environments that use Web services to implement conversations in a user's mailbox. This protocol is applicable to all SOAP-based clients, as described in [\[SOAP1.1\]](#).

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- **Supported Transports:** This protocol uses SOAP 1.1, as specified in section [2.1](#).
- **Protocol Versions:** This protocol specifies only one **WSDL port type** version. The **WSDL** version of the request is identified by using the **t:RequestServerVersion** element, as described in [\[MS-OXWSCDATA\]](#) section 2.2.3.11, and the version of the server responding to the request is identified by using the **t:ServerVersionInfo** element, as described in [\[MS-OXWSCDATA\]](#) section 2.2.3.12.
- **Security and Authentication Methods:** This protocol relies on the Web server that is hosting it to perform authentication.
- **Capability Negotiation:** This protocol does not support version negotiation.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

In the following sections, the schema definition might differ from the processing rules imposed by the protocol. The WSDL in this specification provides a base description of the protocol. The schema in this specification provides a base description of the message syntax. The text that specifies the WSDL and schema might specify restrictions that reflect actual protocol behavior. For example, the schema definition might allow for an element to be **empty**, **null**, or **not present** but the behavior of the protocol as specified restricts the same elements to being **non-empty**, **not null**, or **present**.

2.1 Transport

The SOAP version used for this protocol is SOAP 1.1, as specified in [\[SOAP1.1\]](#).

This protocol relies on the Web server that hosts the application to perform authentication. Protocol servers **MUST** support **SOAP** over HTTP. The protocol servers **SHOULD** additionally support secure communications via HTTPS, as defined in [\[RFC2818\]](#).

2.2 Common Message Syntax

This section contains common definitions that are used by this protocol. The syntax of the definitions uses **XML schema**, as defined in [\[XMLSCHEMA1\]](#) and [\[XMLSCHEMA2\]](#), and Web Services Description Language (WSDL), as defined in [\[WSDL\]](#).

2.2.1 Namespaces

This specification defines and references various **XML namespaces** by using the mechanisms specified in [\[XMLNS\]](#). Although this specification associates a specific XML namespace prefix for each XML namespace that is used, the choice of any particular XML namespace prefix is implementation-specific and not significant for interoperability.

Prefix	Namespace URI	Reference
soap	http://schemas.xmlsoap.org/wsdl/soap/	[SOAP1.1]
tns	http://schemas.microsoft.com/exchange/services/2006/messages	
wsdl	http://schemas.xmlsoap.org/wsdl/	[WSDL]
t	http://schemas.microsoft.com/exchange/services/2006/types	
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1]
m	http://schemas.microsoft.com/exchange/services/2006/messages	
none	http://schemas.microsoft.com/exchange/services/2006/messages	
none	http://schemas.microsoft.com/exchange/services/2006/types	

2.2.2 Messages

This specification does not define any common **WSDL message** definitions.

2.2.3 Elements

This specification does not define any common XML schema element definitions.

2.2.4 Complex Types

This specification does not define any common XML schema complex type definitions.

2.2.5 Simple Types

This specification does not define any common XML schema simple type definitions.

2.2.6 Attributes

This specification does not define any common XML schema attribute definitions.

2.2.7 Groups

This specification does not define any common XML schema group definitions.

2.2.8 Attribute Groups

This specification does not define any common XML schema attribute group definitions.

Preliminary

3 Protocol Details

The client side of this protocol is simply a pass-through. That is, no additional timers or other state is required on the client side of this protocol. Calls made by the higher-layer protocol or application are passed directly to the transport, and the results returned by the transport are passed directly back to the higher-layer protocol or application.

3.1 ExchangeServicePortType Server Details

The Conversations Web Service Protocol defines a single WSDL port type.

3.1.1 Abstract Data Model

None.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Message Processing Events and Sequencing Rules

The following table summarizes the WSDL operations as defined by this specification.

Operation name	Description
ApplyConversationAction (section 3.1.4.3)	Specifies the WSDL messages for applying conversation actions.
FindConversation (section 3.1.4.1)	Specifies the WSDL messages for finding conversations.
GetConversationItems (section 3.1.4.2)	Specifies the WSDL messages that are used to request and return the contents of a conversation in a mailbox.

The **ApplyConversationAction** operation requires a target conversation. The target conversation can be identified by the **FindConversation** operation.

3.1.4.1 FindConversation

The **FindConversation** operation specifies the WSDL messages for finding conversations in a mailbox.

The following is the WSDL port type definition of this operation.

```
<wsdl:operation name="FindConversation">
  <wsdl:input message="tns:FindConversationSoapIn"/>
  <wsdl:output message="tns:FindConversationSoapOut"/>
</wsdl:operation>
```

The following is the WSDL binding specification for this operation.

```

<wsdl:operation name="FindConversation">
  <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/FindConversation" />
  <wsdl:input>
    <soap:body parts="request" use="literal" />
    <soap:header message="tns:FindConversationSoapIn" part="RequestVersion" use="literal"
  />
    <soap:header message="tns:FindConversationSoapIn" part="Impersonation" use="literal"
  />
  </wsdl:input>
  <wsdl:output>
    <soap:body parts="FindConversationResult" use="literal" />
    <soap:header message="tns:FindConversationSoapOut" part="ServerVersion" use="literal"
  />
  </wsdl:output>
</wsdl:operation>

```

3.1.4.1.1 Messages

The following table lists and describes the WSDL message definitions that are specific to the **FindConversation** operation.

Message	Description
FindConversationSoapIn (section 3.1.4.1.1.1)	Specifies the SOAP message that represents a request for a collection of conversations.
FindConversationSoapOut (section 3.1.4.1.1.2)	Specifies the SOAP message that contains a collection of conversations.

3.1.4.1.1.1 FindConversationSoapIn

The **FindConversationSoapIn** WSDL message specifies the SOAP message that represents a request to find a collection of conversations.

```

<wsdl:message name="FindConversationSoapIn">
  <wsdl:part name="request" element="tns:FindConversation"/>
  <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
  <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
</wsdl:message>

```

The **FindConversationSoapIn** WSDL message is the input message for the **SOAP action** <http://schemas.microsoft.com/exchange/services/2006/messages/FindConversation>.

The parts of the **FindConversationSoapIn** WSDL message are listed and described in the following table.

Part	Element/type	Description
request	tns:FindConversation (section 3.1.4.1.2.1)	Specifies the SOAP body of the request to find a conversation. This part is required.
Impersonation	t:ExchangeImpersonation ([MS-OXWSCDATA] section 2.2.3.3)	Specifies a SOAP header that identifies the user whom the client application is impersonating. This part is optional.

Part	Element/type	Description
RequestVersion	t:RequestServerVersion ([MS-OXWSCDATA] section 2.2.3.11)	Specifies a SOAP header that identifies the schema version for the FindConversation operation request. This part is optional.

3.1.4.1.1.2 FindConversationSoapOut

The **FindConversationSoapOut** WSDL message specifies the server response SOAP message that contains a collection of conversations.

```
<wsdl:message name="FindConversationSoapOut">
  <wsdl:part name="FindConversationResult" element="tns:FindConversationResponse"/>
  <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
```

The **FindConversationSoapOut** WSDL message is the output message for the SOAP action <http://schemas.microsoft.com/exchange/services/2006/messages/FindConversation>.

The parts of the **FindConversationSoapOut** WSDL message are listed and described in the following table.

Part	Element/type	Description
FindConversationResult	FindConversationResponse (section 3.1.4.1.2.2)	Specifies the SOAP body of the response to a FindConversation operation request.
ServerVersion	ServerVersionInfo ([MS-OXWSCDATA] section 2.2.3.12)	Specifies the SOAP header that identifies the server version for the response.

3.1.4.1.2 Elements

The following table lists and describes the XML schema element definitions that are specific to the **FindConversation** operation.

Element	Description
FindConversation (section 3.1.4.1.2.1)	Specifies a request to find conversations in a mailbox.
FindConversationResponse (section 3.1.4.1.2.2)	Specifies a response that contains a set of conversations found in a mailbox.

3.1.4.1.2.1 FindConversation

The **FindConversation** element specifies a request to find conversations in a mailbox.

```
<xs:element name="FindConversation"
  type="m:FindConversationType"
/>
```

3.1.4.1.2.2 FindConversationResponse

The **FindConversationResponse** element specifies a response that contains the results of an attempt to find a conversation in a mailbox.

```
<xs:element name="FindConversationResponse"
  type="m:FindConversationResponseMessageType"
  />
```

3.1.4.1.3 Complex Types

The following table lists the XML schema complex type definitions that are specific to the **FindConversation** operation.

Complex type	Description
ArrayOfConversationsType (section 3.1.4.1.3.1)	Specifies an array of conversations that are returned in the response.
ArrayOfItemClassType (section 3.1.4.1.3.2)	Specifies an array of unique item classes that are used by the set of items in the conversation.
ConversationResponseShapeType (section 3.1.4.1.3.3)	Specifies the properties that are returned for conversation items returned in the response.
ConversationType (section 3.1.4.1.3.4)	Specifies a single conversation.
FindConversationType (section 3.1.4.1.3.5)	Specifies the search options to be used in searches for conversations.
FindConversationResponseMessageType (section 3.1.4.1.3.6)	Specifies the results of an attempt to find conversations in a folder.

3.1.4.1.3.1 ArrayOfConversationsType

The **ArrayOfConversationsType** complex type specifies an array of conversations that are returned in the response.

```
<xs:complexType name="ArrayOfConversationsType">
  <xs:choice
    minOccurs="0"
    maxOccurs="unbounded"
  >
    <xs:element name="Conversation"
      type="t:ConversationType"
    />
  </xs:choice>
</xs:complexType>
```

The following table lists and describes the child element of the **ArrayOfConversationsType** complex type.

Element	Type	Description
Conversation	t:ConversationType (section 3.1.4.1.3.4)	Specifies a single conversation.

3.1.4.1.3.2 ArrayOfItemClassType

The **ArrayOfItemClassType** complex type specifies an array of unique item classes.

```
<xs:complexType name="ArrayOfItemClassType">
  <xs:choice
    minOccurs="0"
    maxOccurs="unbounded"
  >
    <xs:element name="ItemClass"
      type="t:ItemClassType"
    />
  </xs:choice>
</xs:complexType>
```

The following table lists the child element of the **ArrayOfConversationsType** complex type.

Element	Type	Description
ItemClass	t:ItemClassType ([MS-OXWSCORE] section 2.2.5.4)	Specifies a unique item class.

3.1.4.1.3.3 ConversationResponseShapeType

The **ConversationResponseShapeType** complex type specifies the properties that are returned for conversation items returned in the response. [<1>](#)

```
<xs:complexType name="ConversationResponseShapeType">
  <xs:sequence>
    <xs:element name="BaseShape" type="t:DefaultShapeNamesType" />
    <xs:element name="AdditionalProperties" type="t:NonEmptyArrayOfPathsToElementType"
      minOccurs="0" />
  </xs:sequence>
</xs:complexType>
```

The following table lists the elements of the **ConversationResponseShapeType** complex type.

Element	Type	Description
BaseShape	t:DefaultShapeNamesType ([MS-OXWSCDATA] section 2.2.5.7)	Specifies the base properties to return in the response.
AdditionalProperties	t:NonEmptyArrayOfPathsToElementType ([MS-OXWSCDATA] section 2.2.4.48)	Specifies the additional properties to be returned in the response.

