

[MS-ASCON]: ActiveSync Conversations Protocol Specification

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft's Open Specification Promise (available here: <http://www.microsoft.com/interop/osp>) or the Community Promise (available here: <http://www.microsoft.com/interop/cp/default.mspx>). 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.
- **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 is preliminary documentation for this technology. Since the documentation may change between this preliminary version and the final version, there are risks in relying on preliminary documentation. To the extent that you incur additional development obligations or any other costs as a result of relying on this preliminary documentation, you do so at your own risk.

Revision Summary			
Author	Date	Version	Comments
Microsoft Corporation	April 10, 2009	0.1	Initial Availability.

Preliminary

Table of Contents

1	Introduction	5
1.1	Glossary	5
1.2	References	5
1.2.1	Normative References	5
1.2.2	Informative References	6
1.3	Protocol Overview	6
1.4	Relationship to Other Protocols	6
1.5	Prerequisites/Preconditions	6
1.6	Applicability Statement	6
1.7	Versioning and Localization	6
1.8	Vendor-Extensible Fields	6
1.9	Standards Assignments	6
2	Messages	7
2.1	Transport	7
2.2	Message Syntax	7
2.2.1	Complex Types	7
2.2.1.1	ItemOperations Command Complex Types	7
2.2.1.1.1	Move	7
2.2.1.1.2	Move.Options	8
2.2.2	Elements	8
2.2.2.1	E-mail Class Elements	9
2.2.2.1.1	ConversationId	9
2.2.2.1.2	ConversationIndex	9
2.2.2.2	GetItemEstimate Command Elements	9
2.2.2.2.1	Collections.Collection.ConversationMode	9
2.2.2.3	ItemOperations Command Elements	10
2.2.2.3.1	Move.ConversationId	10
2.2.2.3.2	Move.DstFldId	10
2.2.2.3.3	Move.Options.MoveAlways	10
2.2.2.3.4	Response.Move.Status	10
2.2.2.3.5	Response.Move.ConversationId	10
2.2.2.4	Search Command Elements	11
2.2.2.4.1	Store.Query.ConversationId	11
2.2.2.5	Sync Command Elements	11
2.2.2.5.1	Collections.Collection.ConversationMode	11
3	Protocol Details	11
3.1	Client Details	11
3.1.1	Abstract Data Model	11
3.1.2	Timers	12
3.1.3	Initialization	12
3.1.4	Higher-Layer Triggered Events	12
3.1.4.1	Deleting a Conversation	12

3.1.4.2	Flagging a Conversation for Follow-up	12
3.1.4.3	Marking a Conversation as Read or Unread	12
3.1.4.4	Ignoring a Conversation	13
3.1.4.5	Moving a Conversation from the Current Folder	13
3.1.4.6	Setting up a Conversation to be Moved Always	13
3.1.4.7	Finding a Conversation	13
3.1.4.8	Synchronizing a Conversation	14
3.1.4.9	Applying a Conversation-based Filter	14
3.1.5	Message Processing Events and Sequencing Rules	14
3.1.6	Timer Events	14
3.1.7	Other Local Events	14
3.2	Server Details	14
3.2.1	Abstract Data Model	14
3.2.2	Timers	15
3.2.3	Initialization	15
3.2.4	Higher-Layer Triggered Events	15
3.2.5	Message Processing Events and Sequencing Rules	15
3.2.5.1	Processing a Sync Command	15
3.2.5.1.1	Deleting a Conversation	15
3.2.5.1.2	Marking a Conversation as Read or Unread	15
3.2.5.1.3	Flagging a Conversation for Follow-up	16
3.2.5.2	Processing a GetItemEstimate Command	16
3.2.5.3	Processing an ItemOperations Command	16
3.2.5.3.1	Ignoring a Conversation	16
3.2.5.3.2	Always Moving a Conversation	16
3.2.5.4	Processing a MoveItems Command	17
3.2.5.5	Processing a Search Command	17
3.2.5.6	Filtering	17
3.2.6	Timer Events	17
3.2.7	Other Local Events	17
4	Protocol Examples	17
4.1	Synchronization From the Server	17
4.2	Ignoring a Conversation	18
5	Security	19
5.1	Security Considerations for Implementers	19
5.2	Index of Security Parameters	19
6	Appendix A: Office/Exchange Behavior	19
	Index	20

1 Introduction

This document specifies the ActiveSync Conversations protocol, which is 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.

