[MS-ASCMD]:
Exchange ActiveSync:
Command Reference Protocol

Revision Summary

Date	Revision History	Revision Class	Comments
12/03/2008	1.0.0	Major	Initial Release.
01/15/2009	1.01	Editorial	Revised and edited technical content.
03/04/2009	1.02	Editorial	Revised and edited technical content.
04/10/2009	2.0.0	Major	Updated technical content and applicable product releases.
07/15/2009	3.0.0	Major	Revised and edited for technical content.
11/04/2009	4.0.0	Major	Updated and revised the technical content.
02/10/2010	5.0.0	Major	Updated and revised the technical content.
05/05/2010	6.0.0	Major	Updated and revised the technical content.
08/04/2010	7.0	Major	Significantly changed the technical content.
11/03/2010	8.0	Major	Significantly changed the technical content.
03/18/2011	9.0	Major	Significantly changed the technical content.
08/05/2011	10.0	Major	Significantly changed the technical content.
10/07/2011	10.1	Minor	Clarified the meaning of the technical content.
01/20/2012	11.0	Major	Significantly changed the technical content.
04/27/2012	11.1	Minor	Clarified the meaning of the technical content.
07/16/2012	12.0	Major	Significantly changed the technical content.
10/08/2012	13.0	Major	Significantly changed the technical content.
02/11/2013	14.0	Major	Significantly changed the technical content.
07/26/2013	15.0	Major	Significantly changed the technical content.
11/18/2013	15.0	No change	No changes to the meaning, language, or formatting of the technical content.
02/10/2014	15.0	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

Copyright © 2014 Microsoft Corporation.

Release: February 10, 2014

1   Introduction

The Exchange ActiveSync: Command Reference Protocol enables a client, typically a mobile device, to synchronize and exchange objects with a server. These objects include email messages, **Short Message Service (SMS)** messages, attachments, folders, **contact (3)** information, **meetings**, **calendar** data, tasks, notes and documents.

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in RFC 2119. Sections 1.5 and 1.9 are also normative but cannot contain those terms. All other sections and examples in this specification are informative.

1.1   Glossary

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

Active Directory
ambiguous name resolution (ANR)
ASCII
Augmented Backus-Naur Form (ABNF)
certificate revocation list (CRL)
certification authority (CA)
domain
Domain Name System (DNS)
GUID
Hypertext Transfer Protocol (HTTP)
resource
Secure Sockets Layer (SSL)
Universal Naming Convention (UNC)
UTF-8
XML
XML namespace

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

address book
address list
alias
Autodiscover server
base64 encoding
binary large object (BLOB)
calendar
Calendar folder
Calendar object
character set
condition
contact
Contacts folder
conversation
Deleted Items folder
distribution list
Drafts folder
ghosted
Global Address List (GAL)
Hypertext Markup Language (HTML)

Inbox folder
locale
mailbox
meeting
meeting request
message body
message store
Multipurpose Internet Mail Extensions (MIME)
named property
OOF message
organizer
Out of Office (OOF)
Outbox folder
Personal Information Manager (PIM)
plain text
recipient information cache
Root folder
S/MIME (Secure/Multipurpose Internet Mail Extensions)
search folder
Sent Items folder
Short Message Service (SMS)
Simple Mail Transfer Protocol (SMTP)
special folder
Transport Neutral Encapsulation Format (TNEF)
Uniform Resource Identifier (URI)
Uniform Resource Locator (URL)
Wireless Application Protocol (WAP) Binary XML (WBXML)
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

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

1.2.1 Normative References

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

[MS-ASAIRS] Microsoft Corporation, "Exchange ActiveSync: AirSyncBase Namespace Protocol".

[MS-ASCAL] Microsoft Corporation, "Exchange ActiveSync: Calendar Class Protocol".

[MS-ASCNTC] Microsoft Corporation, "Exchange ActiveSync: Contact Class Protocol".

[MS-ASCON] Microsoft Corporation, "Exchange ActiveSync: Conversations Protocol".

[MS-ASDOC] Microsoft Corporation, "Exchange ActiveSync: Document Class Protocol".

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

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

[MS-ASHTTP] Microsoft Corporation, "Exchange ActiveSync: HTTP Protocol".

[MS-ASMS] Microsoft Corporation, "Exchange ActiveSync: Short Message Service (SMS) Protocol".