3.1.4.1.3.4 ConversationType

The **ConversationType** complex type specifies a single conversation.

```
<xs:complexType name="ConversationType">
  <xs:sequence>
    <xs:element name="ConversationId" type="t:ItemIdType" minOccurs="0"/>
    <xs:element name="ConversationTopic" type="xs:string" minOccurs="0"/>
    <xs:element name="UniqueRecipients" type="t:ArrayOfStringsType" minOccurs="0"/>
    <xs:element name="GlobalUniqueRecipients" type="t:ArrayOfStringsType" minOccurs="0"/>
    <xs:element name="UniqueUnreadSenders" type="t:ArrayOfStringsType" minOccurs="0"/>
    <xs:element name="GlobalUniqueUnreadSenders" type="t:ArrayOfStringsType"
minOccurs="0"/>
    <xs:element name="UniqueSenders" type="t:ArrayOfStringsType" minOccurs="0"/>
    <xs:element name="GlobalUniqueSenders" type="t:ArrayOfStringsType" minOccurs="0"/>
    <xs:element name="LastDeliveryTime" type="xs:dateTime" minOccurs="0"/>
    <xs:element name="GlobalLastDeliveryTime" type="xs:dateTime" minOccurs="0"/>
    <xs:element name="Categories" type="t:ArrayOfStringsType" minOccurs="0"/>
    <xs:element name="GlobalCategories" type="t:ArrayOfStringsType" minOccurs="0"/>
    <xs:element name="FlagStatus" type="t:FlagStatusType" minOccurs="0"/>
    <xs:element name="GlobalFlagStatus" type="t:FlagStatusType" minOccurs="0"/>
    <xs:element name="HasAttachments" type="xs:boolean" minOccurs="0"/>
    <xs:element name="GlobalHasAttachments" type="xs:boolean" minOccurs="0"/>
    <xs:element name="MessageCount" type="xs:int" minOccurs="0"/>
    <xs:element name="GlobalMessageCount" type="xs:int" minOccurs="0"/>
    <xs:element name="UnreadCount" type="xs:int" minOccurs="0"/>
    <xs:element name="GlobalUnreadCount" type="xs:int" minOccurs="0"/>
    <xs:element name="Size" type="xs:int" minOccurs="0"/>
    <xs:element name="GlobalSize" type="xs:int" minOccurs="0"/>
    <xs:element name="ItemClasses" type="t:ArrayOfItemClassType" minOccurs="0"/>
    <xs:element name="GlobalItemClasses" type="t:ArrayOfItemClassType" minOccurs="0"/>
    <xs:element name="Importance" type="t:ImportanceChoicesType" minOccurs="0"/>
    <xs:element name="GlobalImportance" type="t:ImportanceChoicesType" minOccurs="0"/>
    <xs:element name="ItemIds" type="t:NonEmptyArrayOfBaseItemIdsType" minOccurs="0"/>
    <xs:element name="GlobalItemIds" type="t:NonEmptyArrayOfBaseItemIdsType"
minOccurs="0"/>
    <xs:element name="LastModifiedTime" type="xs:dateTime" minOccurs="0"/>
    <xs:element name="InstanceKey" type="xs:base64Binary" minOccurs="0"/>
    <!-- Maps to ConversationItemSchema.ConversationPreview-->
    <xs:element name="Preview" type="xs:string" minOccurs="0"/>
    <!-- Maps to ConversationItemSchema.NextPredictedAction-->
    <xs:element name="NextPredictedAction" type="t:PredictedMessageActionType"
minOccurs="0" />
    <!-- Maps to ConversationItemSchema.GroupingAction-->
    <xs:element name="GroupingAction" type="t:PredictedMessageActionType" minOccurs="0"/>
    <!-- Flag for the mailbox scope this conversation was found in -->
    <xs:element name="MailboxScope" type="t:MailboxSearchLocationType" minOccurs="0"/>
    <!-- Maps to ConversationItemSchema.ConversationIconIndex-->
    <xs:element name="IconIndex" type="t:IconIndexType" minOccurs="0"/>
    <!-- Maps to ConversationItemSchema.ConversationGlobalIconIndex-->
    <xs:element name="GlobalIconIndex" type="t:IconIndexType" minOccurs="0"/>
    <!-- Maps to ConversationItemSchema.ConversationDraftItemIds-->
    <xs:element name="DraftItemIds" type="t:NonEmptyArrayOfBaseItemIdsType" minOccurs="0"/>
    <!-- Maps to ConversationItemSchema.ConversationHasIrm-->
    <xs:element name="HasIrm" type="xs:boolean" minOccurs="0"/>
    <!-- Maps to ConversationItemSchema.ConversationGlobalHasIrm-->
    <xs:element name="GlobalHasIrm" type="xs:boolean" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

The following table lists and describes the child elements of the **ConversationType** complex type.

Element	Type	Description
ConversationId	t:ItemIdType ([MS-OXWSCORE] section 2.2.4.24)	Specifies the conversation identifier.
ConversationTopic	xs:string [XMLSCHEMA2]	Specifies the conversation topic.
UniqueRecipients	t:ArrayOfStringsType ([MS-OXWSCDATA] section 2.2.4.13)	Specifies a list of all the people who have received messages in this conversation in the current folder.
GlobalUniqueRecipients	t:ArrayOfStringsType	Specifies a list of all the people who have received messages in this conversation across all folders in the mailbox.
UniqueUnreadSenders	t:ArrayOfStringsType	Specifies a list of all the people who have sent messages that are currently unread in this conversation in the specified folder.
GlobalUniqueUnreadSenders	t:ArrayOfStringsType	Specifies a list of all the people who have sent messages that are currently unread in this conversation across all folders in a mailbox.
UniqueSenders	t:ArrayOfStringsType	Specifies a list of all the senders of conversation items in the specified folder.
GlobalUniqueSenders	t:ArrayOfStringsType	Specifies a list of all the senders of conversation items in a mailbox.
LastDeliveryTime	xs:dateTime [XMLSCHEMA2]	Specifies the delivery time of the last message received in this conversation in the specified folder.
GlobalLastDeliveryTime	xs:dateTime	Specifies the delivery time of the last message received in this conversation across all folders in a mailbox.
Categories	t:ArrayOfStringsType	Specifies the category list for conversation items in the folder that is currently open.
GlobalCategories	t:ArrayOfStringsType	Specifies the category list for all conversation items in a mailbox.
FlagStatus	t:FlagStatusType (section 3.1.4.1.4.2)	Specifies the aggregated flag status for conversation items in the folder that is currently open.
GlobalFlagStatus	t:FlagStatusType	Specifies the aggregated flag status for all conversation

Element	Type	Description
		items in mailbox.
HasAttachments	xs:boolean [XMLSCHEMA2]	Specifies a value that indicates whether at least one conversation item in the folder that is currently open has an attachment.
GlobalHasAttachments	xs:boolean	Specifies a value that indicates whether at least one conversation item in a mailbox has an attachment.
MessageCount	xs:int [XMLSCHEMA2]	Specifies the total number of conversation items in the folder that is currently open.
GlobalMessageCount	xs:int	Specifies the total number of conversation items in a mailbox.
UnreadCount	xs:int	Specifies a count of all the unread conversation items in the folder that is currently open.
GlobalUnreadCount	xs:int	Specifies a count of all the unread conversation items in a mailbox.
Size	xs:int	Specifies the size of the conversation calculated from the size of all conversation items in the folder that is currently open.
GlobalSize	xs:int	Specifies the size of the conversation calculated from the size of all conversation items in a mailbox.
ItemClasses	t:ArrayOfItemClassType (section 3.1.4.1.3.2)	Specifies a list of item classes that represents all the item classes of the conversation items in the folder that is currently open.
GlobalItemClasses	t:ArrayOfItemClassType	Specifies a list of item classes that represents all the item classes of the conversation items in a mailbox.
Importance	t:ImportanceChoicesType ([MS-OXWSCORE] section 2.2.5.3)	Specifies the aggregated importance for all conversation items in the folder that is currently open.
GlobalImportance	t:ImportanceChoicesType	Specifies the aggregated importance for all conversation items in mailbox.

Element	Type	Description
ItemIds	t:NonEmptyArrayOfBaseItemIdsType ([MS-OXWSCORE] section 2.2.4.30)	Specifies the collection of item identifiers for all conversation items in the folder that is currently open.
GlobalItemIds	t:NonEmptyArrayOfBaseItemIdsType	Specifies the collection of item identifiers for all conversation items in mailbox.
LastModifiedTime	xs:dateTime	Specifies the time at which the item was last modified. <2>
InstanceKey	xs:base64Binary	Specifies the instance key. <3>
Preview	xs:string	Specifies the first 256 characters of the message so that it can be previewed without being opened.
MailboxScope	t:MailboxSearchLocationType ([MS-OXWSEDISC] section 3.1.4.4.4.1)	Specifies the mailbox scope. <4>
IconIndex	t:IconIndexType ([MS-OXWSCDATA] section 2.2.5.16)	Specifies the index of the icon to display. <5>
GlobalIconIndex	t:IconIndexType	Specifies the aggregate IconIndex element for all messages in a conversation. <6>
DraftItemIds	t:NonEmptyArrayOfBaseItemIdsType	Specifies an array of identifiers for draft items. <7>
HasIrm	xs:boolean	Specifies whether at least one message in this conversation , and in the current folder only, is an Information Rights Management (IRM) protected message. <8>
GlobalHasIrm	xs:boolean	Specifies whether at least one message in this conversation, across all folders in the mailbox, is an IRM protected message. <9>

3.1.4.1.3.5 FindConversationType

The **FindConversationType** complex type specifies the search options to use when searching for conversations. The **FindConversationType** complex type extends the **BaseRequestType** complex type, as specified in [MS-OXWSCDATA] section 2.2.4.17.

