

[MS-ASCON]:

Exchange ActiveSync: Conversations 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.

Revision Summary

Date	Revision History	Revision Class	Comments
4/10/2009	0.1.0	Major	Initial Availability.
7/15/2009	1.0.0	Major	Revised and edited for technical content.
11/4/2009	2.0.0	Major	Updated and revised the technical content.
2/10/2010	2.1.0	Minor	Updated the technical content.
5/5/2010	3.0.0	Major	Updated and revised the technical content.
8/4/2010	4.0	Major	Significantly changed the technical content.
11/3/2010	4.0	No change	No changes to the meaning, language, or formatting of the technical content.
3/18/2011	4.1	Minor	Clarified the meaning of the technical content.
8/5/2011	5.0	Major	Significantly changed the technical content.
10/7/2011	5.0	No Change	No changes to the meaning, language, or formatting of the technical content.
1/20/2012	6.0	Major	Significantly changed the technical content.
4/27/2012	6.0	No Change	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	7.0	Major	Significantly changed the technical content.
10/8/2012	8.0	Major	Significantly changed the technical content.
2/11/2013	8.0	No Change	No changes to the meaning, language, or formatting of the technical content.
7/26/2013	9.0	Major	Significantly changed the technical content.
11/18/2013	9.0	No Change	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	9.0	No Change	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	10.0	Major	Significantly changed the technical content.
7/31/2014	10.0	No Change	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	10.1	Minor	Clarified the meaning of the technical content.
5/26/2015	11.0	Major	Significantly changed the technical content.
6/30/2015	12.0	Major	Significantly changed the technical content.
9/14/2015	13.0	Major	Significantly changed the technical content.

Table of Contents

1	Introduction	5
1.1	Glossary	5
1.2	References	6
1.2.1	Normative References	6
1.2.2	Informative References	6
1.3	Overview	6
1.4	Relationship to Other Protocols	7
1.5	Prerequisites/Preconditions	7
1.6	Applicability Statement	7
1.7	Versioning and Capability Negotiation	7
1.8	Vendor-Extensible Fields	7
1.9	Standards Assignments	7
2	Messages	8
2.1	Transport	8
2.2	Message Syntax	8
2.2.1	Namespaces	8
2.2.2	Elements	8
2.2.2.1	BodyPart	9
2.2.2.2	BodyPartPreference	10
2.2.2.3	ConversationId	10
2.2.2.3.1	ConversationId (ItemOperations)	11
2.2.2.3.2	ConversationId (Search)	11
2.2.2.3.3	ConversationId (Sync)	11
2.2.2.4	ConversationIndex	11
2.2.2.4.1	Conversation Index Header	12
2.2.2.5	ConversationMode	12
2.2.2.6	DstFldId	13
2.2.2.7	Move	13
2.2.2.8	MoveAlways	13
2.2.2.9	Options	14
2.2.2.10	Status	14
3	Protocol Details	15
3.1	Client Details	15
3.1.1	Abstract Data Model	15
3.1.2	Timers	15
3.1.3	Initialization	15
3.1.4	Higher-Layer Triggered Events	15
3.1.4.1	Deleting a Conversation	15
3.1.4.2	Flagging a Conversation for Follow-up	15
3.1.4.3	Marking a Conversation as Read or Unread	16
3.1.4.4	Ignoring a Conversation	16
3.1.4.5	Moving a Conversation from the Current Folder	16
3.1.4.6	Setting up a Conversation to Be Moved Always	16
3.1.4.7	Finding a Conversation	17
3.1.4.8	Synchronizing a Conversation	17
3.1.4.9	Applying a Conversation-based Filter	17
3.1.4.10	Requesting a Message Part	17
3.1.5	Message Processing Events and Sequencing Rules	17
3.1.6	Timer Events	18
3.1.7	Other Local Events	18
3.2	Server Details	18
3.2.1	Abstract Data Model	18
3.2.2	Timers	18

3.2.3	Initialization	18
3.2.4	Higher-Layer Triggered Events	18
3.2.5	Message Processing Events and Sequencing Rules	18
3.2.5.1	Processing a Sync Command	18
3.2.5.1.1	Deleting a Conversation	18
3.2.5.1.2	Marking a Conversation as Read or Unread	19
3.2.5.1.3	Flagging a Conversation for Follow-up	19
3.2.5.2	Processing a GetItemEstimate Command	19
3.2.5.3	Processing an ItemOperations Command	19
3.2.5.3.1	Ignoring a Conversation	19
3.2.5.3.2	Always Moving a Conversation	19
3.2.5.4	Processing a MoveItems Command	20
3.2.5.5	Processing a Search Command	20
3.2.5.6	Filtering	20
3.2.5.7	Sending a Message Part	20
3.2.6	Timer Events.....	21
3.2.7	Other Local Events.....	21
4	Protocol Examples	22
4.1	Synchronizing Conversation Data	22
4.2	Ignoring a Conversation.....	22
5	Security	24
5.1	Security Considerations for Implementers	24
5.2	Index of Security Parameters	24
6	Appendix A: Product Behavior	25
7	Change Tracking.....	26
8	Index.....	28

1 Introduction

The Exchange ActiveSync: Conversations Protocol enables the communication of data between a mobile device and the server in the ActiveSync protocol for e-mail messages that are part of a **conversation**. It describes an **XML**-based format that is used to improve the ways in which e-mail messages are triaged when they are displayed in conversation view.

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.

conversation ID: A unique value that is associated with a conversation. It is assigned to each Message object that is part of a conversation and it is used to identify the conversation to which the message belongs.

conversation index: A value that specifies the location of a message within a conversation. A client can use this value to identify the parent and child messages of a message, and then generate a tree view of the conversation that contains those messages.

Deleted Items folder: A special folder that is the default location for objects that have been deleted.

Drafts folder: A special folder that is the default location for Message objects that have been saved but not sent.

Hypertext Markup Language (HTML): An application of the Standard Generalized Markup Language (SGML) that uses tags to mark elements in a document, as described in [\[HTML\]](#).

message part: A message body (2) with a string property that contains only the portion of an email message that is original to the message. It does not include any previous, quoted messages. If a message does not quote a previous message, the message part is identical to the message body.

Outbox folder: A special folder that contains Message objects that are submitted to be sent.

recipient information cache: An information store that contains a list of the contacts (3) with whom a user has interacted most often and most recently, and with whom the user is likely to interact again.