[MS-ASNOTE] Microsoft Corporation, "Exchange ActiveSync: Notes Class Protocol".

[MS-ASPROV] Microsoft Corporation, "Exchange ActiveSync: Provisioning Protocol".

[MS-ASRM] Microsoft Corporation, "Exchange ActiveSync: Rights Management Protocol".

[MS-ASTASK] Microsoft Corporation, "Exchange ActiveSync: Tasks Class Protocol".

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

[MS-OXCICAL] Microsoft Corporation, "iCalendar to Appointment Object Conversion Algorithm".

[MS-OXDSCLI] Microsoft Corporation, "Autodiscover Publishing and Lookup Protocol".

[MS-OXOSFLD] Microsoft Corporation, "Special Folders Protocol".

[MS-OXTNEF] Microsoft Corporation, "Transport Neutral Encapsulation Format (TNEF) Data Algorithm".

[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

[RFC2447] Dawson, F., Mansour, S., and Silverberg, S., "iCalendar Message-Based Interoperability Protocol (iMIP)", RFC 2447, November 1998, http://www.rfc-editor.org/rfc/rfc2447.txt

[RFC3986] Berners-Lee, T., Fielding, R., and Masinter, L., "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005, http://www.ietf.org/rfc/rfc3986.txt

[RFC5234] Crocker, D., Ed., and Overell, P., "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008, http://www.rfc-editor.org/rfc/rfc5234.txt

[RFC5751] Ramsdell, B., and Turner, S., "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification", RFC 5751, January 2010, http://www.rfc-editor.org/rfc/rfc5751.txt

[WBXML1.2] Martin, B., and Jano, B., Eds., "WAP Binary XML Content Format", W3C Note, June 1999, http://www.w3.org/1999/06/NOTE-wbxml-19990624

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

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

1.2.2   Informative References

[MSDN-ADDP] Microsoft Corporation, "Establishing an ActiveSync Desktop-Device Partnership", http://msdn.microsoft.com/en-us/library/aa917378.aspx

[MSFT-AUTODISCOVER] Masterson, J., Turick, J., Smith IV, R., "White Paper: Exchange 2007 Autodiscover Service", http://technet.microsoft.com/en-us/library/bb332063(EXCHG.80).aspx

[MSFT-DNS-SRV] Microsoft Corporation, "A new feature is available that enables Outlook 2007 to use DNS Service Location (SRV) records to locate the Exchange Autodiscover service", http://support.microsoft.com/?kbid=940881

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

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

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

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

1.3   Overview

This protocol consists of a set of **XML**-based commands that are used by a client device to synchronize and exchange its email, files, and data with a server.

The client first uses the **Autodiscover** command (section 2.2.2.1) to get a user's account configuration. The client can then view and modify server data related to that account, including email messages and attachments, folders, contacts, and calendar requests.

The client then uses the **Provision** command (section 2.2.2.12) to send device information to the server and to get and subsequently acknowledge security policy settings from the server. Next, the client uses the **FolderSync** command (section 2.2.2.4) to retrieve the folder hierarchy of the user.

Next, the client typically uses the **GetItemEstimate** command (section 2.2.2.7) to retrieve the number of changes that need to be downloaded to the client via the first **Sync** command (section 2.2.2.19) request. This is immediately followed by the **Sync** command, to get a synchronization key and then messages from the server. Optionally, the **Ping** command (section 2.2.2.11) or **Sync** command can then be issued to keep the device up-to-date on any server changes.

The client processes outgoing email using the **SendMail** (section 2.2.2.15), **SmartReply** (section 2.2.2.18), and **SmartForward** (section 2.2.2.17) commands. For incoming messages, the client can call the **ItemOperations** command (section 2.2.2.8) to fetch the message, and then use the **MoveItems** command (section 2.2.2.10). **S/MIME** messages are processed with the **ResolveRecipients** (section 2.2.2.13) and **ValidateCert** (section 2.2.2.20) commands.

The client uses the **FolderSync**, **FolderCreate** (section 2.2.2.2), **FolderUpdate** (section 2.2.2.5), and **FolderDelete** (section 2.2.2.3) commands to update, create, and delete **mailbox** folders on the server.

For **meeting requests**, the client uses the **MeetingResponse** command (section 2.2.2.9).

The client can set and request server parameters by using the **Settings** command (section 2.2.2.16).