```
<xs:complexType name="FindConversationType">
  <xs:complexContent>
    <xs:extension
      base="m:BaseRequestType"
    >
    <xs:sequence>
```

```

<xs:choice minOccurs="0">
  <xs:element name="IndexedPageItemView"
    type="t:IndexedPageViewType"
  />
  <xs:element name="SeekToConditionPageItemView"
    type="t:SeekToConditionPageViewType"
  />
</xs:choice>
<xs:element name="SortOrder"
  type="t:NonEmptyArrayOfFieldOrdersType"
  minOccurs="0"
/>
<xs:element name="ParentFolderId"
  type="t:TargetFolderIdType"
  minOccurs="1"
/>
<xs:element name="MailboxScope"
  type="t:MailboxSearchLocationType"
  minOccurs="0"
  maxOccurs="1"
/>
<xs:element name="QueryString"
  type="m:QueryStringType"
  minOccurs="0"
  maxOccurs="1"
/>
<xs:element name="ConversationShape"
  type="t:ConversationResponseShapeType"
  minOccurs="0"
  maxOccurs="1"
/>
</xs:sequence>
<xs:attribute name="Traversal"
  type="t:ConversationQueryTraversalType"
  use="optional"
/>
<xs:attribute name="ViewFilter"
  type="t:ViewFilterType"
  use="optional"
/>
</xs:extension>
</xs:complexContent>
</xs:complexType>

```

The following table lists the child elements of the **FindConversationType** complex type.

Element	Type	Description
IndexedPageItemView	t:IndexedPageViewType ([MS-OXWSSRCH] section 2.2.4.16)	Specifies how conversation search results are returned in a response. This element MUST be present.
SeekToConditionPageItemView	t:SeekToConditionPageViewType ([MS-OXWSSRCH] section 2.2.4.31)	Specifies a condition that has to be satisfied in order to include an item in the result set. <10>
SortOrder	t:NonEmptyArrayOfFieldOrdersType ([MS-OXWSSRCH] section 3.1.4.2.3.8)	Specifies the property and sort order direction that is used to sort conversations in a response.

Element	Type	Description
ParentFolderId	t:TargetFolderIdType ([MS-OXWSFOLD] section 2.2.4.14)	Specifies the folder that is searched for conversations. This element MUST be present.
MailboxScope	t:MailboxSearchLocationType ([MS-OXWSEDISC] section 3.1.4.4.4.1)	Specifies the mailbox scope.
QueryString	m:QueryStringType ([MS-OXWSSRCH] section 2.2.4.33)	Specifies the query string.
ConversationShape	t:ConversationResponseShapeType (section 3.1.4.1.3.3)	Specifies the properties returned in a response.

The following table lists the attributes of the **FindConversationType** complex type.

Attribute	Type	Description
Traversal	t:ConversationQueryTraversalType (section 3.1.4.1.4.1)	Specifies whether a query of conversation items is shallow or deep.
ViewFilter	t:ViewFilterType	Specifies the view filter for the search.

3.1.4.1.3.6 FindConversationResponseMessageType

The **FindConversationResponseMessageType** complex type specifies the results of an attempt to find conversations in the specified folder. The **FindConversationResponseMessageType** complex type extends the **ResponseMessageType** complex type, as specified in [\[MS-OXWSCDATA\]](#) section 2.2.4.65.

```

<xs:complexType name="FindConversationResponseMessageType">
  <xs:complexContent>
    <xs:extension
      base="m:ResponseMessageType"
    >
      <xs:sequence>
        <xs:element name="Conversations"
          type="t:ArrayOfConversationsType"
          minOccurs="0"
        />
        <xs:element name="HighlightTerms"
          type="t:ArrayOfHighlightTermsType"
          minOccurs="0"
        />
        <xs:element name="TotalConversationsInView"
          type="xs:int"
          minOccurs="0"
        />
        <xs:element name="IndexedOffset"
          type="xs:int"
          minOccurs="0"
        />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

The following table lists and describes the child elements of the **FindConversationResponseMessageType** complex type.

Element	Type	Description
Conversations	t:ArrayOfConversationsType (section 3.1.4.1.3.1)	Specifies an array of conversations that is returned in a response. This element MUST be present.
HighlightTerms	t:ArrayOfHighlightTermsType ([MS-OXWSCDATA] section 2.2.4.8)	Specifies an array of highlight terms.
TotalConversationsInView	xs:int ([XMLSCHEMA2])	Specifies the total number of conversations returned.
IndexedOffset	xs:int	Specifies the indexed offset from the origin of the search results.

3.1.4.1.4 Simple Types

The following table lists the XML schema simple definitions that are specific to the **FindConversation** operation.

Simple type	Definition
ConversationQueryTraversalType (section 3.1.4.1.4.1)	Specifies whether a query of conversation items is shallow or deep.
FlagStatusType (section 3.1.4.1.4.2)	Specifies the flag status for conversation items.
ViewFilterType (section 3.1.4.1.4.3)	Specifies a filter for conversation items returned in a response

3.1.4.1.4.1 ConversationQueryTraversalType

The **ConversationQueryTraversalType** simple type specifies whether a query of conversation items is shallow or deep. [<11>](#)

```
<xs:simpleType name="ConversationQueryTraversalType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Shallow" />
    <xs:enumeration value="Deep" />
  </xs:restriction>
</xs:simpleType>
```

The following table lists and describes the enumeration values that are defined by the **ConversationQueryTraversalType** simple type.

Value	Description
Shallow	Specifies a shallow query.
Deep	Specifies a deep query.

3.1.4.1.4.2 FlagStatusType

The **FlagStatusType** simple type specifies the flag status for conversation items.

```
<xs:simpleType name="FlagStatusType">
  <xs:restriction
    base="xs:string"
  >
    <xs:enumeration
      value="NotFlagged"
    />
    <xs:enumeration
      value="Flagged"
    />
    <xs:enumeration
      value="Complete"
    />
  </xs:restriction>
</xs:simpleType>
```

The following table lists and describes the enumeration values that are defined by the **FlagStatusType** simple type.

Value	Description
NotFlagged	This value sets the flag status to "NotFlagged".
Flagged	This value sets the flag status to "Flagged".
Complete	This value sets the flag status to "Complete".

3.1.4.1.4.3 ViewFilterType

The **ViewFilterType** simple type specifies a filter for conversation items returned in a response. [<12>](#)

```
<xs:simpleType name="ViewFilterType">
  <xs:annotation>
    <xs:documentation>Types of view filters for finding
items/conversations</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="All" />
    <xs:enumeration value="Flagged" />
    <xs:enumeration value="HasAttachment" />
    <xs:enumeration value="ToOrCcMe" />
    <xs:enumeration value="Unread" />
    <xs:enumeration value="TaskActive" />
    <xs:enumeration value="TaskOverdue" />
    <xs:enumeration value="TaskCompleted" />
    <xs:enumeration value="Suggestions" />
    <xs:enumeration value="SuggestionsRespond" />
    <xs:enumeration value="SuggestionsDelete" />
    <xs:enumeration value="NoClutter"/>
    <xs:enumeration value="Clutter"/>
  </xs:restriction>
</xs:simpleType>
```


The following table lists and describes the enumeration values that are defined by the **ViewFilterType** simple type.

Value	Description
All	Specifies that no filter is applied.
Flagged	Specifies a filter for all flagged conversation items.
HasAttachment	Specifies a filter for all conversation items with attachments.
ToOrCcMe	Specifies a filter for all conversation items with the receiver specified on the To and Cc recipient list.
Unread	Specifies a filter for all unread conversation items.
TaskActive	Specifies a filter for all active tasks.
TaskOverdue	Specifies a filter for all overdue tasks.
TaskCompleted	Specifies a filter for all completed tasks.
Suggestions	Specifies a filter for all suggestions. <13>
SuggestionsRespond	Specifies a filter for suggestions and items to respond to. <14>
SuggestionsDelete	Specifies a filter for suggestions and items to delete. <15>
NoClutter	For internal use only. <16>
Clutter	For internal use only. <17>

3.1.4.2 GetConversationItems

The **GetConversationItems** operation specifies the WSDL messages that are used to request and return the contents of a conversation in a mailbox. <18>

The following is the WSDL port type definition of this operation.

```
<wsdl:operation name="GetConversationItems">
  <wsdl:input message="tns:GetConversationItemsSoapIn"/>
  <wsdl:output message="tns:GetConversationItemsSoapOut"/>
</wsdl:operation>
```

The following is the WSDL binding specification of this operation.

```
<wsdl:operation name="GetConversationItems">
  <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/GetConversationItems
"/>
  <wsdl:input>
    <soap:body parts="request" use="literal"/>
    <soap:header message="tns:GetConversationItemsSoapIn" part="RequestVersion"
use="literal"/>
    <soap:header message="tns:GetConversationItemsSoapIn" part="Impersonation"
use="literal"/>
  </wsdl:input>
  <wsdl:output>
```

```

    <soap:body parts="GetConversationItemsResult" use="literal"/>
    <soap:header message="tns:GetConversationItemsSoapOut" part="ServerVersion"
use="literal"/>
  </wsdl:output>
</wsdl:operation>

```

3.1.4.2.1 Messages

The following table lists and describes the set of the WSDL message definitions that are specific to the **GetConversationItems** operation.

Message	Description
GetConversationItemsSoapIn (section 3.1.4.2.1.1)	Specifies the SOAP message that represents a request for the contents of a conversation.
GetConversationItemsSoapOut (section 3.1.4.2.1.2)	Specifies the SOAP message that contains the contents of a conversation.

3.1.4.2.1.1 GetConversationItemsSoapIn

The **GetConversationItemsSoapIn** WSDL message specifies the SOAP message that represents a request to return the contents of a conversation.

```

<wsdl:message name="GetConversationItemsSoapIn">
  <wsdl:part name="request" element="tns:GetConversationItems" />
  <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
  <wsdl:part name="RequestVersion" element="t:RequestServerVersion" />
</wsdl:message>

```

The **GetConversationItemsSoapIn** WSDL message is the input message for the SOAP action <http://schemas.microsoft.com/exchange/services/2006/messages/GetConversationItems>.

The parts of the **GetConversationItemsSoapIn** WSDL message are listed and described in the following table.

Part	Element/type	Description
Request	tns:GetConversationItems (section 3.1.4.2.2.1)	Specifies the SOAP body of the request to return the contents of a conversation. This part is required.
Impersonation	t:ExchangeImpersonation ([MS-OXWSCDATA] section 2.2.3.3)	Specifies a SOAP header that identifies the user whom the client application is impersonating. This part is optional.
RequestVersion	t:RequestServerVersion ([MS-OXWSCDATA] section 2.2.3.11)	Specifies a SOAP header that identifies the schema version for the GetConversationItems operation request. This part is optional.

3.1.4.2.1.2 GetConversationItemsSoapOut

The **GetConversationItemsSoapOut** WSDL message specifies the server response SOAP message that contains the contents of a conversation.

```

<wsdl:message name="GetConversationItemsSoapOut">
  <wsdl:part name="GetConversationItemsResult"
    element="tns:GetConversationItemsResponse" />
  <wsdl:part name="ServerVersion" element="t:ServerVersionInfo" />
</wsdl:message>

```

The **GetConversationItemsSoapOut** WSDL message is the output message for the SOAP action <http://schemas.microsoft.com/exchange/services/2006/messages/GetConversationItems>.

The parts of the **GetConversationItemsSoapOut** WSDL message are listed and described in the following table.

Part	Element/type	Description
GetConversationItemsResult	tns:GetConversationItemsResponse (section 3.1.4.2.2.2)	Specifies the SOAP body of the response to a GetConversationItems operation request.
ServerVersion	ServerVersionInfo ([MS-OXWSCDATA] section 2.2.3.12)	Specifies the SOAP header that identifies the server version for the response.

3.1.4.2.2 Elements

The following table lists and describes the XML schema element definitions that are specific to the **GetConversationItems** operation.

Element	Description
GetConversationItems (section 3.1.4.2.2.1)	Specifies a request to return the contents of a conversation in a mailbox.
GetConversationItemsResponse (section 3.1.4.2.2.2)	Specifies a response that contains the results of an attempt to return the contents of a conversation in a mailbox.