Wireless Application Protocol (WAP) Binary XML (WBXML): A compact binary representation of **XML** that is designed to reduce the transmission size of XML documents over narrowband communication channels.

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

XML element: An **XML** structure that typically consists of a start tag, an end tag, and the information between those tags. Elements can have attributes (1) and can contain other elements.

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-ASAIRS] Microsoft Corporation, "[Exchange ActiveSync: AirSyncBase Namespace Protocol](#)".

[MS-ASCMD] Microsoft Corporation, "[Exchange ActiveSync: Command Reference Protocol](#)".

[MS-ASDTYPE] Microsoft Corporation, "[Exchange ActiveSync: Data Types](#)".

[MS-ASEMAIL] Microsoft Corporation, "[Exchange ActiveSync: Email Class Protocol](#)".

[MS-ASWBXML] Microsoft Corporation, "[Exchange ActiveSync: WAP Binary XML \(WBXML\) Algorithm](#)".

[MS-OXOMSG] Microsoft Corporation, "[Email Object Protocol](#)".

[MS-OXOSFLD] Microsoft Corporation, "[Special Folders 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>

[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/>

[XML] World Wide Web Consortium, "Extensible Markup Language (XML) 1.0 (Fourth Edition)", W3C Recommendation 16 August 2006, edited in place 29 September 2006, <http://www.w3.org/TR/2006/REC-xml-20060816/>

1.2.2 Informative References

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

1.3 Overview

This protocol describes an XML-based format that is used to improve the ways in which e-mail messages are triaged, allowing a user to view a series of send/response e-mail messages as a single representation, called a conversation.

A conversation appears in a message folder as one unit and allows the user to read the series of related e-mail messages in a single effort. Each e-mail message is assigned a **conversation ID** that is used to identify the conversation to which the e-mail message belongs.

1.4 Relationship to Other Protocols

This protocol consists of a series of **XML elements** that are used by the command requests and command responses that are described in [\[MS-ASCMD\]](#).

The **Wireless Application Protocol (WAP) Binary XML (WBXML)**, described in [\[MS-ASWBXML\]](#), is used to transmit the XML markup that constitutes the request body or the response body that is transmitted between the client and the server.

This protocol defines elements according to the data type definitions that are described in [\[MS-ASDTYPE\]](#). Common **XML schema** elements that are used by other classes are defined in [\[MS-ASAIRS\]](#).

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

1.5 Prerequisites/Preconditions

None.

1.6 Applicability Statement

This protocol is applicable when a mobile device is synchronizing its e-mail messages and files with a server by using the commands described in [\[MS-ASCMD\]](#), and wants to present e-mail messages in a view that groups e-mail messages by conversation instead of listing e-mail messages serially.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

This protocol consists of a series of XML elements that are embedded inside a command request or a command response, as specified in [\[MS-ASCMD\]](#).

The XML markup that constitutes the request body or the response body that is transmitted between the client and the server uses Wireless Application Protocol (WAP) Binary XML (WBXML), as specified in [\[MS-ASWBXML\]](#).

2.2 Message Syntax

The XML schemas for the **Email2** and **AirSyncBase** namespaces are described in [\[MS-AEMAIL\]](#) section 6.2 and [\[MS-ASAIRS\]](#) section 6, respectively.

The schema for the **AirSync** namespace is described in [\[MS-ASCMD\]](#) section 6.1. Schemas for the **ItemOperations** namespace are described in [\[MS-ASCMD\]](#) sections 6.19, 6.20, and 6.21 and for the **Search** namespace are described in [\[MS-ASCMD\]](#) sections 6.30, 6.31, and 6.32.

The XML markup that is used by this protocol MUST be well-formed XML, as specified in [\[XML\]](#).

2.2.1 Namespaces

This specification defines and references various **XML namespaces** 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
airsync	AirSync	[MS-ASCMD] section 2.2.2.20
airsyncbase	AirSyncBase	[MS-ASAIRS]
email	Email	[MS-AEMAIL]
email2	Email2	[MS-AEMAIL]
getitemestimate	GetItemEstimate	[MS-ASCMD] section 2.2.2.8
itemoperations	ItemOperations	[MS-ASCMD] section 2.2.2.9
search	Search	[MS-ASCMD] section 2.2.2.15

2.2.2 Elements

Elements for conversations are defined in five namespaces: **Email2**, **AirSyncBase**, **AirSync**, **ItemOperations**, and **Search**. The namespace in which an element is defined is indicated by the presence of a namespace prefix, as defined in section [2.2.1](#).

The following table summarizes the set of common XML schema elements that are defined by this specification. Elements that are specific to a particular operation are specified further in sections [3.1.4](#), [3.1.5](#), [3.2.4](#), and [3.2.5](#).

Element name	Description
airsyncbase:BodyPart (section 2.2.2.1)	Used in ItemOperations ([MS-ASCMD] section 2.2.2.9), Search ([MS-ASCMD] section 2.2.2.15), and Sync ([MS-ASCMD] section 2.2.2.20) command responses to encapsulate a message part and its meta-data.
airsyncbase:BodyPartPreference (section 2.2.2.2)	Used in ItemOperations , Search , and Sync command requests to specify preferences for receiving a message part from the server.
itemoperations:ConversationId (section 2.2.2.3.1)	Used in an ItemOperations command request to specify the conversation ID of the conversation that is to be moved. Used in an ItemOperations command response to specify the conversation ID of the conversation that was moved.
search:ConversationId (section 2.2.2.3.2)	Used in a Search command request to specify the conversation ID of the conversation for which to search.
email2:ConversationId (section 2.2.2.3.3)	Used in a Sync command response to specify the conversation ID for an e-mail message.
email2:ConversationIndex (section 2.2.2.4)	Used in a Sync command response to specify the conversation index for an e-mail message.
airsync:ConversationMode (section 2.2.2.5)	Used in a GetItemEstimate command ([MS-ASCMD] section 2.2.2.8) request to enable conversation-based filtering of item estimates. Used in a Sync command request to enable conversation-based filtering and synchronization of conversation-based properties.
itemoperations:DstFldId (section 2.2.2.6)	Used in an ItemOperations command request to specify the destination folder, which is the folder to which the conversation is moved.
itemoperations:Move (section 2.2.2.7)	Used in an ItemOperations command request to indicate that a conversation is to be moved. Used in an ItemOperations command response to specify the results of the attempt to move a conversation.
itemoperations:MoveAlways (section 2.2.2.8)	Used in an ItemOperations command request to specify that the conversation always be moved.
itemoperations:Options (section 2.2.2.9)	Used in an ItemOperations command request to specify the options for moving a conversation.
itemoperations:Status (section 2.2.2.10)	Used in an ItemOperations command response to specify the status of the move action.