1.1 Glossary

The following terms are defined in [MS-OXGLOS]:

class
conversation
conversation ID
conversation index
GUID
WAP Binary XML (WBXML)
XML
XML schema

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

1.2 References

1.2.1 Normative References

[MS-ASCMD] Microsoft Corporation, "ActiveSync Command Reference Protocol Specification", December 2008.

[MS-ASDTYPE] Microsoft Corporation, "ActiveSync Data Types Protocol Specification", December 2008.

[MS-AEMAIL] Microsoft Corporation, "ActiveSync E-Mail Class Protocol Specification", December 2008.

[MS-ASWBXML] Microsoft Corporation, "ActiveSync WAP Binary XML (WBXML) Protocol Specification", December 2008.

[MS-OXGLOS] Microsoft Corporation, "Exchange Server Protocols Master Glossary", June 2008.

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

[XML] Bray, T., et al., "Extensible Markup Language (XML) 1.0 (Fifth Edition)", <http://www.w3.org/TR/REC-xml/>.

1.2.2 Informative References

None.

1.3 Protocol Overview

The ActiveSync Conversations protocol is 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 the Inbox 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

The ActiveSync Conversations protocol consists of a series of **XML** elements that are embedded inside a command request or a command response. For details about command requests and responses, see [MS-ASCMD]. The **WAP Binary XML (WBXML)**, as specified in [MS-ASWBXML], is used to transmit the XML markup that constitutes the request body or the response body.

1.5 Prerequisites/Preconditions

None.

1.6 Applicability Statement

This protocol is applicable in scenarios in which a client needs to synchronize its e-mail messages and files with a server and wants to present a view in which e-mail messages are grouped by conversation rather than listed serially.

1.7 Versioning and Localization

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

The ActiveSync Conversations protocol consists of a series of **XML** elements that are embedded inside a command request or a command response. The XML markup that constitutes the request body or the response body is transmitted between client and server by using **WAP Binary XML (WBXML)**, as specified in [MS-ASWBXML].

2.2 Message Syntax

The **XML** markup that is used by the ActiveSync Conversations protocol **MUST** be well-formed XML, as specified in [XML].

The XML elements that are used by the ActiveSync Conversations protocol are embedded inside a request or response for the following commands:

- **GetItemEstimate**
- **ItemOperations**
- **MoveItems**
- **Search**
- **Sync**

For details about the requests and responses for these commands, see [MS-ASCMD].

The complex types and elements of the ActiveSync Conversations protocol are defined in the following namespaces: **AirSync**, **Email2**, **ItemOperations**, and **Search**.

2.2.1 Complex Types

The following table summarizes the set of common **XML schema** complex types that are defined by this specification for the **ItemOperations** command. For details about the **ItemOperations** command, see [MS-ASCMD] section 2.2.1.8.

Complex Type	Description
Move	Indicates that the conversation is to be moved.
Move.Options	Specifies the options for the Move type.

2.2.1.1 ItemOperations Command Complex Types

The following complex types are defined in the **ItemOperations** namespace.

2.2.1.1.1 Move

The **Move** type indicates that a **conversation** is to be moved from all folders to a destination folder. The **Move** type is a required **container** type in an **ItemOperations** command request and response. It contains the following child elements:

- **ConversationId** (request and response) — see sections 2.2.2.3.1 and 2.2.2.3.6
- **DstFldId** (request only) — see section 2.2.2.3.2
- **Options** (request only) — see section 2.2.1.1.2
- **Status** (response only) — see section 2.2.2.3.4

The **container** type is specified in [MS-ASDTYPE] section 2.8.

2.2.1.1.2 *Move.Options*

The **Move.Options** type specifies the options for the **Move** type. The **Move.Options** type is an optional **container** type in an **ItemOperations** command request. It contains the following child element:

- **MoveAlways** (request only) — see section 2.2.2.3.3

The **container** type is specified in [MS-ASDTYPE] section 2.8.

2.2.2 Elements

The following tables summarize the set of common **XML schema** elements that are defined by this specification for the E-mail class, **GetItemEstimate** command, **ItemOperations** command, **Search** command, and **Sync** command. For details about the E-mail class, see [MS-ASEMAIL]. For details about the commands, see [MS-ASCMD] sections 2.2.1.7, 2.2.1.8, 2.2.1.14, and 2.2.1.19.

Elements **MUST NOT** have child elements in either the command request or command response.