3.1.4.2.2.1 GetConversationItems

The **GetConversationItems** element specifies a request to return the contents of a conversation in a mailbox.

```

<xs:element name="GetConversationItems"
  type="m:GetConversationItemsType"
/>

```

3.1.4.2.2.2 GetConversationItemsResponse

The **GetConversationItemsResponse** element specifies a response that contains the results of an attempt to return the contents of a conversation in a mailbox.

```

<xs:element name="GetConversationItemsResponse"
  type="m:GetConversationItemsResponseType"

```

/>

3.1.4.2.3 Complex Types

The following table lists and describes the XML schema complex type definitions that are specific to the **GetConversationItems** operation.

Complex type	Description
m:GetConversationItemsType (section 3.1.4.2.3.5)	Specifies the request to retrieve conversation items.
m:GetConversationItemsResponseMessageType (section 3.1.4.2.3.6)	Specifies the response message for the GetConversationItems operation.
m:GetConversationItemsResponseType (section 3.1.4.2.3.7)	Specifies the response for the GetConversationItems operation.

3.1.4.2.3.1 ArrayOfConversationNodesType

The **ArrayOfConversationNodesType** complex type specifies an array of conversation nodes. [<19>](#)

```
<xs:complexType name="ArrayOfConversationNodesType">  
  <xs:choice minOccurs="0" maxOccurs="unbounded">  
    <xs:element name="ConversationNode" type="t:ConversationNodeType"/>  
  </xs:choice>  
</xs:complexType>
```

The following table lists the child elements of the **ArrayOfConversationNodesType** complex type.

Element	Type	Description
ConversationNode	t:ConversationNodeType (section 3.1.4.2.3.3)	Specifies a single conversation node.

3.1.4.2.3.2 ArrayOfConversationRequestsType

The **ArrayOfConversationRequestsType** complex type specifies one or more conversations to be returned in a response. [<20>](#)

```
<xs:complexType name="ArrayOfConversationRequestsType">  
  <xs:choice minOccurs="0" maxOccurs="unbounded">  
    <xs:element name="Conversation" type="t:ConversationRequestType"/>  
  </xs:choice>  
</xs:complexType>
```

The following table lists the child elements of the **ArrayOfConversationRequestsType** complex type.

Element	Type	Description
Conversation	t:ConversationRequestType (section 3.1.4.2.3.8)	Specifies a conversation to return in a response.

3.1.4.2.3.3 ConversationNodeType

The **ConversationNodeType** complex type specifies a node of items in a conversation. [<21>](#)

```
<xs:complexType name="ConversationNodeType">
  <xs:sequence>
    <xs:element name="InternetMessageId" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="ParentInternetMessageId" type="xs:string" minOccurs="0"
maxOccurs="1"/>
    <xs:element name="Items" type="t:NonEmptyArrayOfAllItemsType" minOccurs="0"
maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
```

The following table lists the child elements of the **ConversationNodeType** complex type.

Element	Type	Description
InternetMessageId	xs:string (XMLSCHEMA21)	Specifies the internet message identifier of this conversation node.
ParentInternetMessageId	xs:string	Specifies the internet message identifier of the parent conversation node.
Items	t:NonEmptyArrayOfAllItemsType (IMS-OXWSCDATA1 section 2.2.4.46)	Specifies all the items in the conversation node.

3.1.4.2.3.4 ConversationResponseType

The **ConversationResponseType** complex type specifies a conversation that is returned in a response. [<22>](#)

```
<xs:complexType name="ConversationResponseType">
  <xs:sequence>
    <xs:element name="ConversationId" type="t:ItemIdType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="SyncState" type="xs:base64Binary" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
```

```

    <xs:element name="ConversationNodes" type="t:ArrayOfConversationNodesType"
minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

The following table lists the child elements of the **ConversationResponseType** complex type.

Element	Type	Description
ConversationId	t:ItemIdType	Specifies the conversation identifier.
SyncState	xs:base64Binary	Specifies the conversation synchronization state.
ConversationNodes	t:ArrayOfConversationNodesType (section 3.1.4.2.3.1)	Specifies the conversation nodes in the conversation.

3.1.4.2.3.5 GetConversationItemsType

The **GetConversationItemsType** complex type specifies the request to return the contents of a conversation. The **GetConversationItemsType** complex type extends the **BaseRequestType**, as specified in [\[MS-OXWSCDATA\]](#) section 2.2.4.17.

```

<xs:complexType name="GetConversationItemsType">
  <xs:complexContent>
    <xs:extension
      base="m:BaseRequestType"
    >
      <xs:sequence>
        <xs:element name="ItemShape"
          type="t:ItemResponseShapeType"
          minOccurs="1"
          maxOccurs="1"
        />
        <xs:element name="FoldersToIgnore"
          type="t:NonEmptyArrayOfBaseFolderIdsType"
          minOccurs="0"
          maxOccurs="1"
        />
        <xs:element name="MaxItemsToReturn"
          type="xs:int"
          minOccurs="0"
          maxOccurs="1"
        />
        <xs:element name="SortOrder"
          type="t:ConversationNodeSortOrder"
          minOccurs="0"
          maxOccurs="1"
        />
        <xs:element name="MailboxScope"
          type="t:MailboxSearchLocationType"
          minOccurs="0"
          maxOccurs="1"
        />
        <xs:element name="Conversations"
          type="t:ArrayOfConversationRequestsType"
          minOccurs="1"
        />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```

    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

The following table lists the child elements of the **GetConversationItemsType** complex type

Element	Type	Description
ItemShape	t:ItemResponseShapeType ([MS-OXWSCDATA] section 2.2.4.42)	Specifies the properties to return.
FoldersToIgnore	t:NonEmptyArrayOfBaseFolderIdsType ([MS-OXWSCORE] section 2.2.4.30)	Specifies the folders to be ignored in the retrieving conversation items.
MaxItemsToReturn	xs:int	Specifies the maximum number of items to return.
SortOrder	t:ConversationNodeSortOrder (section 3.1.4.2.4.1)	Specifies the sort order for conversation nodes.
MailboxScope	t:MailboxSearchLocationType ([MS-OXWSEDISC] section 3.1.4.4.4.1)	Specifies the mailbox scope.
Conversations	t:ArrayOfConversationRequestsType (section 3.1.4.2.3.2)	Specifies an array of conversation requests.

3.1.4.2.3.6 GetConversationItemsResponseMessageType

The **GetConversationItemsResponseMessageType** complex type specifies the results of an attempt to return the contents of a conversation. The **GetConversationItemsResponseMessageType** complex type extends the **ResponseMessageType** complex type, as specified in [\[MS-OXWSCDATA\]](#) section 2.2.4.65.

```

<xs:complexType name="GetConversationItemsResponseMessageType">
  <xs:complexContent>
    <xs:extension
      base="m:ResponseMessageType"
    >
      <xs:sequence>
        <xs:element name="Conversation"
          type="t:ConversationResponseType"
          minOccurs="0"
        />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

The following table lists the child elements of the **GetConversationItemsResponseMessageType** complex type.

Element	Type	Description
Conversation	t:ConversationResponseType (section 3.1.4.2.3.4)	Specifies a conversation that is returned in a response.

3.1.4.2.3.7 GetConversationItemsResponseType

The **GetConversationItemsResponseType** complex type specifies the response for a **GetConversationItems** operation. The **GetConversationItemsResponseType** complex type extends the **BaseResponseMessageType** complex type, as specified in [\[MS-OXWSCDATA\]](#) section 2.2.4.18.

```
<xs:complexType name="GetConversationItemsResponseType">
  <xs:complexContent>
    <xs:extension
      base="m:BaseResponseMessageType"
    />
  </xs:complexContent>
</xs:complexType>
```

3.1.4.2.3.8 ConversationRequestType

The **ConversationRequestType** complex type specifies a conversation to return in a response. [<23>](#)

```
<xs:complexType name="ConversationRequestType">
  <xs:sequence>
    <xs:element name="ConversationId" type="t:ItemIdType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="SyncState" type="xs:base64Binary" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
```

The following table lists the child elements of the **ConversationRequestType** complex type.

Element	Type	Description
ConversationId	t:ItemIdType ([MS-OXWSCORE] section 2.2.4.24)	Specifies the conversation identifier.
SyncState	xs:base64Binary	Specifies the conversation synchronization state.

3.1.4.2.4 Simple Types

The following table lists and describes the XML schema simple type definitions that are specific to the **GetConversationItems** operation

Simple type	Description
t: ConversationNodeSortOrder (section 3.1.4.2.4.1)	Specifies how conversation nodes are ordered in the response.

3.1.4.2.4.1 ConversationNodeSortOrder

The **ConversationNodeSortOrder** simple type specifies how conversation nodes are ordered in the response. [<24>](#)


```

<xs:simpleType name="ConversationNodeSortOrder">
  <xs:restriction base="xs:string">
    <xs:enumeration value="TreeOrderAscending"/>
    <xs:enumeration value="TreeOrderDescending"/>
    <xs:enumeration value="DateOrderAscending"/>
    <xs:enumeration value="DateOrderDescending"/>
  </xs:restriction>
</xs:simpleType>

```

The following table lists the values that are defined by the **ConversationNodeSortOrder** simple type.

Value	Description
TreeOrderAscending	Specifies that the conversation node sort order is ascending based on the position in the node tree.
TreeOrderDescending	Specifies that the conversation node sort order is descending based on the position in the node tree.
DateOrderAscending	Specifies that the conversation node sort order is ascending based on the last date in the node tree.
DateOrderDescending	Specifies that the conversation node sort order is descending based on the last date in the node tree.

3.1.4.3 ApplyConversationAction

The **ApplyConversationAction** operation specifies the WSDL message that is used to apply actions to conversations in a mailbox.

The following is the WSDL port type definition of this operation.

```

<wsdl:operation name="ApplyConversationAction">
  <wsdl:input message="tns:ApplyConversationActionSoapIn"/>
  <wsdl:output message="tns:ApplyConversationActionSoapOut"/>
</wsdl:operation>

```

The following is the WSDL binding specification of this operation.

```

<wsdl:operation name="ApplyConversationAction">
  <soap:operation
  soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/ApplyConversationAction" />
  <wsdl:input>
    <soap:body parts="request" use="literal" />
    <soap:header message="tns:ApplyConversationActionSoapIn" part="RequestVersion"
    use="literal" />
    <soap:header message="tns:ApplyConversationActionSoapIn" part="Impersonation"
    use="literal" />
  </wsdl:input>
  <wsdl:output>
    <soap:body parts="ApplyConversationActionResult" use="literal" />
    <soap:header message="tns:ApplyConversationActionSoapOut" part="ServerVersion"
    use="literal"/>
  </wsdl:output>

```

</wsdl:operation>

3.1.4.3.1 Messages

The following table lists the set of WSDL message definitions that are specific to the **ApplyConversationAction** operation.

Message	Description
ApplyConversationActionSoapIn (section 3.1.4.3.1.1)	Specifies the SOAP message that represents a request to apply actions to items in a specified conversation.
ApplyConversationActionSoapOut (section 3.1.4.3.1.2)	Specifies the SOAP message that represents a response to a request to apply actions to items in a specified conversation.

3.1.4.3.1.1 ApplyConversationActionSoapIn

The **ApplyConversationActionSoapIn** WSDL message specifies the SOAP message that represents a request to apply actions to items in a specified conversation.

The following is the WSDL message specification of the **ApplyConversationActionSoapIn** message.