Protocol Versions

The functionality of conversations is supported only by protocol versions 16.0, 14.1, and 14.0. Therefore, the elements that are defined by this specification MUST be used with protocol version 16.0, 14.1, or 14.0 for any operation involving conversations.

The **airsyncbase:BodyPart** element, as specified in section 2.2.2.1, and the **airsyncbase:BodyPartPreference** element, as specified in section 2.2.2.2, are supported only by protocol versions 16.0 and 14.1.

2.2.2.1 BodyPart

The **airsyncbase:BodyPart** element ([\[MS-ASAIRS\]](#) section 2.2.2.10) encapsulates a message part and its meta-data in a **Sync** command response ([\[MS-ASCMD\]](#) section 2.2.2.20), an **ItemOperations** command response ([\[MS-ASCMD\]](#) section 2.2.2.9), or a **Search** command response ([\[MS-ASCMD\]](#)

section 2.2.2.15). The **airsyncbase:BodyPart** element is an optional child element of the **airsync:ApplicationData** element ([MS-ASCMD] section 2.2.3.11) in a **Sync** command response, an optional child element of the **itemoperations:Properties** element ([MS-ASCMD] section 2.2.3.131.1) in an **ItemOperations** command response, and an optional child element of the **search:Properties** element ([MS-ASCMD] section 2.2.3.131.2) in a **Search** command response.

The **airsyncbase:BodyPart** element is defined in the **AirSyncBase** namespace. For more details about the **AirSyncBase** namespace, see [MS-ASAIRS].

The **airsyncbase:BodyPart** element is a **container** ([MS-ASDTYPE] section 2.2) element. It has the following child elements:

- **airsyncbase:Status** ([MS-ASAIRS] section 2.2.2.37)
- **airsyncbase:Type** ([MS-ASAIRS] section 2.2.2.41.2)
- **airsyncbase:EstimatedDataSize** ([MS-ASAIRS] section 2.2.2.23.3)
- **airsyncbase:Truncated** ([MS-ASAIRS] section 2.2.2.39.2)
- **airsyncbase:Data** ([MS-ASAIRS] section 2.2.2.20.2)
- **airsyncbase:Preview** ([MS-ASAIRS] section 2.2.2.35.2)

The **airsyncbase:BodyPart** element and its child elements are further specified in [MS-ASAIRS].

2.2.2.2 BodyPartPreference

The **airsyncbase:BodyPartPreference** element ([MS-ASAIRS] section 2.2.2.11) specifies the client's preferences for receiving a message part from the server. The **airsyncbase:BodyPartPreference** element is an optional child element of the **itemoperations:Options** element ([MS-ASCMD] section 2.2.3.118.2) in an **ItemOperations** command request ([MS-ASCMD] section 2.2.2.9), an optional child element of the **search:Options** element ([MS-ASCMD] section 2.2.3.118.4) in a **Search** command request ([MS-ASCMD] section 2.2.2.15), and an optional child element of the **airsync:Options** element ([MS-ASCMD] section 2.2.3.118.5) in a **Sync** command request ([MS-ASCMD] section 2.2.2.20).

The **airsyncbase:BodyPart** element is defined in the **AirSyncBase** namespace. For more details about the **AirSyncBase** namespace, see [MS-ASAIRS].

The **airsyncbase:BodyPartPreference** element is a **container** ([MS-ASDTYPE] section 2.2) element. It has the following child elements:

- **airsyncbase:Type** ([MS-ASAIRS] section 2.2.2.41.3)
- **airsyncbase:TruncationSize** ([MS-ASAIRS] section 2.2.2.40.1)
- **airsyncbase:AllOrNone** ([MS-ASAIRS] section 2.2.2.3.1)
- **airsyncbase:Preview** ([MS-ASAIRS] section 2.2.2.35.3)

The **airsyncbase:BodyPartPreference** element and its child elements are further specified in [MS-ASAIRS].

2.2.2.3 ConversationId

The **ConversationId** element is used in the following command requests and responses:

- **ItemOperations** command requests and responses ([MS-ASCMD] section 2.2.2.9)

- **Search** command requests ([MS-ASCMD] section 2.2.2.15)
- **Sync** command responses ([MS-ASCMD] section 2.2.2.20)

2.2.2.3.1 ConversationId (ItemOperations)

In an **ItemOperations** command request ([MS-ASCMD] section 2.2.2.9), the **itemoperations:ConversationId** element ([MS-ASCMD] section 2.2.3.35.1) is a required child element of the **itemoperations:Move** element ([MS-ASCMD] section 2.2.3.110.1) that specifies the conversation ID of the conversation that is to be moved.

In an **ItemOperations** command response, the **itemoperations:ConversationId** is a required child element of the **itemoperations:Move** element that specifies the conversation ID of the conversation that was moved. The **itemoperations:ConversationId** element is defined in the **ItemOperations** namespace.

The value of this element is a **byte array**, as specified in [MS-ASDTYPE] section 2.7.1.

2.2.2.3.2 ConversationId (Search)

The **search:ConversationId** element ([MS-ASCMD] section 2.2.3.35.2) is an optional child element of the **search:Query** element ([MS-ASCMD] section 2.2.3.132) in a **Search** command request ([MS-ASCMD] section 2.2.2.15) that specifies the conversation ID of the conversation for which to search.

The **search:ConversationId** element is defined in the **Search** namespace.

The value of this element is a **byte array**, as specified in [MS-ASDTYPE] section 2.7.1.

2.2.2.3.3 ConversationId (Sync)

The **email2:ConversationId** element ([MS-ASEMAIL] section 2.2.2.21) is a required child element of the **airsync:ApplicationData** element ([MS-ASCMD] section 2.2.3.11) in a **Sync** command response ([MS-ASCMD] section 2.2.2.20) that specifies the conversation ID of a conversation.

The **email2:ConversationId** element is defined in the **Email2** namespace.

The value of this element is a **byte array**, as specified in [MS-ASDTYPE] section 2.7.1. The **email2:ConversationId** element is not present if there is no conversation ID associated with the message.

The client MUST NOT change the **email2:ConversationId** value.