Element	Description
ConversationId	Used by the E-mail class to specify the conversation ID for an e-mail message.
ConversationIndex	Used by the E-mail class to specify the conversation index for an e-mail message.
Collections.Collection.ConversationMode	Used by the GetItemEstimate command to enable conversation-based filtering of item estimates.
Move.ConversationId	Used by the ItemOperations command to specify the conversation ID of the conversation that is to be moved.
Move.DstFldId	Used by the ItemOperations command to specify the destination folder, which is the folder to which the conversation is moved.
Move.Options.MoveAlways	Used by the ItemOperations command to set up the conversation to be always moved.

Response.Move.Status	Used by the ItemOperations command to specify the status of the move action.
Response.Move.ConversationId	Used by the ItemOperations command to specify the conversation ID of the conversation that is moved.
Store.Query.ConversationId	Used by the Search command to specify the conversation ID of the conversation for which to search.
Collections.Collection..ConversationMode	Used by the Sync command to enable conversation-based filtering and synchronization of conversation-based properties.

2.2.2.1 E-mail Class Elements

The following elements are defined in the **Email2** namespace and are children of the **ApplicationData** element in the **Sync** command, as specified in [MS-ASCMD] sections 2.2.1.19.1.6 and 2.2.1.19.2.2.

2.2.2.1.1 ConversationId

The **ConversationId** element is a required element that specifies the **conversation ID** for an e-mail message.

The value of this element is a **GUID**. The value **MUST NOT** be null for the E-mail content **class**. The value can be null in the following cases:

- The content class of the item is not E-mail.
- The e-mail message never was transmitted and does not have a subject.

2.2.2.1.2 ConversationIndex

The **ConversationIndex** element is a required element that specifies the **conversation index** for an e-mail message.

The value of this element is an array of bytes in the following format.

{Timestamp of first message}+-{Timestamps of subsequent replies}

2.2.2.2 GetItemEstimate Command Elements

The following elements are defined in the **AirSync** namespace.

2.2.2.2.1 Collections.Collection.ConversationMode

The **Collections.Collection.ConversationMode** element enables **conversation**-based filtering of item estimates. This element is optional in the **GetItemEstimate** command request.

The **Collections.Collection.ConversationMode** element does not have a value.

2.2.2.3 ItemOperations Command Elements

The following elements are defined in the **ItemOperations** namespace.

2.2.2.3.1 Move.ConversationId

The **Move.ConversationId** element specifies the **conversation ID** of the **conversation** that is to be moved. This element is required in the **ItemOperations** command request.

The value of this element is a **GUID**.

2.2.2.3.2 Move.DstFldId

The **Move.DstFldId** element specifies the destination folder, which is the folder to which the **conversation** is moved. This element is required in the **ItemOperations** command request.

The value of this element is a string, as specified in [MS-ASDTYPE] section 2.1.

2.2.2.3.3 Move.Options.MoveAlways

The **Move.Options.MoveAlways** element sets 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 to a destination folder. This element is required in the **ItemOperations** command request.

The **Move.Options.MoveAlways** element does not have a value.

2.2.2.3.4 Response.Move.Status

The **Response.Move.Status** element specifies the status of the move action. This element is required in the **ItemOperations** command response.

The value of this element is an integer. The values listed in the following table are valid.

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 does not exist.
155	Protocol error. The Options element does not contain a MoveAlways element.
156	Action not supported. The destination folder cannot be Draft, Junk E-mail, or Outbox.

2.2.2.3.5 Response.Move.ConversationId

The **Response.Move.ConversationId** element specifies the **conversation ID** of the **conversation** that is moved. This element is required in the **ItemOperations** command response.

The value of this element is a **GUID**.

2.2.2.4 Search Command Elements

The following elements are defined in the **Search** namespace.

2.2.2.4.1 *Store.Query.ConversationId*

The **Store.Query.ConversationId** element specifies the **conversation ID** of the **conversation** for which to search. This element is optional in the **Search** command request.

The value of this element is a **GUID**.

2.2.2.5 Sync Command Elements

The following elements are defined in the **AirSync** namespace.

2.2.2.5.1 *Collections.Collection.ConversationMode*

The **Collections.Collection.ConversationMode** element enables **conversation**-based filtering and synchronization of conversation-based properties. This element is optional in the **Sync** command request.

This element does not have a value.

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 messages: Every e-mail message **MUST** include the following to support **conversations**:

- **Conversation ID:** A unique value that is associated with a conversation. This value is specified by the **ConversationId** element that is included in the E-mail class. For details about the E-mail class, see [MS-ASEMAIL].
- **Conversation index:** An index that is used by clients to generate a conversation tree view. This value is specified by the **ConversationIndex** element that is included in the E-mail class. For details about the E-mail class, see [MS-ASEMAIL].