The client uses the **Search** command (section 2.2.2.14) to find particular items on the server.

1.4   Relationship to Other Protocols

The ActiveSync commands described in this document are sent and received over a **Hypertext Transfer Protocol (HTTP)** connection, as described in [RFC2616] in an HTTP **POST** method. The information contained in the HTTP **POST** header is described in [MS-ASHTTP]. The information

contained in the HTTP message is sent and received in **Wireless Application Protocol (WAP) Binary XML (WBXML)** format, as described in [MS-ASWBXML], where the content of the WBXML adheres to the commands described in this document.

Some of the ActiveSync commands described in this document are used to synchronize or retrieve more than one class of content. For example, the **Ping** command (section 2.2.2.11) can be used to monitor changes to the email, note, contact, calendar, or task classes. The elements included in the **Ping** command change depending on which content class is being monitored. Because each content class is used by multiple commands, each content class is described by an individual protocol. The content class protocols are described in [MS-ASEMAIL], [MS-ASCNTC], [MS-ASDOC], [MS-ASCAL], [MS-ASNOTE], [MS-ASMS], and [MS-ASTASK].

Another protocol that contains elements used by multiple commands is [MS-ASAIRS]. [MS-ASAIRS] describes the **AirSyncBase** namespace, which is used by multiple commands to specify the formatting preference of body content, truncation sizes, and other commonly used elements.

This document describes all of the ActiveSync commands except for the **Provision** command, which is described in [MS-ASPROV] section 2.2.2.43.

The **Autodiscover** command (section 2.2.2.1) is described in this document, but more information about Autodiscover publishing and lookup are available in [MS-OXDSCLI].

All simple data types in this document conform to the data type definitions in [MS-ASDTYPE].

For information about how to control the view of related email messages or conversations, see [MS-ASCON].

For information about how outbound Short Message Service (SMS) email messages are sent from mobile devices, see [MS-ASMS].

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

1.5 Prerequisites/Preconditions

This protocol assumes that authentication has been performed by the underlying protocols.

1.6 Applicability Statement

This protocol is applicable in scenarios where a client has to synchronize its messages and files with a server.

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 contained in request or response messages between a client and server. The XML block containing the command and parameter elements is transmitted in either the request body of a request, or in the response body of a response. The request body and request response are always preceded by the HTTP header, as specified in [MS-ASHTTP].

All command messages use WBXML, except for the **Autodiscover** command, which uses plain XML. For more details about WBXML, see [MS-ASWBXML].

2.2 Message Syntax

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**	
airsyncbase	**AirSyncBase**	[MS-ASAIRS]
autodiscover	**http://schemas.microsoft.com/exchange/autodiscover/mobilesync/requestschema/2006**	
calendar	**Calendar**	[MS-ASCAL]
composemail	**ComposeMail**	
contacts	**Contacts**	[MS-ASCNTC]
contacts2	**Contacts2**	[MS-ASCNTC]
documentlibrary	**DocumentLibrary**	[MS-ASDOC]
email	**Email**	[MS-ASEMAIL]
email2	**Email2**	[MS-ASEMAIL]
folderhierarchy	**FolderHierarchy**	
gal	**GAL**	
getitemestim	**GetItemEstimate**	

Prefix	Namespace URI	Reference
ate		
itemoperations	**ItemOperations**	
meetingresponse	**MeetingResponse**	
move	**Move**	
notes	**Notes**	[MS-ASNOTE]
ping	**Ping**	
provision	**Provision**	[MS-ASPROV]
resolverecipients	**ResolveRecipients**	
rm	**RightsManagement**	[MS-ASRM]
search	**Search**	
settings	**Settings**	
tasks	**Tasks**	[MS-ASTASK]
validatecert	**ValidateCert**	
xs	**http://www.w3.org/2001/XMLSchema**	[XMLSCHEMA1]

2.2.2   Commands

2.2.2.1   Autodiscover

The **Autodiscover** command facilitates the discovery of core account configuration information by using the user's **Simple Mail Transfer Protocol (SMTP)** address as the primary input. For information about the **Autodiscover** service, see [MSFT-AUTODISCOVER]. For more details about the Autodiscover HTTP Service, see [MS-OXDSCLI].

The **Autodiscover** command request and response messages are sent in XML format, not WBXML format.