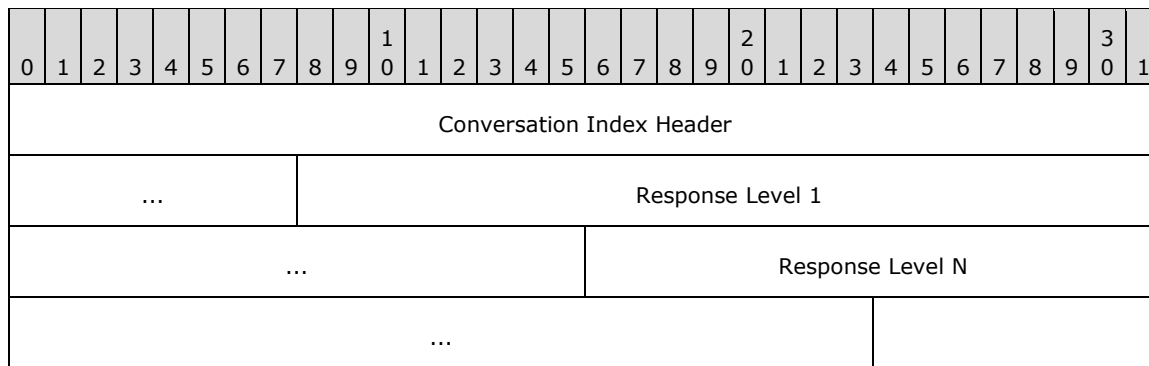
2.2.2.4 ConversationIndex

The **email2:ConversationIndex** element ([MS-ASEMAIL] section 2.2.2.22) is a required child element of the **airsync:ApplicationData** element ([MS-ASCMD] section 2.2.3.11) in a **Sync** command response ([MS-ASCMD] section 2.2.2.20) that specifies the conversation index for an e-mail message.

The **email2:ConversationIndex** element is defined in the **Email2** namespace.

The value of this element is a **byte array**, as specified in [MS-ASDTYPE] section 2.7.1. The value comprises a set of timestamps, which can be used by a client to generate a tree-view of a conversation. The value of the first timestamp is derived from the date and time when the message was originally sent by the server. Each additional timestamp specifies the difference between the current time and the time specified by the first timestamp. Additional timestamps are added when the message is forwarded or replied to.

The **email2:ConversationIndex** value is set according to the description in the following diagram.



Conversation Index Header (5 bytes): A **Conversation Index Header** that is derived from the date and time when the message was originally sent by the server, as specified in section [2.2.2.4.1](#).

Response Level 1 (5 bytes): A **Response Level** that contains information about the time the message was forwarded or replied to. This field corresponds to the **Response Levels** field of the **PidTagConversationIndex** property as specified in [\[MS-OXOMSG\]](#) section 2.2.1.3. Additional **Response Level** fields are added to the **email2:ConversationIndex** each time the message is forwarded or replied to.

Response Level N (5 bytes): Additional **Response Level** fields for each time the message is forwarded or replied to. This field is set according to the description for the **Response Level 1** field.

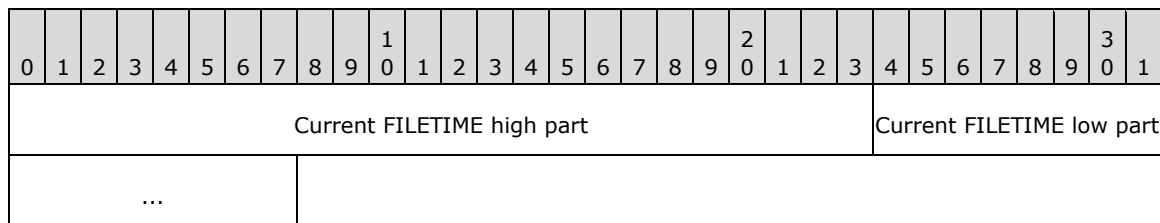
The content of the **email2:ConversationIndex** element is transferred as a **byte array** within the WBXML tags.

The client MUST NOT change the conversation index.

2.2.2.4.1 Conversation Index Header

The **Conversation Index Header** value is derived from the date and time when the message was originally sent by the server.

The **Conversation Index Header** value is set according to the description in the following diagram.



Current FILETIME high part (3 bytes): This field corresponds to the **Current FILETIME high part** field of the **Conversation Index Header** structure specified in [\[MS-OXOMSG\]](#) section 2.2.1.3.

Current FILETIME low part (2 bytes): This field corresponds to the **Current FILETIME low part** field of the **Conversation Index Header** structure specified in [\[MS-OXOMSG\]](#) section 2.2.1.3.

The data is stored in big-endian format – the five bytes of the time are written from most significant byte to least significant byte.

2.2.2.5 ConversationMode

In a **GetItemEstimate** command request ([\[MS-ASCMD\]](#) section 2.2.2.8), the **airsync:ConversationMode** element ([\[MS-ASCMD\]](#) section 2.2.3.36.1) is an optional child element

of the **getitemestimate:Collection** element ([MS-ASCMD] section 2.2.3.29.1) that enables or disables conversation-based filtering of item estimates.

In a **Sync** command request ([MS-ASCMD] section 2.2.2.20), the **airsync:ConversationMode** element ([MS-ASCMD] section 2.2.3.36.2) is an optional child element of the **airsync:Collection** element ([MS-ASCMD] section 2.2.3.29.2) that enables or disables conversation-based filtering and synchronization of conversation-based properties.

The **airsync:ConversationMode** element is defined in the **AirSync** namespace.

The value of this element is a **Boolean** data type, as specified in [MS-ASDTYPE] section 2.1. The value 1 enables conversation-based filtering of item estimates; the value 0 (zero) disables it. If this element is present without a value, the default value is 1.

2.2.2.6 DstFldId

The **itemoperations:DstFldId** element ([MS-ASCMD] section 2.2.3.49.1) is a required child element of the **itemoperations:Move** element ([MS-ASCMD] section 2.2.3.110.1) in an **ItemOperations** command request ([MS-ASCMD] section 2.2.2.9) that specifies the folder to which the conversation is moved.

The **itemoperations:DstFldId** element is defined in the **ItemOperations** namespace.

The value of this element is a **string**, as specified in [MS-ASDTYPE] section 2.7. The destination folder MUST be of type "IPF.Note". For more details about folder types, see [MS-OXOSFLD] section 2.2.8.

