[MS-ASCON]: ActiveSync Conversations Protocol Specification

Revision Summary

Date	Revision History	Revision Class	Comments
04/10/2009	0.1.0	Major	Initial Availability.
07/15/2009	1.0.0	Major	Revised and edited for technical content.
11/04/2009	2.0.0	Major	Updated and revised the technical content.
02/10/2010	2.1.0	Minor	Updated the technical content.

Table of Contents

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]:

binary large object (BLOB)
class
conversation
conversation ID
conversation index
GUID
WAP Binary XML (WBXML)
XML
XML schema

The following terms are specific to this document:

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

1.2 References

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

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

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

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

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

[MS-DTYP] Microsoft Corporation, "Windows Data Types", March 2007, http://go.microsoft.com/fwlink/?LinkId=111558.

[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)", November 2008, 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)**, specified in [MS-ASWBXML], is used to transmit the XML markup that constitutes the request body or the response body.

The ActiveSync Conversations protocol defines elements and complex types according to the data type definitions that are specified in [MS-ASDTYPE].

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 Capability Negotiation

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 a conversation is to be moved.
Move.Options	Specifies the options for the **Move** type.
Response.Move	Specifies the results of the attempt to move a conversation.

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. It contains the following child elements and type:

- **ConversationId** element — see section 2.2.2.3.1
- **DstFldId** element — see section 2.2.2.3.2
- **Options** type — see section 2.2.1.1.2

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** element — see section 2.2.2.3.3

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

2.2.1.1.3 Response.Move

The **Response.Move** type specifies the results of the attempt to move a conversation. The **Response.Move** type is a required **container** type in an **ItemOperations** command response. It contains the following child elements:

- **ConversationId** element — see section 2.2.2.3.5
- **Status** element — see section 2.2.2.3.4

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 (**GetItemEstimate** command)	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.

Element	Description
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 (**Sync** command)	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 2.2.1.19.1.7 and 2.2.1.19.2.2.

2.2.2.1.1 ConversationId

The **ConversationId** element specifies a unique identifier for a conversation. This element is a required child element of the **ApplicationData** type in the **Sync** command response, as specified in [MS-ASCMD] section 2.2.1.19.2.2

The value of this element is a **byte array**, as specified in [MS-ASDTYPE] section 2.11. 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.

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

2.2.2.1.2 ConversationIndex

The **ConversationIndex** element specifies the conversation index for an e-mail message. This element is a required child element of the **ApplicationData** type in the **Sync** command response, as specified in [MS-ASCMD] section 2.2.1.19.2.2

The value of this element is a **byte array**, as specified in [MS-ASDTYPE] section 2.11. The value comprises a set of timestamps, which can be used by a client to generate a tree-view of a conversation. The first timestamp identifies the date and time when the message was originally sent by the server. Additional timestamps are added when the message is forwarded or replied to.

Each timestamp is five bytes. The first 5-byte timestamp in the **ConversationIndex** specifies the system time when the message was sent by the server, converted to the **FILETIME** structure format, as specified in [MS-DTYP] section 2.3.1. Each additional timestamp is divided as follows:

- One bit containing a code representing the difference between the current time and the time stored in the first timestamp. This bit is 0 if the difference is less than .02 seconds and greater than two years and 1 if the difference is less than one second and greater than 56 years.

- Thirty one bits containing the difference between the current time and the time in the first timestamp, expressed in **FILETIME** units. This part of the value is produced by using one of two strategies, depending on the value of the first bit. If this bit is zero, the server discards the high 15 bits and the low 18 bits. If this bit is one, the server discards the high 10 bits and the low 23 bits.
- Four bits containing a random number generated by calling the Win32 function **GetTickCount**.
- Four bits containing a sequence count that is taken from part of the random number.

The content of the **ConversationIndex** element is transferred as an opaque **BLOB** within the WBXML tags.

The client MUST NOT change the **ConversationIndex** value.

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 **ConversationMode** element enables or disables conversation-based filtering of item estimates. This element is an optional child element of the **Collections.Collection** type in the **GetItemEstimate** command request. For details about the **Collections.Collection** type, see [MS-ASCMD] section 2.2.1.7.1.3.

The value of this element is a **Boolean**, as specified in [MS-ASDTYPE] section 2.3. The value 1 enables conversation-based filtering of item estimates; the value zero disables it. If this element is present without a value, the default is 1.

2.2.2.3 ItemOperations Command Elements

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

2.2.2.3.1 Move.ConversationId

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

The value of this element is a **byte array**, as specified in [MS-ASDTYPE] section 2.11.

2.2.2.3.2 Move.DstFldId

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

The value of this element is a **string**, as specified in [MS-ASDTYPE] section 2.1. The destination folder MUST NOT be the Draft folder, the Outbox folder, or the Recipient Information cache.

2.2.2.3.3 Move.Options.MoveAlways

The **MoveAlways** element indicates whether a conversation is 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 an optional child element of the **Move.Options** type in the **ItemOperations** command request.

This element is a flag, which does not have a value. If this element is present, the conversation is set to be moved always.

2.2.2.3.4 Response.Move.Status

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

The value of this element is an **integer**, as specified in [MS-ASDTYPE] section 2.2. 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.
105	Invalid Combination of IDs. The destination folder cannot be the Recipient Information Cache.
155	Protocol error. The **Move.Options** type does not contain a **MoveAlways** element.
156	Action not supported. The destination folder cannot be Drafts or Outbox.

2.2.2.3.5 Response.Move.ConversationId

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

The value of this element is a **byte array**, as specified in [MS-ASDTYPE] section 2.11.