When sending an **Autodiscover** command request, the Content-Type header value MUST be set to text/xml.<1> For more details about the Content-Type header, see [MS-ASHTTP] section 2.2.1.1.2.2.

The client SHOULD use the **Autodiscover** command as an initial response to common HTTP errors. Common HTTP errors are specified in [MS-ASHTTP] section 2.2.2.1.1. **Autodiscover** has the ability to retrieve an updated **URL** when a mailbox has been moved, a user is trying to connect to a server that cannot access the user's mailbox, or when there is a more efficient server to use to reach the user's mailbox.

After a successful **Autodiscover** command response, the client sends an HTTP **Options** command to the server identified in the **Autodiscover** command response. The HTTP **Options** command returns the newly supported protocol versions and commands if they changed due to the **Autodiscover** command.

The http://schemas.microsoft.com/exchange/autodiscover/mobilesync/requestschema/2006 namespace is the primary namespace for the **Autodiscover** command and this section of the specification. Elements referenced in this section that are not defined in the http://schemas.microsoft.com/exchange/autodiscover/mobilesync/requestschema/2006 namespace use the namespace prefixes defined in section 2.2.1.

The following table lists the elements that are used in **Autodiscover** command requests and responses.

Element name	Scope	Reference
Autodiscover	Request and Response	section 2.2.3.15
Request	Request	section 2.2.3.137.1
EMailAddress	Request and Response	section 2.2.3.52
AcceptableResponseSchema	Request	section 2.2.3.1
Response	Response	section 2.2.3.140.1
Culture	Response	section 2.2.3.38
User	Response	section 2.2.3.173
DisplayName	Response	section 2.2.3.47.1
Action	Response	section 2.2.3.6
Redirect	Response	section 2.2.3.134
Settings	Response	section 2.2.3.153.1
Server	Response	section 2.2.3.149
Type	Response	section 2.2.3.170.1
Url	Response	section 2.2.3.172
Name	Response	section 2.2.3.110.1
ServerData	Response	section 2.2.3.150
Error	Response	section 2.2.3.60
Status	Response	section 2.2.3.162.1
Message	Response	section 2.2.3.98
DebugData	Response	section 2.2.3.40
ErrorCode	Response	section 2.2.3.61

The **XML schema** for the **Autodiscover** command request is described in section 6.3. The XML schema for the **Autodiscover** command response is described in section 6.4.

2.2.2.2 FolderCreate

The **FolderCreate** command creates a new folder as a child folder of the specified parent folder. A parent ID of 0 (zero) signifies the mailbox **Root folder**.

The **FolderCreate** command cannot be used to create a **recipient information cache** or a subfolder of a recipient information cache.

The **FolderHierarchy** namespace is the primary namespace for this section. Elements referenced in this section that are not defined in the **FolderHierarchy** namespace use the namespace prefixes defined in section 2.2.1.

The following table lists the elements that are used in **FolderCreate** command requests and responses.

Element name	Scope	Reference
FolderCreate	Request and Response	section 2.2.3.67
SyncKey	Request and Response	section 2.2.3.166.1
ParentId	Request	section 2.2.3.119.1
DisplayName	Request	section 2.2.3.47.2
Type	Request	section 2.2.3.170.2
Status	Response	section 2.2.3.162.2
ServerId	Response	section 2.2.3.151.1

The XML schema for the **FolderCreate** command request is described in section 6.6. The XML schema for the **FolderCreate** command response is described in section 6.7.

2.2.2.3 FolderDelete

The **FolderDelete** command deletes a folder from the server. The **ServerId** (section 2.2.3.151.2) of the folder is passed to the server in the **FolderDelete** command request (section 2.2.2.3), which deletes the collection with the matching identifier. The server then sends a response indicating the status of the deletion.

The **FolderDelete** command cannot be used to delete a recipient information cache. Attempting to delete a recipient information cache using this command results in a **Status** element (section 2.2.3.162.3) value of 3.

The **FolderHierarchy** namespace is the primary namespace for this section. Elements referenced in this section that are not defined in the **FolderHierarchy** namespace use the namespace prefixes defined in section 2.2.1.

The following table lists the elements that are used in **FolderDelete** command requests and responses.