2.2.2.7 Move

In an **ItemOperations** command request ([MS-ASCMD] section 2.2.2.9), the **itemoperations:Move** element ([MS-ASCMD] section 2.2.3.110.1) is an optional **container** ([MS-ASDTYPE] section 2.2) element that identifies the body of the request as containing the operation that moves a given conversation. It has the following child elements:

- **itemoperations:ConversationId** (section [2.2.2.3.1](#))
- **itemoperations:DstFldId** (section [2.2.2.6](#))
- **itemoperations:Options** (section [2.2.2.9](#))

In an **ItemOperations** command response, the **itemoperations:Move** element is an optional **container** element that specifies the results of the attempt to move a conversation. It has the following child elements:

- **itemoperations:ConversationId** (section [2.2.2.3.1](#))
- **itemoperations:Status** (section [2.2.2.10](#))

The **itemoperations:Move** element is defined in the **ItemOperations** namespace.

2.2.2.8 MoveAlways

The **itemoperations:MoveAlways** element ([MS-ASCMD] section 2.2.3.111) is an optional child element of the **itemoperations:Options** element ([MS-ASCMD] section 2.2.3.118.2) in the **ItemOperations** command request ([MS-ASCMD] section 2.2.2.9) that indicates whether a conversation is always to be moved. When a conversation is set to always be moved, all e-mail messages in the conversation, including all future e-mail messages in the conversation, are moved from all folders except the Sent Items folder to the destination folder that is specified by the **DstFldId** element (section [2.2.2.6](#)).

The **itemoperations:MoveAlways** element is defined in the **ItemOperations** namespace.

This element is an empty tag element, meaning it has no value or data type. It is distinguished only by the presence or absence of the <MoveAlways/> tag. If this element is present, the conversation is set to always be moved.

2.2.2.9 Options

The **itemoperations:Options** element ([\[MS-ASCMD\]](#) section 2.2.3.118.2) is an optional **container** ([\[MS-ASDTYPE\]](#) section 2.2) element in an **ItemOperations** command request ([\[MS-ASCMD\]](#) section 2.2.2.9) that specifies the options for the move operation ([\[MS-ASCMD\]](#) section 2.2.3.110.1). It contains the following child element:

- **itemoperations:MoveAlways** (section [2.2.2.8](#))

The **itemoperations:Options** element is defined in the **ItemOperations** namespace.

2.2.2.10 Status

The **itemoperations:Status** element ([\[MS-ASCMD\]](#) section 2.2.3.166.7) is a required child element of the **itemoperations:Move** element ([\[MS-ASCMD\]](#) section 2.2.3.110.1) in the **ItemOperations** command response ([\[MS-ASCMD\]](#) section 2.2.2.9) that specifies the status of the move operation.

The **itemoperations:Status** element is defined in the **ItemOperations** namespace.

The value of this element is an **integer**, as specified in [\[MS-ASDTYPE\]](#) section 2.6. The following table lists some of the status values and their meanings.

Value	Meaning
1	Success. The server successfully completed the operation.
2	Protocol error. The XML is not valid.
3	Server error. There was a complete or partial failure of the operation.
6	Not Found. The conversation or destination folder does not exist.
105	Invalid Combination of IDs. The destination folder cannot be the recipient information cache .
156	Action not supported. The destination folder MUST be of type "IPF.Note". For more details about folder types, see [MS-OXOSFLD] section 2.2.8.

3 Protocol Details

3.1 Client Details

3.1.1 Abstract Data Model

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

Command request: A WBXML-formatted message that adheres to the command schemas specified in [\[MS-ASCMD\]](#).

E-mail message: An e-mail message includes the following information to support conversations:

- **Conversation ID:** A unique value that is associated with a conversation.
- **Conversation index:** An index that is used by clients to generate a conversation tree view.

The server creates a conversation ID and a conversation index on the e-mail item when the user creates or sends an e-mail message. The client does not change the conversation ID or the conversation index.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

3.1.4.1 Deleting a Conversation

When a conversation is deleted, all e-mail messages that are in the conversation are moved from the current folder to the **Deleted Items folder**. Future e-mail messages for the same conversation are not affected.

To delete a conversation, the client sends a **Sync** command request ([\[MS-ASCMD\]](#) section 2.2.2.20) that contains an **airsync:Delete** element ([\[MS-ASCMD\]](#) section 2.2.3.42.2) for each item in the conversation.

3.1.4.2 Flagging a Conversation for Follow-up

When a conversation is flagged for follow-up, the most recent e-mail message that is in the conversation and that is in the current folder is flagged. Clearing a flag on a conversation will clear flags on all e-mail messages that are in the conversation and that are in the current folder. Marking a flagged conversation as complete will mark all flagged e-mail messages that are in the conversation and that are in the current folder as complete.

To set a flag on a conversation, the client sends a **Sync** command request ([\[MS-ASCMD\]](#) section 2.2.2.20) that contains an **airsync:Change** element for at least one item in the conversation and in the current folder, as specified in [\[MS-ASCMD\]](#) section 2.2.3.24.

To clear a flag on a conversation, the client sends a **Sync** command request ([\[MS-ASCMD\]](#) section 2.2.2.20) that contains an **airsync:Change** element for each item in the conversation and in the current folder, as specified in [\[MS-ASCMD\]](#) section 2.2.3.24.

To mark a flagged conversation as complete, the client sends a **Sync** command request ([\[MS-ASCMD\]](#) section 2.2.2.20) that contains an **airsync:Change** element for each flagged item in the conversation and in the current folder, as specified in [\[MS-ASCMD\]](#) section 2.2.3.24.

3.1.4.3 Marking a Conversation as Read or Unread

When a conversation is marked as read or unread, all e-mail messages that are in the conversation and that are in the current folder are marked as such.

To mark a conversation as read, the client sends a **Sync** command request ([\[MS-ASCMD\]](#) section 2.2.2.20) that contains an **airsync:Change** element for each item in the conversation and in the current folder, as specified in [\[MS-ASCMD\]](#) section 2.2.3.24.

To mark a conversation as unread, the client sends a **Sync** command request ([\[MS-ASCMD\]](#) section 2.2.2.20) that contains an **airsync:Change** element for at least one item in the conversation and in the current folder, as specified in [\[MS-ASCMD\]](#) section 2.2.3.24.

3.1.4.4 Ignoring a Conversation

When a conversation is ignored, all e-mail messages in the conversation, including all future e-mail messages for that conversation, are moved from all folders except Sent Items folder to the Deleted Items folder.