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 **MUST 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 that contains a **Delete** element for each item in the conversation. For more details about the **Delete** element in the **Sync** command, see [MS-ASCMD].

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, clear a flag on a conversation, or mark a flagged conversation as complete, the client sends a **Sync** command request that contains a **Change** element, as specified in [MS-ASCMD] section 2.2.1.19.1.7.

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 or unread, the client sends a **Sync** command request that contains a **Change** element, as specified in [MS-ASCMD] section 2.2.1.19.1.7.

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 to the Deleted Items folder.

To ignore a conversation, the client sends an **ItemOperations** command request that contains a **Move** element and its child elements, as specified in section 2.2.1.1.1 of this document. The **Move.DstFldId** element **MUST** contain the ID of the Deleted Items folder. Multiple **Move** elements, one for each conversation to be moved, can be included within one **ItemOperations** request. In this case, the **Move** elements are processed in the order specified. For details about the **ItemOperations** command request, see [MS-ASCMD] section 2.2.1.8.2.

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, as specified in [MS-ASCMD] section 2.2.1.10.1.

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 to a destination folder.

To set a conversation to be moved always, the client sends an **ItemOperations** command request that contains a **Move** element and its child elements, as specified in section 2.2.1.1.1 of this document. The Outbox folder and Drafts folder **MUST NOT** be specified as the destination folder. For details about the **ItemOperations** command request, see [MS-ASCMD] section 2.2.1.8.2.

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 with the **Store.Query.ConversationId** element, which is specified in section 2.2.2.4.1 of this document. The **Store.Query.ConversationId** element can be used in conjunction with other **Query** child elements. The client **MUST** scope the query to the E-mail **class**. For details about the **Search** command request, see [MS-ASCMD] section 2.2.1.14.1.

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 with a **Collections.Collection.ConversationMode** element 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 2.2.1.19.1.21.

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 **Collections.Collection.ConversationMode** element in a **Sync** command request. For details about the **Sync** command request, see [MS-ASCMD] section 2.2.1.19.1.

A conversation-based filter can also be applied to the **GetItemEstimate** command to get an estimate of the items meeting the filter criteria that need to be synchronized. The client can apply the filter by including the **Collections.Collection.ConversationMode** element in a **GetItemEstimate** command request. For details about the **GetItemEstimate** command request, see [MS-ASCMD] section 2.2.1.7.1.

3.1.5 Message Processing Events and Sequencing Rules

None.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 *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 messages: Every e-mail item MUST include the following to support **conversations**:

- **Conversation ID:** A unique value that is associated with a conversation. This value is specified by the **ConversationId** element that is included in the E-mail class. For details about the E-mail class, see [MS-ASEMAIL].
- **Conversation index:** An index that is used by clients to generate a conversation tree view. This value is specified by the **ConversationIndex** element that is included in the E-mail class. For details about the E-mail class, see [MS-ASEMAIL].

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 MUST 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.

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*

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** response, as specified in [MS-ASCMD] section 2.2.1.19.2.

3.2.5.1.2 *Marking a Conversation as Read or Unread*

The server marks all e-mails that are in the conversation and that are in the current folder as read or unread, whichever is specified.

The server sends a **Sync** response, as specified in [MS-ASCMD] section 2.2.1.19.2.

3.2.5.1.3 *Flagging a Conversation for Follow-up*

If a **conversation** is flagged for follow-up, 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** response, as specified in [MS-ASCMD] section 2.2.1.19.2.

3.2.5.2 **Processing a GetItemEstimate Command**

When a **conversation**-based filter is applied to the **GetItemEstimate** command, the server sends 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. For details about the **GetItemEstimate** response, see [MS-ASCMD] section 2.2.1.7.2.

Value	Meaning
2	Protocol error. The conversation-based filter cannot be applied to a message that is not of the E-mail class.

3.2.5.3 **Processing an ItemOperations Command**

3.2.5.3.1 *Ignoring a Conversation*

When a **conversation** is ignored, the server moves all e-mail messages in the conversation, including all future e-mail messages for that conversation, from all folders to the Deleted Items folder.

The server's response includes the **Response.Move.Status** element, which contains one of the values specified in section 2.2.2.3.4, and the **Response.Move.ConversationId** element.