Element name	Scope	Reference
FolderDelete	Request and Response	section 2.2.3.68
SyncKey	Request and Response	section 2.2.3.166.1

Element name	Scope	Reference
ServerId	Request	section 2.2.3.151.2
Status	Response	section 2.2.3.162.3

The XML schema for the **FolderDelete** command request is described in section 6.8. The XML schema for the **FolderDelete** command response is described in section 6.9.

2.2.2.4 FolderSync

The **FolderSync** command synchronizes the collection hierarchy but does not synchronize the items in the collections themselves.

This command works similarly to the **Sync** command (section 2.2.2.19). An initial **FolderSync** command with a **SyncKey** element (section 2.2.3.166.2) value of 0 (zero) is required in order to obtain the list of folders and the synchronization key associated with that list. The synchronization key is returned in the **SyncKey** element of the response if the **FolderSync** command succeeds. This synchronization key is used in subsequent **FolderSync** commands to obtain folder hierarchy changes.

Unlike a **Sync** request, there is no **GetChanges** element (section 2.2.3.76) submitted in the **FolderSync** request (section 2.2.2.4) to get changes from the server. All folders MUST be returned to the client when initial folder synchronization is done with a synchronization key of 0 (zero).

The **FolderHierarchy** namespace is the primary namespace for this section. Elements referenced in this section that are not defined in the **FolderHierarchy** namespace use the namespace prefixes defined in section 2.2.1.

The following table lists the elements that are used in **FolderSync** command requests and responses.

Element name	Scope	Reference
FolderSync	Request and Response	section 2.2.3.71
SyncKey	Request and Response	section 2.2.3.166.2
Status	Response	section 2.2.3.162.4
Changes	Response	section 2.2.3.25
Count	Response	section 2.2.3.37
Update	Response	section 2.2.3.171
ServerId	Response	section 2.2.3.151.3
ParentId	Response	section 2.2.3.119.2
DisplayName	Response	section 2.2.3.47.3
Type	Response	section 2.2.3.170.3
Delete	Response	section 2.2.3.42.1
Add	Response	section 2.2.3.7.1

The XML schema for the **FolderSync** command request is described in section 6.11. The XML schema for the **FolderSync** command response is described in section 6.12.

2.2.2.5 FolderUpdate

The **FolderUpdate** command moves a folder from one location to another on the server. The command is also used to rename a folder.

The **FolderUpdate** command cannot be used to update a recipient information cache, or to move a folder under the recipient information cache. Attempting to update a recipient information cache using this command results in a **Status** element (section 2.2.3.162.5) value of 3.

The **FolderHierarchy** namespace is the primary namespace for this section. Elements referenced in this section that are not defined in the **FolderHierarchy** namespace use the namespace prefixes defined in section 2.2.1.

The following table lists the elements that are used in **FolderUpdate** command requests and responses.

Element name	Scope	Reference
FolderUpdate	Request and Response	section 2.2.3.72
SyncKey	Request and Response	section 2.2.3.166.1
ServerId	Request	section 2.2.3.151.4
ParentId	Request	section 2.2.3.119.3
DisplayName	Request	section 2.2.3.47.2
Status	Response	section 2.2.3.162.5

The XML schema for the **FolderUpdate** command request is described in section 6.13. The XML schema for the **FolderUpdate** command response is described in section 6.14.

2.2.2.6 GetAttachment

The **GetAttachment** command retrieves an email attachment from the server.<2>

Attachment data is not automatically included with email messages in a synchronization. Instead, an **Attachment** element ([MS-ASAIRS] section 2.2.2.2) is included for each attachment. The **Attachment** element specifies the name of the attachment in the **DisplayName** element ([MS-ASAIRS] section 2.2.2.11). Attachments are explicitly retrieved by attachment name by using the **GetAttachment** command.

This command is issued within the HTTP **POST** command, and does not require any additional information in an XML body. The name of the attachment to be retrieved is specified in the *AttachmentName* command parameter.

The content of the attachment is returned in the response body with the content type being specified in the Content-Type header of the response. When the Content-Type header is missing, this indicates that the default encoding of 7-bit **ASCII** has been used.

If the **GetAttachment** command is used to retrieve an attachment that has been deleted on the server, a 500 status code is returned in the HTTP **POST** response.

2.2.2.7 GetItemEstimate

The **GetItemEstimate** command gets an estimate of the number of items in a collection or folder on the server that have to be synchronized.