To ignore a conversation, the client sends an **ItemOperations** command request ([\[MS-ASCMD\]](#) section 2.2.2.9) that contains an **itemoperations:Move** element ([\[MS-ASCMD\]](#) section 2.2.3.110.1) and its child elements, as specified in section [2.2.2.7](#). The **itemoperations:MoveAlways** element (section [2.2.2.8](#)) MUST be present in the command request and the **itemoperations:DstFldId** element (section [2.2.2.6](#)) MUST contain the ID of the Deleted Items folder. Multiple **itemoperations:Move** elements, one for each conversation to be moved, can be included within one **ItemOperations** command request. In this case, the **itemoperations:Move** elements are processed in the order specified.

3.1.4.5 Moving a Conversation from the Current Folder

When a conversation is moved from the current folder to another folder, all e-mail messages that are in the conversation are moved from the current folder to the destination folder.

To move a conversation from the current folder to a destination folder, the client sends a **MoveItems** command request for each item in the conversation, as specified in [\[MS-ASCMD\]](#) section 2.2.2.11.

3.1.4.6 Setting up a Conversation to Be Moved Always

When a conversation is set to be moved always, all e-mail messages in the conversation, including all future e-mail messages for that conversation, are moved from all folders except Sent Items folder to a destination folder.

To set a conversation to be moved always, the client sends an **ItemOperations** command request ([\[MS-ASCMD\]](#) section 2.2.2.9) that contains a **itemoperations:Move** element ([\[MS-ASCMD\]](#) section 2.2.3.110.1) and its child elements, as specified in section [2.2.2.7](#). The

itemoperations:MoveAlways element (section [2.2.2.8](#)) MUST be present in the command request. The client MUST NOT specify the **Outbox folder**, the **Drafts folder**, or the recipient information cache as the destination folder.

3.1.4.7 Finding a Conversation

Searching for a particular conversation will search across all folders for all e-mail messages that are in the conversation.

To search for a conversation, the client sends a **Search** command request ([\[MS-ASCMD\]](#) section 2.2.2.15) that contains the **search:ConversationId** element ([\[MS-ASCMD\]](#) section 2.2.3.35.2), as specified in section [2.2.2.3.2](#). The **search:ConversationId** element can be used in conjunction with other child elements of the **search:Query** element ([\[MS-ASCMD\]](#) section 2.2.3.132). The client MUST scope the query to the **Email** class by setting the **search:Name** element ([\[MS-ASCMD\]](#) section 2.2.3.113.2) to "Mailbox".

3.1.4.8 Synchronizing a Conversation

When a conversation is synchronized, all e-mail messages that are part of the conversation and that are in the specified folder are synchronized.

To synchronize a conversation, the client sends a **Sync** command request ([\[MS-ASCMD\]](#) section 2.2.2.20) that contains an **airsync:ConversationMode** element (section [2.2.2.5](#)) for the particular collection to be synchronized.

3.1.4.9 Applying a Conversation-based Filter

Conversation-based filtering augments the date-based filtering. For details about date-based filtering, see [\[MS-ASCMD\]](#) section 2.2.3.65.

When a conversation-based filter is applied to a synchronization of the current folder, the complete conversation is retrieved if any e-mail message in the conversation falls within the date-based filter.

To apply a conversation-based filter to a synchronization, the client includes the **airsync:ConversationMode** element (section [2.2.2.5](#)) in a **Sync** command request ([\[MS-ASCMD\]](#) section 2.2.2.20).

A conversation-based filter can also be applied to the **GetItemEstimate** command request ([\[MS-ASCMD\]](#) section 2.2.2.8) to get an estimate of the items that both meet the filter criteria and need to be synchronized. The client can apply the filter by including the **airsync:ConversationMode** element in a **GetItemEstimate** command request.

3.1.4.10 Requesting a Message Part

When a client synchronizes, searches, or fetches an e-mail message, the client can choose to receive a message part by including the **airsyncbase:BodyPartPreference** element ([\[MS-ASAIRS\]](#) section 2.2.2.11), as specified in section [2.2.2.2](#), in the **Sync** command request ([\[MS-ASCMD\]](#) section 2.2.2.20), the **Search** command request ([\[MS-ASCMD\]](#) section 2.2.2.15), or the **ItemOperations** command request ([\[MS-ASCMD\]](#) section 2.2.2.9). The value of the **airsyncbase:Type** element ([\[MS-ASAIRS\]](#) section 2.2.2.41.3) MUST be 2 to specify **Hypertext Markup Language (HTML)** as the encoding format.

3.1.5 Message Processing Events and Sequencing Rules

The client creates command requests and processes command responses as specified in section [3.1.4](#) and in [\[MS-ASCMD\]](#). The Exchange ActiveSync: Conversations Protocol uses the following commands:

- **GetItemEstimate** command ([MS-ASCMD] section 2.2.2.8)
- **ItemOperations** command ([MS-ASCMD] section 2.2.2.9)
- **MoveItems** command ([MS-ASCMD] section 2.2.2.11)
- **Search** command ([MS-ASCMD] section 2.2.2.15)
- **Sync** command ([MS-ASCMD] section 2.2.2.20)

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Server Details

3.2.1 Abstract Data Model

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

Command response: A WBXML-formatted message that adheres to the command schemas specified in [\[MS-ASCMD\]](#).

E-mail message: An e-mail message includes the following information to support conversations:

- **Conversation ID:** A unique value that is associated with a conversation.
- **Conversation Index:** An index that is used by clients to generate a conversation tree view.

The server creates a conversation ID and a conversation index on the e-mail item when the user sends an e-mail message. The client does not change the conversation ID or the conversation index.

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Higher-Layer Triggered Events

The server creates a conversation ID and a conversation index on the e-mail item when the user sends an e-mail message. The conversation ID is specified by the **email2:ConversationId** element ([\[MS-ASEMAIL\]](#) section 2.2.2.21) that is included in the **Email** class. The conversation index is specified by the **email2:ConversationIndex** element ([MS-ASEMAIL] section 2.2.2.22) that is included in the **Email** class. For details about the **Email** class, see [MS-ASEMAIL].

3.2.5 Message Processing Events and Sequencing Rules

3.2.5.1 Processing a Sync Command

3.2.5.1.1 Deleting a Conversation

When the server receives a request to delete a conversation, as specified in section [3.1.4.1](#), the server moves all e-mail messages that are in the conversation from the current folder to the Deleted Items folder. The server does not move future e-mail messages for the conversation.

The server sends a **Sync** command response, as specified in [\[MS-ASCMD\]](#) section 2.2.2.20.