```
<wsdl:message name="ApplyConversationActionSoapIn">
  <wsdl:part name="request" element="tns:ApplyConversationAction"/>
  <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
  <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
</wsdl:message>
```

The **ApplyConversationActionSoapIn** WSDL message is the input message for the SOAP action <http://schemas.microsoft.com/exchange/services/2006/messages/ApplyConversationAction>.

The parts of the **ApplyConversationActionSoapIn** WSDL message are listed and described in the following table.

Part	Element/type	Description
request	ApplyConversationAction (section 3.1.4.3.2.1)	Specifies the SOAP body of the request to apply conversation actions. This part is required.
Impersonation	ExchangeImpersonation ([MS-OXWSCDATA] section 2.2.3.3)	Specifies a SOAP header that identifies the user whom the client application is impersonating. This part is optional.
RequestVersion	RequestServerVersion ([MS-OXWSCDATA] section 2.2.3.11)	Specifies a SOAP header that identifies the schema version for the ApplyConversationAction operation request. This part is optional.

3.1.4.3.1.2 ApplyConversationActionSoapOut

The **ApplyConversationActionSoapOut** WSDL message specifies the SOAP message that represents a response to a request to apply actions to items in a specified conversation.

```

<wsdl:message name="ApplyConversationActionSoapOut">
  <wsdl:part name="ApplyConversationActionResult"
  element="tns:ApplyConversationActionResponse"/>
  <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>

```

The **ApplyConversationActionSoapOut** WSDL message is the output message for the SOAP action <http://schemas.microsoft.com/exchange/services/2006/messages/ApplyConversationAction>.

The parts of the **ApplyConversationActionSoapOut** WSDL message are listed and described in the following table.

Part	Element/type	Description
ApplyConversationActionResult	ApplyConversationActionResponse (section 3.1.4.2.2.2)	Specifies the SOAP body of the response to an ApplyConversationAction operation request.
ServerVersion	ServerVersionInfo ([MS-OXWSCDATA] section 2.2.3.12)	Specifies the SOAP header that identifies the server version for the response.

3.1.4.3.2 Elements

The following table lists the XML schema element definitions that are specific to the **ApplyConversationAction** operation.

Element	Description
ApplyConversationAction (section 3.1.4.3.2.1)	Specifies a request to apply actions to a conversation in a mailbox.
ApplyConversationActionResponse (section 3.1.4.2.2.2)	Specifies a response that contains the results of an attempt to apply conversation actions to a conversation in a mailbox.

3.1.4.3.2.1 ApplyConversationAction

The **ApplyConversationAction** element specifies a request to apply actions to a conversation in a mailbox.

```

<xs:element name="ApplyConversationAction"
  type="m:ApplyConversationActionType"
/>

```

3.1.4.3.2.2 ApplyConversationActionResponse

The **ApplyConversationActionResponse** element specifies a response that contains the results of an attempt to apply conversation actions to a conversation in a mailbox. This element **MUST** be present.

```

<xs:element name="ApplyConversationActionResponse"

```

```

        type="m:ApplyConversationActionResponseType"
    />

```

3.1.4.3.3 Complex Types

The following table lists the XML schema complex type definitions that are specific to the **ApplyConversationAction** operation.

Complex type	Description
ApplyConversationActionResponseType (section 3.1.4.3.3.1)	Specifies the contents of a response that contains the results of an attempt to apply conversation actions to a conversation in mailbox.
ApplyConversationActionType (section 3.1.4.3.3.2)	Specifies the contents of a request to apply actions to a conversation in a mailbox.
ConversationActionType (section 3.1.4.3.3.3)	Specifies a single action to be applied to a conversation.
NonEmptyArrayOfApplyConversationActionType (section 3.1.4.3.3.4)	Specifies an array of actions to be applied to one or more conversations.

3.1.4.3.3.1 ApplyConversationActionResponseType

The **ApplyConversationActionResponseType** complex type specifies the contents of a response that contains the results of an attempt to apply conversation actions to a conversation in a mailbox. The **ApplyConversationActionResponseType** complex type extends the **BaseResponseMessageType** complex type, as specified in [\[MS-OXWSCDATA\]](#) section 2.2.4.18.

```

<xs:complexType name="ApplyConversationActionResponseType">
  <xs:complexContent>
    <xs:extension
      base="m:BaseResponseMessageType"
    />
  </xs:complexContent>
</xs:complexType>

```

3.1.4.3.3.2 ApplyConversationActionType

The **ApplyConversationActionType** complex type specifies the contents of a request to apply actions to a conversation in a mailbox. The **ApplyConversationActionType** complex type extends the **BaseRequestType** complex type, as specified in [\[MS-OXWSCDATA\]](#) section 2.2.4.17.

```

<xs:complexType name="ApplyConversationActionType">
  <xs:complexContent>
    <xs:extension
      base="m:BaseRequestType"
    >
      <xs:sequence>
        <xs:element name="ConversationActions"
          type="t:NonEmptyArrayOfApplyConversationActionType"
        />
      </xs:sequence>
    </xs:extension>

```

```

</xs:complexContent>
</xs:complexType>

```

The following table lists the child elements of the **ApplyConversationActionType** complex type.

Element	Type	Description
ConversationActions	t:NonEmptyArrayOfApplyConversationActionType (section 3.1.4.3.3.4)	Specifies a collection of conversations and the actions to apply to them. This element MUST be present.

3.1.4.3.3.3 ConversationActionType

The **ConversationActionType** complex type specifies a single action to be applied to a conversation.

```

<xs:complexType name="ConversationActionType">
  <xs:sequence>
    <xs:element name="Action" type="t:ConversationActionTypeType" minOccurs="1"
maxOccurs="1"/>
    <xs:element name="ConversationId" type="t:ItemIdType" minOccurs="1" maxOccurs="1"/>
    <xs:element name="ContextFolderId" type="t:TargetFolderIdType" minOccurs="0"
maxOccurs="1"/>
    <xs:element name="ConversationLastSyncTime" type="xs:dateTime" minOccurs="0"
maxOccurs="1"/>
    <xs:element name="ProcessRightAway" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
    <xs:element name="DestinationFolderId" type="t:TargetFolderIdType" minOccurs="0"
maxOccurs="1"/>
    <xs:element name="Categories" type="t:ArrayOfStringsType" minOccurs="0"
maxOccurs="1"/>
    <xs:element name="EnableAlwaysDelete" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
    <xs:element name="IsRead" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
    <xs:element name="DeleteType" type="t:DisposalType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="RetentionPolicyType" type="t:RetentionType" minOccurs="0"
maxOccurs="1"/>
    <xs:element name="RetentionPolicyTagId" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="Flag" type="t:FlagType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="SuppressReadReceipts" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>

```

The following table lists and describes the child elements of the **ConversationActionType** complex type.

Element	Type	Description
Action	t:ConversationActionTypeType (section 3.1.4.3.4.1)	Specifies the action to perform on the conversation specified by the ConversationId element. This element MUST be present.
ConversationId	t:ItemIdType ([MS-OXWSCORE] section 2.2.4.24)	Specifies the conversation identifier of the targeted conversation for applying an action. This element MUST be present.

Element	Type	Description
ContextFolderId	t:TargetFolderIdType ([MS-OXWSFOLD] section 2.2.4.14)	Specifies the folder that is targeted for actions that use folders. This element MUST be present when copying, deleting, moving, and setting the read state on conversation items in a target folder. This element MUST NOT be present when copying, deleting, moving, and setting the read state on all conversation items in a mailbox.
ConversationLastSyncTime	xs:dateTime ([XMLSCHEMA2])	Specifies the date and time that a conversation was last synchronized. This element MUST be present when trying to delete all items in a conversation that were received up to the specified time. All items received after the specified time MUST NOT be deleted.
ProcessRightAway	xs:boolean ([XMLSCHEMA2])	Specifies whether the response is sent as soon as the action starts processing on the server, or whether the response is sent after the action has completed. This element MUST be present for the response to be sent asynchronous to the requested action.
DestinationFolderId	t:TargetFolderIdType ([MS-OXWSFOLD] section 2.2.4.14)	Specifies the destination folder for copy and move actions. This element MUST be present when the Action element value is set to "AlwaysMove", "Move", or "Copy".
Categories	t:ArrayOfStringsType ([MS-OXWSCDATA] section 2.2.4.13)	Specifies the categories to set on items in a conversation. This element MUST be present when the Action element value is set to "AlwaysCategorize".
EnableAlwaysDelete	xs:boolean ([XMLSCHEMA2])	Specifies a flag that enables deletion for all new items in a conversation. This element MUST be present when the Action element value is set to "AlwaysDelete".
IsRead	xs:boolean ([XMLSCHEMA2])	Specifies a flag that enables setting the read state of items in a conversation. This element MUST be present when the Action element value is set to "SetReadState".
DeleteType	t:DisposalType ([MS-OXWSCDATA] section 2.2.5.9)	Specifies how items in a conversation are deleted. This element MUST be present when the Action element value is set to "AlwaysDelete" or "Delete".
RetentionPolicyType	RetentionType ([MS-OXWSURPT] section 2.2.5.1)	Specifies the retention policy for the item. <25>
RetentionPolicyTagId	xs:string ([XMLSCHEMA2])	Specifies the retention policy tag identifier. <26>
Flag	FlagType ([MS-OXWSCORE])	Specifies a flag on an item. <27>

Element	Type	Description
	section 2.2.4.22)	
SuppressReadReceipts	xs:boolean ([XMLSCHEMA2])	Specifies whether to suppress a read receipt. <28>

3.1.4.3.3.4 NonEmptyArrayOfApplyConversationAction

The **NonEmptyArrayOfApplyConversationActionType** complex type specifies an array of actions to be applied to one or more conversations.

```
<xs:complexType name="NonEmptyArrayOfApplyConversationActionType">
  <xs:sequence>
    <xs:element name="ConversationAction"
      type="t:ConversationActionType"
      maxOccurs="unbounded"
    />
  </xs:sequence>
</xs:complexType>
```

The following table lists and describes the child elements of the **ConversationType** complex type.

Element	Type	Description
ConversationAction	t:ConversationActionType (section 3.1.4.3.3.3)	Specifies a single conversation action. At least one instance of this element MUST be present.

3.1.4.3.4 Simple Types

The following table lists the XML schema simple type definitions that are specific to the **ApplyConversationAction** operation.

Simple type	Description
ConversationActionTypeType (section 3.1.4.3.4.1)	Specifies the action to apply to items in a conversation.

3.1.4.3.4.1 ConversationActionTypeType

The **ConversationActionTypeType** simple type specifies the action to apply to items in a conversation.