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 **ConversationId** element specifies the conversation ID of the conversation for which to search. This element is an optional child element of the **Store.Query** type in the **Search** command request. For details about the **Store.Query** type, see [MS-ASCMD] section 2.2.1.14.1.2.

The value of this element is a **byte array**, as specified in [MS-ASDTYPE] section 2.11.

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 **ConversationMode** element enables or disables conversation-based filtering and synchronization of conversation-based properties. This element is an optional child element of the **Collections.Collection** type in the **Sync** command request. For details about the **Collections.Collection** type, see [MS-ASCMD] section 2.2.1.19.1.10.

The value of this element is a **Boolean**, as specified in [MS-ASDTYPE] section 2.3. The value 1 enables conversation-based filtering and synchronization of conversation-based properties; the value zero disables it. If this element is present without a value, the default is 1.

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 includes 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 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 that contains a **Delete** element for each item in the conversation. For more details about the **Delete** element in the **Sync** command request, see [MS-ASCMD] section 2.2.1.19.1.18.

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 2.2.1.19.1.8.

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 2.2.1.19.1.8.

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** type and its child elements, as specified in section 2.2.1.1.1 of this document. The **Move.Options.MoveAlways** element MUST be present and the **Move.DstFldId** element MUST contain the ID of the Deleted Items folder. Multiple **Move** types, one for each conversation to be moved, can be included within one **ItemOperations** request. In this case, the **Move** types 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** type and its child elements, as specified in section 2.2.1.1.1 of this document. The **Move.Options.MoveAlways** element MUST be present. The client MUST NOT specify the Outbox folder, the Drafts folder, or the Recipient Information Cache 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 child elements of the **Store.Query** type. The client MUST scope the query to the Email class by setting the **Store.Name** element to "Mailbox" and then doing one of the following:

- Use the **Store.Query.CollectionId** element to specify an e-mail folder.
- Use the **Store.Query.Class** element to specify the Email class

For details about the **Search** command request, see [MS-ASCMD] section 2.2.1.14.1. For details about the **Store.Name** element, see [MS-ASCMD] section 2.2.1.14.1.1. For details about the **Store.Query** type and its child elements, see [MS-ASCMD] section 2.2.1.14.1.2.

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 2.2.1.19.1.23.

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 includes 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 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.

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 either read or unread, whichever is specified in the client's request (see section 3.1.4.3).

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 this set of e-mail messages. 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 (**Sync** command response) for those messages only during the aging-off process. The aging-off process is explained in the following paragraph. For details about the **Sync** command and soft deletes, see [MS-ASCMD] sections 2.2.1.19 and 2.2.1.19.2.10, respectively.

The aging-off process is the process in which the server deletes from the client objects that are older than the given time-window, which is specified by the client in the **Collections.Collection.Options.FilterType** element of the **Sync** command request. (For details about this element, see [MS-ASCMD] section 2.2.1.19.1.23.) The server typically performs the aging-off process daily at midnight, but the time and frequency of execution is 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 is less than three days old, 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 becomes older than three days, the server will send soft deletes for all of the e-mails that are within the conversation.

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. 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.<1> These alternate names map to the actual namespaces as follows:

POOMMAIL maps to Email

POOMMAIL2 maps to Email2

Response from the server:

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:A="POOMMAIL:" xmlns:B="AirSyncBase:"
xmlns:C="POOMMAIL2:" 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>
            <C:ConversationId>BBA4726D4399D44C83297D4BD904ED2D</C:ConversationId>
            <C:ConversationIndex>01C9A9F345</C:ConversationIndex>
            <A:Categories/>
          </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: Product Behavior

The information in this specification is applicable to the following product versions. References to product versions include released service packs.

- Microsoft Exchange Server 2010

Exceptions, if any, are noted below. If a service pack number appears with the product version, behavior changed in that service pack. The new behavior also applies to subsequent service packs of the product unless otherwise specified.

Unless otherwise specified, any statement of optional behavior in this specification 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 product does not follow the prescription.

<1> Section 4.1: Exchange uses alternate names that are prefixed with "POOM", which stands for "Pocket Outlook Object Model".

7 Change Tracking

This section identifies changes made to [MS-ASCON] protocol documentation between November 2009 and February 2010 releases. Changes are classed as major, minor, or editorial.

Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- A protocol is deprecated.
- The removal of a document from the documentation set.
- Changes made for template compliance.

Minor changes do not affect protocol interoperability or implementation. Examples are updates to fix technical accuracy or ambiguity at the sentence, paragraph, or table level.

Editorial changes apply to grammatical, formatting, and style issues.

No changes means that the document is identical to its last release.

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

- New content added.
- Content update.
- 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.
- New content added for template compliance.
- Content updated for template compliance.

- Content removed for template compliance.
- Obsolete document removed.

Editorial changes always have the revision type "Editorially updated."

Some important terms used in revision 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.

Changes are listed in the following table. If you need further information, please contact protocol@microsoft.com.

Section	**Tracking number (if applicable) and description**	**Major change (Y or N)**	**Revision Type**
2.2.2.1.2 ConversationIndex	53781 Added link to [MS-DTYP].	N	Content update.
3.1.1 Abstract Data Model	51043 Removed Normative wording.	N	Content update.
3.1.4.1 Deleting a Conversation	51044 Added relevant section number to [MS-ASCMD] reference.	N	Content update.
3.2.1 Abstract Data Model	51043 Removed Normative wording.	N	Content update.
4.1 Synchronization From the Server	51046 Updated the namespace declarations in the example code, and added a product behavior note about the "POOM" namespace names that Exchange uses.	N	Content update.

8 Index