3.2.5.1.2 Marking a Conversation as Read or Unread

When the server receives a request to mark a conversation as read or unread, as specified in section [3.1.4.3](#), the server marks all e-mails that are in the conversation and that are in the current folder as either read or unread, whichever is specified in the client request.

The server sends a **Sync** command response, as specified in [\[MS-ASCMD\]](#) section 2.2.2.20.

3.2.5.1.3 Flagging a Conversation for Follow-up

When the server receives a request to flag a conversation for follow-up, as specified in section [3.1.4.2](#), the server flags the most recent e-mail message that is in the conversation and that is in the current folder. If a flag is cleared on a conversation, the server clears flags on all e-mail messages that are in the conversation and that are in the current folder. If a flagged conversation is marked as complete, the server marks all flagged e-mail messages that are in the conversation and that are in the current folder as complete.

The server sends a **Sync** command response, as specified in [\[MS-ASCMD\]](#) section 2.2.2.20.

3.2.5.2 Processing a GetItemEstimate Command

When a conversation-based filter is applied to the **GetItemEstimate** command request, as specified in section [3.1.4.9](#), the server sends a **GetItemEstimate** command response ([\[MS-ASCMD\]](#) section 2.2.2.8) that specifies an estimate of the items that meet the filter criteria and need to be synchronized.

In the event of failure, the server sends the following status code.

Value	Meaning
4	Protocol error. The conversation-based filter cannot be applied to a folder that is not of the Email class.

3.2.5.3 Processing an ItemOperations Command

3.2.5.3.1 Ignoring a Conversation

When the server receives a request to ignore a conversation, as specified in section [3.1.4.4](#), the server moves all e-mail messages in the conversation, including all future e-mail messages for that conversation, from all folders except Sent Items folder to the Deleted Items folder.

The server sends an **ItemOperations** command response ([\[MS-ASCMD\]](#) section 2.2.2.9) that includes the **itemoperations:Status** element, as specified in section [2.2.2.10](#), and the **itemoperations:ConversationId** element (section [2.2.2.3.1](#)).

3.2.5.3.2 Always Moving a Conversation

When the server receives a request to always move a conversation, as specified in section [3.1.4.6](#), the server moves all e-mail messages in the conversation, including all future e-mail messages for that conversation, from all folders except Sent Items folder to the specified destination folder.

The server sends an **ItemOperations** command response ([\[MS-ASCMD\]](#) section 2.2.2.9) that includes the **itemoperations:Status** element, as specified in section [2.2.2.10](#), and the **itemoperations:ConversationId** element (section [2.2.2.3.1](#)).

3.2.5.4 Processing a MoveItems Command

When the server receives a request to move a conversation, as specified in section [3.1.4.5](#), the server moves all e-mail messages that are in the conversation from the current folder to the specified destination folder. The server sends a **MoveItems** command response, as specified in [\[MS-ASCMD\]](#) section 2.2.2.11.

3.2.5.5 Processing a Search Command

When the server receives a request to find a conversation, as specified in section [3.1.4.7](#), the server searches across all folders for all e-mail messages that are in the conversation and returns this set of e-mail messages. The server sends a **Search** command response, as specified in [\[MS-ASCMD\]](#) section 2.2.2.15.

3.2.5.6 Filtering

If an individual e-mail message is moved or deleted, and, as a result, the rest of the messages in the conversation fall out of filter, the server SHOULD send soft deletes (a **Sync** command response ([\[MS-ASCMD\]](#) section 2.2.2.20)) for those messages only during the aging-off process. The aging-off process is explained in the following paragraph. For details about soft deletes, see [\[MS-ASCMD\]](#) section 2.2.3.161.

The aging-off process is the process in which the server deletes objects from the client that are older than the given time-window. The time-window is specified by the client in the **airsync:FilterType** element ([\[MS-ASCMD\]](#) section 2.2.3.65.2) of the **Sync** command request. The server typically performs the aging-off process daily at midnight, but the time and frequency of execution are implementation-dependent.

An example of how the aging-off process is applied to conversations is as follows: Suppose that the client specifies a three-day time window. If any e-mail within a conversation has existed less than three days, all e-mails (going back in time to the oldest item in the mailbox) within that conversation will be synchronized to the client. Once the newest e-mail within the conversation has existed longer than three days, the server will send soft deletes for all of the e-mails that are within the conversation.

3.2.5.7 Sending a Message Part

If the client **Sync** command request ([\[MS-ASCMD\]](#) section 2.2.2.20), **Search** command request ([\[MS-ASCMD\]](#) section 2.2.2.15), or **ItemOperations** command request ([\[MS-ASCMD\]](#) section 2.2.2.9) includes the **airsyncbase:BodyPartPreference** element (section [2.2.2.2](#)), then the server uses the **airsyncbase:BodyPart** element (section [2.2.2.1](#)) to encapsulate the message part in the response. The **airsyncbase:BodyPart** element is not present in the response if the client did not request the message part, as specified in section [3.1.4.10](#).

The client's preferences affect the server response as follows:

- If the size of the message part exceeds the value specified in the **airsyncbase:TruncationSize** element ([MS-ASAIRS] section 2.2.2.40.1) of the request, then the server truncates the message part. The server includes the **airsyncbase:Truncated** element ([MS-ASAIRS] section 2.2.2.39.1) and the **airsyncbase:EstimatedDataSize** element ([MS-ASAIRS] section 2.2.2.23.2) in the response when it truncates the message part.
- If a value other than 2 is specified in the **airsyncbase:Type** element ([MS-ASAIRS] section 2.2.2.41.3) of the request, then the server returns a status value of 164.
- If the client enables conversation-based filtering, as specified in section 3.1.4.9, the server returns a message part for each of the conversation's e-mails that falls outside of the date-based filter.

The **airsyncbase:BodyPart** element and the **airsyncbase:Body** element ([MS-ASAIRS] section 2.2.2.9) can co-exist in the server response. The presence or absence of **airsyncbase:BodyPart** or **airsyncbase:Body** in the server response depends on the preferences specified in the client request, as summarized in the following table.