```
<xs:simpleType name="ConversationActionTypeType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="AlwaysCategorize"/>
    <xs:enumeration value="AlwaysDelete"/>
    <xs:enumeration value="AlwaysMove"/>
    <xs:enumeration value="Delete"/>
    <xs:enumeration value="Move"/>
    <xs:enumeration value="Copy"/>
    <xs:enumeration value="SetReadState"/>
  </xs:restriction>
</xs:simpleType>
```

```

    <xs:enumeration value="SetRetentionPolicy" />
    <xs:enumeration value="Flag" />
  </xs:restriction>
</xs:simpleType>

```

The following table lists and describes the enumeration values that are defined by the **ConversationActionTypeType** simple type.

Value	Description
AlwaysCategorize	Specifies that items in the conversation are always categorized.
AlwaysDelete	Specifies that items in the conversation are always deleted.
AlwaysMove	Specifies that items in the conversation are always moved.
Delete	Specifies that items in the conversation are to be deleted.
Move	Specifies that items in the conversation are to be moved.
Copy	Specifies that items in the conversation are to be copied.
SetReadState	Specifies that items in the conversation are to have the read state set.
SetRetentionPolicy	Specifies that items in the conversation are to have the retention policy set. <29>
Flag	Specifies a flag on an item. <30>

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

This section provides a simplified example of a request/response sequence for the **FindConversation** operation, as specified in section [3.1.4.1](#).

The following example shows a **FindConversation** request to return a single conversation from the default Inbox folder.

```
<?xml version="1.0" encoding="utf-8" ?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
  xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages">
  <soap:Header>
    <t:RequestServerVersion Version="Exchange2010 SP1" />
  </soap:Header>
  <soap:Body>
    <m:FindConversation>
      <m:IndexedPageItemView BasePoint="Beginning" MaxEntriesReturned="1" Offset="0"/>
      <m:ParentFolderId>
        <t:DistinguishedFolderId Id="inbox"/>
      </m:ParentFolderId>
    </m:FindConversation>
  </soap:Body>
</soap:Envelope>
```

The following example shows a **FindConversation** response to the request to return a single conversation from the default Inbox folder.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <t:ServerVersionInfo MajorVersion="14"
      MinorVersion="1"
      MajorBuildNumber="139"
      MinorBuildNumber="0"
      Version="Exchange2010_SP1"
      xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
      xmlns="http://schemas.microsoft.com/exchange/services/2006/types"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"/>
  </soap:Header>
  <soap:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <FindConversationResponse ResponseClass="Success"
      xmlns="http://schemas.microsoft.com/exchange/services/2006/messages">
      <ResponseCode>NoError</ResponseCode>
      <Conversations>
        <t:Conversation xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types">
          <t:ConversationId
            Id="AAQkAGYzZjZmNjk5LTIyNTQtNDBjZS05OTk0LTM4OGQ5ZDk4NDE5ZQAAGD7JNksE4zm4ATHm5DvYc="/>
          <t:ConversationTopic>Most fascinating</t:ConversationTopic>
          <t:UniqueRecipients>
            <t:String>User1@aujing-dom.extest.microsoft.com</t:String>
            <t:String>User2@aujing-dom.extest.microsoft.com</t:String>
          </t:UniqueRecipients>
          <t:GlobalUniqueRecipients>
            <t:String>User1@aujing-dom.extest.microsoft.com</t:String>
            <t:String>User2@aujing-dom.extest.microsoft.com</t:String>
          </t:GlobalUniqueRecipients>
          <t:UniqueUnreadSenders>
```

```

    <t:String>Microsoft Outlook</t:String>
  </t:UniqueUnreadSenders>
  <t:GlobalUniqueUnreadSenders>
    <t:String>Microsoft Outlook</t:String>
  </t:GlobalUniqueUnreadSenders>
  <t:UniqueSenders>
    <t:String>Microsoft Outlook</t:String>
  </t:UniqueSenders>
  <t:GlobalUniqueSenders>
    <t:String>Microsoft Outlook</t:String>
    <t:String>UserOne</t:String>
  </t:GlobalUniqueSenders>
  <t:LastDeliveryTime>2010-04-03T23:15:49Z</t:LastDeliveryTime>
  <t:GlobalLastDeliveryTime>2010-04-03T23:15:49Z</t:GlobalLastDeliveryTime>
  <t:HasAttachments>>false</t:HasAttachments>
  <t:GlobalHasAttachments>>false</t:GlobalHasAttachments>
  <t:MessageCount>1</t:MessageCount>
  <t:GlobalMessageCount>2</t:GlobalMessageCount>
  <t:UnreadCount>1</t:UnreadCount>
  <t:GlobalUnreadCount>1</t:GlobalUnreadCount>
  <t:Size>6505</t:Size>
  <t:GlobalSize>7814</t:GlobalSize>
  <t:ItemClasses>
    <t:ItemClass>REPORT.IPM.Note.NDR</t:ItemClass>
  </t:ItemClasses>
  <t:GlobalItemClasses>
    <t:ItemClass>REPORT.IPM.Note.NDR</t:ItemClass>
    <t:ItemClass>IPM.Note</t:ItemClass>
  </t:GlobalItemClasses>
  <t:Importance>Normal</t:Importance>
  <t:GlobalImportance>Normal</t:GlobalImportance>
  <t:ItemIds>
    <t:ItemId Id="AAMkAGYzZjidlC+NAAAAY89GAAA=" ChangeKey="CQAAAA==" />
  </t:ItemIds>
  <t:GlobalItemIds>
    <t:ItemId Id="AAMkAGYzZjidlC+NAAAAY89GAAA=" ChangeKey="CQAAAA==" />
    <t:ItemId Id="AAMkAGYzZjidlC+NAAAAY9LrAAA=" ChangeKey="CQAAAA==" />
  </t:GlobalItemIds>
</t:Conversation>
</Conversations>
</FindConversationResponse>
</soap:Body>
</soap:Envelope>

```

PRE

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

Preliminary

6 Appendix A: Full WSDL

The XML files that are listed in the following table are required in order to implement the functionality described in this document.

File name	Description	Section
MS-OXWSCONV.wsdl	Contains the WSDL for the implementation of this protocol.	6
MS-OXWSCONV-messages.xsd	Contains the XML schema message definitions that are used in this protocol.	7.1
MS-OXWSCONV-types.xsd	Contains the XML schema type definitions that are used in this protocol.	7.2

These files have to be placed in a common folder in order for the WSDL to validate and operate. Also, any schema files that are included in or imported into the MS-OXWSCONV-types.xsd or MS-OXWSCONV-messages.xsd schemas have to be placed in the common folder along with the files listed in the table.

This section contains the contents of the MS-OXWSCONV.wsdl file.

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:wsd1="http://schemas.xmlsoap.org/wsdl/"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages">
  <wsdl:types>
    <xs:schema id="messages" elementFormDefault="qualified" version="Exchange2016"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages">
      <xs:include schemaLocation="MS-OXWSCONV-messages.xsd"/>
    </xs:schema>
  </wsdl:types>
  <wsdl:portType name="ExchangeServicePortType">
    <wsdl:operation name="FindConversation">
      <wsdl:input message="tns:FindConversationSoapIn"/>
      <wsdl:output message="tns:FindConversationSoapOut"/>
    </wsdl:operation>
    <wsdl:operation name="GetConversationItems">
      <wsdl:input message="tns:GetConversationItemsSoapIn"/>
      <wsdl:output message="tns:GetConversationItemsSoapOut"/>
    </wsdl:operation>
    <wsdl:operation name="ApplyConversationAction">
      <wsdl:input message="tns:ApplyConversationActionSoapIn"/>
      <wsdl:output message="tns:ApplyConversationActionSoapOut"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="ExchangeServiceBinding" type="tns:ExchangeServicePortType">
    <wsdl:documentation>
      <wsi:Claim conformsTo="http://ws-i.org/profiles/basic/1.0" xmlns:wsi="http://ws-
i.org/schemas/conformanceClaim/" />
    </wsdl:documentation>
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="FindConversation">
      <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/FindConversation"/>
      <wsdl:input>
        <soap:body parts="request" use="literal"/>
        <soap:header message="tns:FindConversationSoapIn" part="RequestVersion"
use="literal"/>
      </wsdl:input>
    </wsdl:operation>
  </wsdl:binding>
</wsdl:definitions>
```

```

        <soap:header message="tns:FindConversationSoapIn" part="Impersonation"
use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="FindConversationResult" use="literal"/>
            <soap:header message="tns:FindConversationSoapOut" part="ServerVersion"
use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="GetConversationItems">
        <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/GetConversationItems
"/>
        <wsdl:input>
            <soap:body parts="request" use="literal"/>
            <soap:header message="tns:GetConversationItemsSoapIn" part="RequestVersion"
use="literal"/>
            <soap:header message="tns:GetConversationItemsSoapIn" part="Impersonation"
use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="GetConversationItemsResult" use="literal"/>
            <soap:header message="tns:GetConversationItemsSoapOut" part="ServerVersion"
use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="ApplyConversationAction">
        <soap:operation
soapAction="http://schemas.microsoft.com/exchange/services/2006/messages/ApplyConversationAct
ion"/>
        <wsdl:input>
            <soap:body parts="request" use="literal"/>
            <soap:header message="tns:ApplyConversationActionSoapIn" part="RequestVersion"
use="literal"/>
            <soap:header message="tns:ApplyConversationActionSoapIn" part="Impersonation"
use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body parts="ApplyConversationActionResult" use="literal"/>
            <soap:header message="tns:ApplyConversationActionSoapOut" part="ServerVersion"
use="literal"/>
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:message name="FindConversationSoapIn">
    <wsdl:part name="request" element="tns:FindConversation"/>
    <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
    <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
</wsdl:message>
<wsdl:message name="FindConversationSoapOut">
    <wsdl:part name="FindConversationResult" element="tns:FindConversationResponse"/>
    <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
<wsdl:message name="ApplyConversationActionSoapIn">
    <wsdl:part name="request" element="tns:ApplyConversationAction"/>
    <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
    <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
</wsdl:message>
<wsdl:message name="ApplyConversationActionSoapOut">
    <wsdl:part name="ApplyConversationActionResult"
element="tns:ApplyConversationActionResponse"/>
    <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
<wsdl:message name="GetConversationItemsSoapIn">
    <wsdl:part name="request" element="tns:GetConversationItems"/>
    <wsdl:part name="Impersonation" element="t:ExchangeImpersonation"/>
    <wsdl:part name="RequestVersion" element="t:RequestServerVersion"/>
</wsdl:message>

```