3.2.5.3.2 *Always Moving a Conversation*

When a **conversation** is set to be moved always, the server moves all e-mail messages in the conversation, including all future e-mail messages for that conversation, from all folders to a destination folder.

The server's response includes the **Response.Move.Status** element, which contains one of the values specified in section 2.2.2.3.4, and the **Response.Move.ConversationId** element.

3.2.5.4 Processing a MoveItems Command

The server moves all e-mail messages that are in the **conversation** from the current folder to the destination folder. The server sends a **MoveItems** response, as specified in [MS-ASCMD] section 2.2.1.10.2.

3.2.5.5 Processing a Search Command

The server searches across all folders for all e-mail messages that are in the conversation and returns a result set. For details about the **Search** command response, see [MS-ASCMD] section 2.2.1.14.2.

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 for those messages only during the aging-off process. For details about soft deletes, see [MS-ASCMD] sections 2.2.1.19.1.16 and 2.2.1.19.2.9.

The server will not distinguish between a date-based filter and a **conversation**-based filter when setting a flag, clearing a flag, or marking a flagged conversation as complete, as specified in section 3.1.4.2.

The server will not distinguish between a date-based filter and a conversation-based filter when marking a conversation as read or unread, as specified in section 3.1.4.3.

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

4 Protocol Examples

4.1 Synchronization From the Server

The following example shows the server returning an e-mail message. Note that the **conversation ID** and **conversation index** are included in the **ApplicationData** node.

Response from the server:

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:email="POOMMAIL:"
xmlns:airsyncbase="AirSyncBase:" xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>1601897837</SyncKey>
      <CollectionId>7</CollectionId>
```

```

<Status>1</Status>
<Commands>
  <Add>
    <ServerId>7:1</ServerId>
    <ApplicationData>
      <A:To>"deviceuser" &lt;someone@example.com&gt;</A:To>
      <A:From>"deviceuser2" &lt;someone2@example.com&gt;</A:From>
      <A:Subject>Test report</A:Subject>
      <A:DateReceived>2009-03-21T07:04:26.948Z</A:DateReceived>
      <A:DisplayTo>deviceuser</A:DisplayTo>
      <A:ThreadTopic>Test report</A:ThreadTopic>
      <A:Importance>1</A:Importance>
      <A:Read>1</A:Read>
      <B:Body>
        <B:Type>1</B:Type>
        <B:EstimatedDataSize>100</B:EstimatedDataSize>
        <B:Truncated>1</B:Truncated>
        <B:Data>Test data</B:Data>
      </B:Body>
      <A:MessageClass>IPM.Note</A:MessageClass>
      <A:InternetCPID>20127</A:InternetCPID>
      <A:Flag/>
      <A:ContentClass>urn:content-classes:message</A:ContentClass>
      <B:NativeBodyType>2</B:NativeBodyType>
    </ApplicationData>
  </Add>
</Commands>
</Collection>
</Collections>
</Sync>

```

4.2 Ignoring a Conversation

The following example shows the client's request to ignore a conversation and the server's response.

Request from the client:

```

<ItemOperations>
  <Move>
    <ConversationId>...</ConversationId>
    <DstFldId>DeletedItems-Folder-ID</DstFldId>
    <Options>
      <MoveAlways/>
    </Options>
  </Move>
</ItemOperations>

```

Response from the server:

```
<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: Office/Exchange Behavior

The information in this specification is applicable to the following versions of Office/Exchange:

- Microsoft Outlook 2010
- Microsoft Exchange Server 2010

Exceptions, if any, are noted below. Unless otherwise specified, any statement of optional behavior in this specification prescribed using the terms SHOULD or SHOULD NOT implies Office/Exchange behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies Office/Exchange does not follow the prescription.

Index

Applicability statement, 6

Client details, 11

Glossary, 5

Index of security parameters, 19

Informative references, 6

Introduction, 5

Message syntax, 7

Messages, 7

 Message syntax, 7

 Transport, 7

Normative references, 5

Office/Exchange behavior, 19

Prerequisites/preconditions, 6

Protocol details, 11

 Client details, 11

 Server details, 14

Protocol examples, 17

Protocol Overview, 6

References

 Informative references, 6

 Normative references, 5

Relationship to other protocols, 6

Security, 19

 Index of security parameters, 19

 Security considerations for implementers, 19

Security considerations for implementers, 19

Server details, 14

Standards assignments, 6

Transport, 7

Vendor-extensible fields, 6

Versioning and localization, 6