Request Contains	Response Contains
Neither airsyncbase:BodyPreference nor airsyncbase:BodyPartPreference elements	Only airsyncbase:Body element
Only airsyncbase:BodyPreference element	Only airsyncbase:Body element
Only airsyncbase:BodyPartPreference element	Only airsyncbase:BodyPart element
Both airsyncbase:BodyPreference and airsyncbase:BodyPartPreference elements	Both airsyncbase:Body and airsyncbase:BodyPart elements

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

4 Protocol Examples

4.1 Synchronizing Conversation Data

The following example demonstrates a **Sync** command response ([\[MS-ASCMD\]](#) section 2.2.2.20) from the server that contains data representing an e-mail message that is part of a conversation. The conversation ID (**email2:ConversationId** element ([\[MS-ASEMAIL\]](#) section 2.2.2.21)) and conversation index (**email2:ConversationIndex** element ([\[MS-ASEMAIL\]](#) section 2.2.2.22)) are included as child elements of the **airsync:ApplicationData** element ([\[MS-ASCMD\]](#) section 2.2.3.11). A server can choose any name for a namespace and then map its chosen name to the actual namespace name. This example shows alternate namespace names being used by the server.

Response:

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:email="Email:" xmlns:airsyncbase="AirSyncBase:"
xmlns:email2="Email2:" xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>1601897837</SyncKey>
      <CollectionId>7</CollectionId>
      <Status>1</Status>
      <Commands>
        <Add>
          <ServerId>7:1</ServerId>
          <ApplicationData>
            <email:To>"deviceuser" &lt;someone@example.com&gt;</email:To>
            <email:From>"deviceuser2" &lt;someone2@example.com&gt;</email:From>
            <email:Subject>Test report</email:Subject>
            <email:DateReceived>2009-03-21T07:04:26.948Z</email:DateReceived>
            <email:DisplayTo>deviceuser</email:DisplayTo>
            <email:ThreadTopic>Test report</email:ThreadTopic>
            <email:Importance>1</email:Importance>
            <email:Read>1</email:Read>
            <airsyncbase:Body>
              <airsyncbase:Type>1</airsyncbase:Type>
              <airsyncbase:EstimatedDataSize>100</airsyncbase:EstimatedDataSize>
              <airsyncbase:Truncated>1</airsyncbase:Truncated>
              <airsyncbase:Data>Test data</airsyncbase:Data>
            </airsyncbase:Body>
            <email:MessageClass>IPM.Note</email:MessageClass>
            <email:InternetCPID>20127</email:InternetCPID>
            <email:Flag/>
            <email:ContentClass>urn:content-classes:message</email:ContentClass>
            <airsyncbase:NativeBodyType>2</airsyncbase:NativeBodyType>
            <email2:ConversationId>BBA4726D4399D44C83297D4BD904ED2D</email2:ConversationId>
            <email2:ConversationIndex>01C9A9F345</email2:ConversationIndex>
            <email:Categories/>
          </ApplicationData>
        </Add>
      </Commands>
    </Collection>
  </Collections>
</Sync>
```

4.2 Ignoring a Conversation

The following example demonstrates a client request to ignore a conversation and the server response. For the sake of brevity, the value of the **email2:ConversationId** element ([\[MS-ASEMAIL\]](#) section 2.2.2.21) in the request and response is represented by "...". and the value of the **itemoperations:DstFldId** element ([\[MS-ASCMD\]](#) section 2.2.3.49.1) is represented by "DeletedItems-Folder-ID" in the request.

Request:

```
<ItemOperations xmlns="ItemOperations:">
  <Move>
    <ConversationId>...</ConversationId>
    <DstFldId>DeletedItems-Folder-ID</DstFldId>
    <Options>
      <MoveAlways/>
    </Options>
  </Move>
</ItemOperations>
```

Response:

```
<ItemOperations xmlns="ItemOperations:">
  <Response>
    <Status>1</Status>
    <Move>
      <Status>1</Status>
      <ConversationId>...</ConversationId>
    </Move>
  </Response>
</ItemOperations>
```

5 Security

5.1 Security Considerations for Implementers

None.

5.2 Index of Security Parameters

None.

6 Appendix A: Product Behavior

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

- Microsoft Exchange Server 2010
- Microsoft Exchange Server 2013
- Microsoft Exchange Server 2016
- Windows 8.1
- Windows Communication Apps
- Windows 10 operating system

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.

7 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
6 Appendix A: Product Behavior	Updated list of supported products.	Y	Content updated due to protocol revision.

8 Index

A

Abstract data model
[client](#) 15
[server](#) 18
[Applicability](#) 7

C

[Capability negotiation](#) 7
[Change tracking](#) 26
Client
[abstract data model](#) 15
[initialization](#) 15
[message processing](#) 17
[other local events](#) 18
[sequencing rules](#) 17
[timer events](#) 18
[timers](#) 15

D

Data model - abstract
[client](#) 15
[server](#) 18

E

Elements
[BodyPart](#) 9
[BodyPartPreference](#) 10
[ConversationId](#) 10
[ConversationIndex](#) 11
[ConversationMode](#) 12
[DstFldId](#) 13
[Move](#) 13
[MoveAlways](#) 13
[Options](#) 14
[Status](#) 14
[Elements message](#) 8
Examples
[ignoring a conversation](#) 22
[synchronizing conversation data](#) 22

F

[Fields - vendor-extensible](#) 7

G

[Glossary](#) 5

H

Higher-layer triggered events
[server](#) 18

I

[Implementer - security considerations](#) 24
[Index of security parameters](#) 24

[Informative references](#) 6
Initialization
[client](#) 15
[server](#) 18
[Introduction](#) 5

M

Message processing
[client](#) 17
Messages
[Elements](#) 8
[Namespaces](#) 8
[syntax](#) 8
[transport](#) 8

N

[Namespaces message](#) 8
[Normative references](#) 6

O

Other local events
[client](#) 18
[server](#) 21
[Overview \(synopsis\)](#) 6

P

[Parameters - security index](#) 24
[Preconditions](#) 7
[Prerequisites](#) 7
[Product behavior](#) 25

R

[References](#) 6
[informative](#) 6
[normative](#) 6
[Relationship to other protocols](#) 7

S

Security
[implementer considerations](#) 24
[parameter index](#) 24
Sequencing rules
[client](#) 17
Server
[abstract data model](#) 18
[higher-layer triggered events](#) 18
[initialization](#) 18
[other local events](#) 21
[timer events](#) 21
[timers](#) 18
[Standards assignments](#) 7

T

Timer events

[client](#) 18
[server](#) 21
Timers
[client](#) 15
[server](#) 18
[Tracking changes](#) 26
[Transport](#) 8
Triggered events - higher-layer
[server](#) 18

V

[Vendor-extensible fields](#) 7
[Versioning](#) 7