```
<wsdl:message name="GetConversationItemsSoapOut">
  <wsdl:part name="GetConversationItemsResult" element="tns:GetConversationItemsResponse"/>
  <wsdl:part name="ServerVersion" element="t:ServerVersionInfo"/>
</wsdl:message>
</wsdl:definitions>
```

Preliminary

7 Appendix B: Full XML Schema

For ease of implementation, the following sections provide the full XML schema for this protocol.

Schema name	Prefix	Section
Messages schema	m:	7.1
Types schema	t:	7.2

These files have to be placed in a common folder in order for the WSDL to validate and operate. Also, any schema files that are included in or imported into the MS-OXWSCONV-types.xsd or MS-OXWSCONV-messages.xsd schemas have to be placed in the common folder along with the files listed in the table.

7.1 Messages Schema

This section provides the contents of the messages schema and information about additional files that the messages schema file requires successful validation. The messages schema is referenced from the WSDL file as MS-OXWSCONV-messages.xsd. The following table lists additional files that have to be present in the folder that contains the messages schema.

File name	Defining specification
MS-OXWSCONV-types.xsd	7.2
MS-OXWSCDATA-messages.xsd	[MS-OXWSCDATA] section 7.1

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="messages" elementFormDefault="qualified" version="Exchange2016"
xmlns:m="http://schemas.microsoft.com/exchange/services/2006/messages"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/messages">
  <xs:import namespace="http://schemas.microsoft.com/exchange/services/2006/types"
schemaLocation="MS-OXWSCONV-types.xsd"/>
  <xs:include schemaLocation="MS-OXWSCDATA-messages.xsd"/>
  <xs:complexType name="FindConversationType">
    <xs:complexContent>
      <xs:extension base="m:BaseRequestType">
        <xs:sequence>
          <xs:choice minOccurs="0">
            <xs:element name="IndexedPageItemView" type="t:IndexedPageItemType"/>
            <xs:element name="SeekToConditionPageItemView"
type="t:SeekToConditionPageItemType"/>
          </xs:choice>
          <xs:element name="SortOrder" type="t:NonEmptyArrayOfFieldOrdersType"
minOccurs="0"/>
          <xs:element name="ParentFolderId" type="t:TargetFolderIdType" minOccurs="1"/>
          <xs:element name="MailboxScope" type="t:MailboxSearchLocationType" minOccurs="0"
maxOccurs="1"/>
          <xs:element name="QueryString" type="m:QueryStringType" minOccurs="0"
maxOccurs="1"/>
          <xs:element name="ConversationShape" type="t:ConversationResponseShapeType"
minOccurs="0" maxOccurs="1"/>
        </xs:sequence>
        <xs:attribute name="Traversal" type="t:ConversationQueryTraversalType"
use="optional"/>
        <xs:attribute name="ViewFilter" type="t:ViewFilterType" use="optional"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
```

```

</xs:complexType>
<xs:element name="FindConversation" type="m:FindConversationType"/>
<xs:complexType name="FindConversationResponseMessageType">
  <xs:complexContent>
    <xs:extension base="m:ResponseMessageType">
      <xs:sequence>
        <xs:element name="Conversations" type="t:ArrayOfConversationsType" minOccurs="0"/>
        <xs:element name="HighlightTerms" type="t:ArrayOfHighlightTermsType"
minOccurs="0"/>
        <xs:element name="TotalConversationsInView" type="xs:int" minOccurs="0"/>
        <xs:element name="IndexedOffset" type="xs:int" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="FindConversationResponse" type="m:FindConversationResponseMessageType"/>
<xs:complexType name="ApplyConversationActionType">
  <xs:complexContent>
    <xs:extension base="m:BaseRequestType">
      <xs:sequence>
        <xs:element name="ConversationActions"
type="t:NonEmptyArrayOfApplyConversationActionType"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="ApplyConversationAction" type="m:ApplyConversationActionType"/>
<xs:complexType name="ApplyConversationActionResponseType">
  <xs:complexContent>
    <xs:extension base="m:BaseResponseMessageType"/>
  </xs:complexContent>
</xs:complexType>
<xs:element name="ApplyConversationActionResponse"
type="m:ApplyConversationActionResponseType"/>
<xs:complexType name="GetConversationItemsType">
  <xs:complexContent>
    <xs:extension base="m:BaseRequestType">
      <xs:sequence>
        <xs:element name="ItemShape" type="t:ItemResponseShapeType" minOccurs="1"
maxOccurs="1"/>
        <xs:element name="FoldersToIgnore" type="t:NonEmptyArrayOfBaseFolderIdsType"
minOccurs="0" maxOccurs="1"/>
        <xs:element name="MaxItemsToReturn" type="xs:int" minOccurs="0" maxOccurs="1"/>
        <xs:element name="SortOrder" type="t:ConversationNodeSortOrder" minOccurs="0"
maxOccurs="1"/>
        <xs:element name="MailboxScope" type="t:MailboxSearchLocationType" minOccurs="0"
maxOccurs="1"/>
        <xs:element name="Conversations" type="t:ArrayOfConversationRequestsType"
minOccurs="1"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:element name="GetConversationItems" type="m:GetConversationItemsType"/>
<xs:complexType name="GetConversationItemsResponseMessageType">
  <xs:complexContent>
    <xs:extension base="m:ResponseMessageType">
      <xs:sequence>
        <xs:element name="Conversation" type="t:ConversationResponseType" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="GetConversationItemsResponseType">
  <xs:complexContent>
    <xs:extension base="m:BaseResponseMessageType"/>
  </xs:complexContent>
</xs:complexType>

```



```
<xs:element name="GetConversationItemsResponse" type="m:GetConversationItemsResponseType"/>
</xs:schema>
```

7.2 Types Schema

This section provides the contents of the types schema and information about additional files that the types schema file requires for successful validation. The types schema is referenced from the messages schema file as MS-OXWSCONV-types.xsd. The following table lists additional files that have to be present in the folder that contains the types schema file.

File name	Defining specification
MS-OXWSCDATA-types.xsd	[MS-OXWSCDATA] section 7.2



```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="types" elementFormDefault="qualified" version="Exchange2016"
xmlns:t="http://schemas.microsoft.com/exchange/services/2006/types"
targetNamespace="http://schemas.microsoft.com/exchange/services/2006/types"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:include schemaLocation="MS-OXWSCDATA-types.xsd"/>
  <xs:include schemaLocation="MS-OXWSEDISC-types.xsd"/>
  <xs:complexType name="ArrayOfItemClassType">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element name="ItemClass" type="t:ItemClassType"/>
    </xs:choice>
  </xs:complexType>
  <xs:simpleType name="ConversationActionTypeType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="AlwaysCategorize"/>
      <xs:enumeration value="AlwaysDelete"/>
      <xs:enumeration value="AlwaysMove"/>
      <xs:enumeration value="Delete"/>
      <xs:enumeration value="Move"/>
      <xs:enumeration value="Copy"/>
      <xs:enumeration value="SetReadState"/>
      <xs:enumeration value="SetRetentionPolicy" />
      <xs:enumeration value="Flag" />
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="ConversationActionType">
    <xs:sequence>
      <xs:element name="Action" type="t:ConversationActionTypeType" minOccurs="1"
maxOccurs="1"/>
      <xs:element name="ConversationId" type="t:ItemIdType" minOccurs="1" maxOccurs="1"/>
      <xs:element name="ContextFolderId" type="t:TargetFolderIdType" minOccurs="0"
maxOccurs="1"/>
      <xs:element name="ConversationLastSyncTime" type="xs:dateTime" minOccurs="0"
maxOccurs="1"/>
      <xs:element name="ProcessRightAway" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
      <xs:element name="DestinationFolderId" type="t:TargetFolderIdType" minOccurs="0"
maxOccurs="1"/>
      <xs:element name="Categories" type="t:ArrayOfStringsType" minOccurs="0" maxOccurs="1"/>
      <xs:element name="EnableAlwaysDelete" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
      <xs:element name="IsRead" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
      <xs:element name="DeleteType" type="t:DisposalType" minOccurs="0" maxOccurs="1"/>
      <xs:element name="RetentionPolicyType" type="t:RetentionType" minOccurs="0"
maxOccurs="1"/>
      <xs:element name="RetentionPolicyTagId" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="Flag" type="t:FlagType" minOccurs="0" maxOccurs="1"/>
      <xs:element name="SuppressReadReceipts" type="xs:boolean" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="NonEmptyArrayOfApplyConversationActionType">
    <xs:sequence>
```

```

    <xs:element name="ConversationAction" type="t:ConversationActionType"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ConversationType">
  <xs:sequence>
    <xs:element name="ConversationId" type="t:ItemIdType" minOccurs="0"/>
    <xs:element name="ConversationTopic" type="xs:string" minOccurs="0"/>
    <xs:element name="UniqueRecipients" type="t:ArrayOfStringsType" minOccurs="0"/>
    <xs:element name="GlobalUniqueRecipients" type="t:ArrayOfStringsType" minOccurs="0"/>
    <xs:element name="UniqueUnreadSenders" type="t:ArrayOfStringsType" minOccurs="0"/>
    <xs:element name="GlobalUniqueUnreadSenders" type="t:ArrayOfStringsType"
minOccurs="0"/>
    <xs:element name="UniqueSenders" type="t:ArrayOfStringsType" minOccurs="0"/>
    <xs:element name="GlobalUniqueSenders" type="t:ArrayOfStringsType" minOccurs="0"/>
    <xs:element name="LastDeliveryTime" type="xs:dateTime" minOccurs="0"/>
    <xs:element name="GlobalLastDeliveryTime" type="xs:dateTime" minOccurs="0"/>
    <xs:element name="Categories" type="t:ArrayOfStringsType" minOccurs="0"/>
    <xs:element name="GlobalCategories" type="t:ArrayOfStringsType" minOccurs="0"/>
    <xs:element name="FlagStatus" type="t:FlagStatusType" minOccurs="0"/>
    <xs:element name="GlobalFlagStatus" type="t:FlagStatusType" minOccurs="0"/>
    <xs:element name="HasAttachments" type="xs:boolean" minOccurs="0"/>
    <xs:element name="GlobalHasAttachments" type="xs:boolean" minOccurs="0"/>
    <xs:element name="MessageCount" type="xs:int" minOccurs="0"/>
    <xs:element name="GlobalMessageCount" type="xs:int" minOccurs="0"/>
    <xs:element name="UnreadCount" type="xs:int" minOccurs="0"/>
    <xs:element name="GlobalUnreadCount" type="xs:int" minOccurs="0"/>
    <xs:element name="Size" type="xs:int" minOccurs="0"/>
    <xs:element name="GlobalSize" type="xs:int" minOccurs="0"/>
    <xs:element name="ItemClasses" type="t:ArrayOfItemClassType" minOccurs="0"/>
    <xs:element name="GlobalItemClasses" type="t:ArrayOfItemClassType" minOccurs="0"/>
    <xs:element name="Importance" type="t:ImportanceChoicesType" minOccurs="0"/>
    <xs:element name="GlobalImportance" type="t:ImportanceChoicesType" minOccurs="0"/>
    <xs:element name="ItemIds" type="t:NonEmptyArrayOfBaseItemIdsType" minOccurs="0"/>
    <xs:element name="GlobalItemIds" type="t:NonEmptyArrayOfBaseItemIdsType"
minOccurs="0"/>
    <xs:element name="LastModifiedTime" type="xs:dateTime" minOccurs="0"/>
    <xs:element name="InstanceKey" type="xs:base64Binary" minOccurs="0"/>
    <!-- Maps to ConversationItemSchema.ConversationPreview-->

```

```

<xs:element name="Preview" type="xs:string" minOccurs="0"/>

```

```

    <!-- Flag for the mailbox scope this conversation was found in -->
    <xs:element name="MailboxScope" type="t:MailboxSearchLocationType" minOccurs="0"/>
    <!-- Maps to ConversationItemSchema.ConversationIconIndex-->
    <xs:element name="IconIndex" type="t:IconIndexType" minOccurs="0"/>
    <!-- Maps to ConversationItemSchema.ConversationGlobalIconIndex-->
    <xs:element name="GlobalIconIndex" type="t:IconIndexType" minOccurs="0"/>
    <!-- Maps to ConversationItemSchema.ConversationDraftItemIds-->
    <xs:element name="DraftItemIds" type="t:NonEmptyArrayOfBaseItemIdsType" minOccurs="0"/>
    <!-- Maps to ConversationItemSchema.ConversationHasIrm-->
    <xs:element name="HasIrm" type="xs:boolean" minOccurs="0"/>
    <!-- Maps to ConversationItemSchema.ConversationGlobalHasIrm-->
    <xs:element name="GlobalHasIrm" type="xs:boolean" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ArrayOfConversationsType">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="Conversation" type="t:ConversationType"/>
  </xs:choice>
</xs:complexType>
<xs:simpleType name="ConversationQueryTraversalType">
  <xs:annotation>
    <xs:documentation>Types of sub-tree traversal for conversations</xs:documentation>

```

```

    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Shallow"/>
      <xs:enumeration value="Deep"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="ConversationResponseShapeType">
    <xs:sequence>
      <xs:element name="BaseShape" type="t:DefaultShapeNamesType"/>
      <xs:element name="AdditionalProperties" type="t:NonEmptyArrayOfPathsToElementType"
minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ConversationRequestType">
    <xs:sequence>
      <xs:element name="ConversationId" type="t:ItemIdType" minOccurs="1" maxOccurs="1"/>
      <xs:element name="SyncState" type="xs:base64Binary" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ArrayOfConversationRequestsType">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element name="Conversation" type="t:ConversationRequestType"/>
    </xs:choice>
  </xs:complexType>
  <xs:complexType name="ConversationNodeType">
    <xs:sequence>
      <xs:element name="InternetMessageId" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="ParentInternetMessageId" type="xs:string" minOccurs="0"
maxOccurs="1"/>
      <xs:element name="Items" type="t:NonEmptyArrayOfAllItemsType" minOccurs="0"
maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ArrayOfConversationNodesType">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element name="ConversationNode" type="t:ConversationNodeType"/>
    </xs:choice>
  </xs:complexType>
  <xs:complexType name="ConversationResponseType">
    <xs:sequence>
      <xs:element name="ConversationId" type="t:ItemIdType" minOccurs="1" maxOccurs="1"/>
      <xs:element name="SyncState" type="xs:base64Binary" minOccurs="0" maxOccurs="1"/>
      <xs:element name="ConversationNodes" type="t:ArrayOfConversationNodesType"
minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="ConversationNodeSortOrder">
    <xs:restriction base="xs:string">
      <xs:enumeration value="TreeOrderAscending"/>
      <xs:enumeration value="TreeOrderDescending"/>
      <xs:enumeration value="DateOrderAscending"/>
      <xs:enumeration value="DateOrderDescending"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="ViewFilterType">
    <xs:annotation>
      <xs:documentation>Types of view filters for finding
items/conversations</xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:string">
      <xs:enumeration value="All"/>
      <xs:enumeration value="Flagged"/>
      <xs:enumeration value="HasAttachment"/>
      <xs:enumeration value="ToOrCcMe"/>
      <xs:enumeration value="Unread"/>
      <xs:enumeration value="TaskActive"/>
      <xs:enumeration value="TaskOverdue"/>
      <xs:enumeration value="TaskCompleted"/>
    </xs:restriction>
  </xs:simpleType>

```

```
<xs:enumeration value="NoClutter"/>

<xs:enumeration value="Clutter"/>

    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

Preliminary

8 Appendix C: 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 Exchange Server 2010 Service Pack 1 (SP1)
- Microsoft Exchange Server 2013
- Microsoft Exchange Server 2016 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 3.1.4.1.3.3](#): Exchange 2010 SP1 does not use the **ConversationResponseShapeType** complex type.

<2> [Section 3.1.4.1.3.4](#): Exchange 2010 SP1 does not use the **LastModifiedTime** element.

<3> [Section 3.1.4.1.3.4](#): Exchange 2010 SP1 does not use the **InstanceKey** element.

<4> [Section 3.1.4.1.3.4](#): Exchange 2010 SP1 does not use the **MailboxScope** element.

<5> [Section 3.1.4.1.3.4](#): Exchange 2010 SP1 does not use the **IconIndex** element.

<6> [Section 3.1.4.1.3.4](#): Exchange 2010 SP1 does not use the **GlobalIconIndex** element.

<7> [Section 3.1.4.1.3.4](#): Exchange 2010 SP1 does not use the **DraftItemIds** element.

<8> [Section 3.1.4.1.3.4](#): Microsoft Exchange Server 2007, Microsoft Exchange Server 2010, and the initial release of Exchange 2013 do not support the **HasIrm** element. This element was introduced in Microsoft Exchange Server 2013 Service Pack 1 (SP1),

<9> [Section 3.1.4.1.3.4](#): Exchange 2007, Exchange 2010, and the initial release of Exchange 2013 do not support the **GlobalHasIrm** element. This element was introduced in Exchange 2013 SP1,

<10> [Section 3.1.4.1.3.5](#): Exchange 2010 SP1 does not use the **SeekToConditionPageItemView**, **MailboxScope**, **QueryString**, **ConversationShape**, **Traversal**, or **ViewFilter** elements.

<11> [Section 3.1.4.1.4.1](#): Exchange 2010 SP1 does not use the **ConversationQueryTraversalType** simple type. The **ConversationQueryTraversalType** simple type was introduced in Exchange 2013.

<12> [Section 3.1.4.1.4.3](#): Exchange 2010 SP1 does not use the **ViewFilterType** simple type. The **ViewFilterType** simple type was introduced in Exchange 2013.

<13> [Section 3.1.4.1.4.3](#): Exchange 2013 SP1 and Exchange 2016 Preview do not use the **Suggestions** value.

<14> [Section 3.1.4.1.4.3](#): Exchange 2013 SP1 and Exchange 2016 Preview do not use the **SuggestionsRespond** value.

<15> [Section 3.1.4.1.4.3](#): Exchange 2013 SP1 and Exchange 2016 Preview do not use the **SuggestionsDelete** value.

<16> [Section 3.1.4.1.4.3](#): Exchange 2007, Exchange 2010, and the initial release of Exchange 2013 do not use the **NoClutter** value. This value was introduced in Exchange 2013 SP1.

<17> [Section 3.1.4.1.4.3](#): Exchange 2007, Exchange 2010, and the initial release of Exchange 2013 do not use the **Clutter** value. This value was introduced in Exchange 2013 SP1.

<18> [Section 3.1.4.2](#): Exchange 2010 SP1 does not use the **GetConversationItems** operation.

<19> [Section 3.1.4.2.3.1](#): Exchange 2010 SP1 does not use the **ArrayOfConversationNodesType** complex type.

<20> [Section 3.1.4.2.3.2](#): Exchange 2010 SP1 does not use the **ArrayOfConversationRequestsType** complex type.

<21> [Section 3.1.4.2.3.3](#): Exchange 2010 SP1 does not use the **ConversationNodeType** complex type.

<22> [Section 3.1.4.2.3.4](#): Exchange 2010 SP1 does not use the **ConversationResponseType** complex type.

<23> [Section 3.1.4.2.3.8](#): Exchange 2010 SP1 does not use the **ConversationRequestType** complex type.

<24> [Section 3.1.4.2.4.1](#): Exchange 2010 SP1 does not use the **ConversationNodeSortOrder** simple type.

<25> [Section 3.1.4.3.3.3](#): Exchange 2010 SP1 does not support the **RetentionPolicyType** element.

<26> [Section 3.1.4.3.3.3](#): Exchange 2010 SP1 does not support the **RetentionPolicyTagId** element.

<27> [Section 3.1.4.3.3.3](#): Exchange 2010 SP1 does not support the **Flag** element.

<28> [Section 3.1.4.3.3.3](#): Exchange 2010 SP1 does not support the **SuppressReadReceipt** element.

<29> [Section 3.1.4.3.4.1](#): Exchange 2010 SP1 does not support the **SetRetentionPolicy** value.

<30> [Section 3.1.4.3.4.1](#): Exchange 2010 SP1 does not support the **Flag** value.

9 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

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

- A document revision that incorporates changes to interoperability requirements or functionality.
- The removal of a document from the documentation set.

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

The revision class **Editorial** means that the formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the technical content of the document is identical to the last released version.

Major and minor changes can be described further using the following change types:

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.
- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated**.

Some important terms used in the change type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
3.1.4.1.3.4 ConversationType	Added new elements HasIrm and GlobalHasIrm.	Y	Content update.
3.1.4.1.3.4 ConversationType	Specified that elements HasIrm and GlobalHasIrm were introduced in Exchange 2013 SP1.	Y	New product behavior note added.
3.1.4.1.4.3 ViewFilterType	Added values NoClutter and Clutter. Specified that Suggestions, SuggestionsRespond, and SuggestionsDelete are no longer used.	Y	Content update.
3.1.4.1.4.3 ViewFilterType	Specified that values NoClutter and Clutter were introduced in Exchange 2013 SP1.	Y	New product behavior note added.
8 Appendix C: Product Behavior	Added Exchange 2016 to list of applicable products.	Y	Content update.

10 Index

A

Abstract data model
[server](#) 12
[Applicability](#) 9
[Attribute groups](#) 11
[Attributes](#) 11

C

[Capability negotiation](#) 9
[Change tracking](#) 55
[Complex types](#) 11

D

Data model - abstract
[server](#) 12

E

Events
[local - server](#) 40
[timer - server](#) 40
Examples
[finding a conversation](#) 41

F

[Fields - vendor-extensible](#) 9
[Finding a conversation example](#) 41
[Full WSDL](#) 44
[Full XML schema](#) 47
[Messages Schema](#) 47
[Types Schema](#) 49

G

[Glossary](#) 6
[Groups](#) 11

I

[Implementer - security considerations](#) 43
[Index of security parameters](#) 43
[Informative references](#) 8
Initialization
[server](#) 12
[Introduction](#) 6

L

Local events
[server](#) 40

M

Message processing
[server](#) 12

Messages

[attribute groups](#) 11
[attributes](#) 11
[complex types](#) 11
[elements](#) 11
[enumerated](#) 10
[groups](#) 11
[namespaces](#) 10
[simple types](#) 11
[syntax](#) 10
[transport](#) 10

N

[Namespaces](#) 10
[Normative references](#) 7

O

Operations
[ApplyConversationAction](#) 33
[FindConversation](#) 12
[GetConversationItems](#) 25
[Overview \(synopsis\)](#) 8

P

[Parameters - security index](#) 43
[Preconditions](#) 9
[Prerequisites](#) 9
[Product behavior](#) 53
Protocol Details
[overview](#) 12

R

[References](#) 7
[informative](#) 8
[normative](#) 7
[Relationship to other protocols](#) 8

S

Security
[implementer considerations](#) 43
[parameter index](#) 43
Sequencing rules
[server](#) 12
Server
[abstract data model](#) 12
[ApplyConversationAction operation](#) 33
[FindConversation operation](#) 12
[GetConversationItems operation](#) 25
[initialization](#) 12
[local events](#) 40
[message processing](#) 12
[sequencing rules](#) 12
[timer events](#) 40
[timers](#) 12
[Simple types](#) 11

[Standards assignments](#) 9
Syntax
[messages - overview](#) 10

T

Timer events
[server](#) 40
Timers
[server](#) 12
[Tracking changes](#) 55
[Transport](#) 10
Types
[complex](#) 11
[simple](#) 11

V

[Vendor-extensible fields](#) 9
[Versioning](#) 9

W

[WSDL](#) 44

X

[XML schema](#) 47
[Messages Schema](#) 47
[Types Schema](#) 49

Preliminary