The **GetItemEstimate** namespace is the primary namespace for this section. Elements referenced in this section that are not defined in the **GetItemEstimate** namespace use the namespace prefixes defined in section 2.2.1.

The following table lists the elements that are used in **GetItemsEstimate** command requests and responses.

Element name	Scope	Reference
GetItemEstimate	Request and Response	section 2.2.3.77
Collections	Request	section 2.2.3.31.1
Collection	Request and Response	section 2.2.3.29.1
airsync:SyncKey	Request	section 2.2.3.166.3
CollectionId	Request and Response	section 2.2.3.30.1
airsync:ConversationMode	Request	section 2.2.3.36.1
airsync:Options	Request	section 2.2.3.115.1
airsync:Class	Request	section 2.2.3.27.1
airsync:FilterType	Request	section 2.2.3.64.1
airsync:MaxItems	Request	section 2.2.3.93.1
Response	Response	section 2.2.3.140.2
Status	Response	section 2.2.3.162.6
Estimate	Response	section 2.2.3.62

The XML schema for the **GetItemEstimate** command request is described in section 6.16. The XML schema for the **GetItemEstimate** command response is described in section 6.17.

2.2.2.8 ItemOperations

The **ItemOperations** command acts as a container for the **Fetch** element (section 2.2.3.63.1), the **EmptyFolderContents** element (section 2.2.3.55), and the **Move** element (section 2.2.3.107.1) to provide batched online handling of these operations against the server.

Operations that are contained within the **ItemOperations** element (section 2.2.3.85) MUST be executed by the server in the specified order. The server MUST report the status per operation to the client. Accordingly, the client correlates these responses to the initial operation and proceeds appropriately.

The **ItemOperations** command cannot perform operations on items in the recipient information cache.

The fetch operation is intended to be used on Microsoft Windows SharePoint Services technology, for **Universal Naming Convention (UNC)** access, or for user mailbox access. Document metadata, search results, and mailbox items and attachments can be retrieved.

The **EmptyFolderContents** operation enables the client to empty a folder of all its items. Clients typically use the **EmptyFolderContents** operation to clear out all items in the **Deleted Items folder** if the user runs out of storage quota.

The **Move** operation moves a **conversation** to a destination folder.

The **ItemOperations** namespace is the primary namespace for this section. Elements referenced in this section that are not defined in the **ItemOperations** namespace use the namespace prefixes defined in section 2.2.1.

The following table lists the elements that are used in **ItemOperations** command requests and responses.

Element name	Scope	Reference
ItemOperations	Request and Response	section 2.2.3.85
EmptyFolderContents	Request and Response	section 2.2.3.55
airsync:CollectionId	Request and Response	section 2.2.3.30.2
Options	Request	section 2.2.3.115.2
DeleteSubFolders	Request	section 2.2.3.44
Fetch	Request and Response	section 2.2.3.63.1
Store	Request	section 2.2.3.163.1
airsync:ServerId	Request and Response	section 2.2.3.151.5
documentlibrary:LinkId	Request and Response	section 2.2.3.48
search:LongId	Request	section 2.2.3.89.1
airsyncbase:FileReference	Request	section 2.2.3.8
Schema	Request	section 2.2.3.145
Range	Request and Response	section 2.2.3.130.1
UserName	Request	section 2.2.3.177.1
Password	Request	section 2.2.3.122.1
airsync:MIMESupport	Request	section 2.2.3.100.1
Move	Request and Response	section 2.2.3.107.1
ConversationId	Request and Response	section 2.2.3.35.1
DstFldId	Request	section 2.2.3.49.1
MoveAlways	Request	section 2.2.3.108
Status	Response	section 2.2.3.162.7

Element name	Scope	Reference
Response	Response	section 2.2.3.140.3
airsync:Class	Response	section 2.2.3.27.2
Properties	Response	section 2.2.3.128.1
Total	Response	section 2.2.3.169.1
Data	Response	section 2.2.3.39.1
Part	Response	section 2.2.3.120
Version	Response	section 2.2.3.181

The XML schema for the **ItemOperations** command request is described in section 6.19. The XML schema for the **ItemOperations** command response is described in section 6.20.

2.2.2.8.1 Delivery of Content Requested by Fetch

Because the **ItemOperations** response (section 2.2.2.8) potentially contains large amounts of binary data, the client can choose a delivery method that is most efficient for its implementation by providing the following two methods for delivering the content that is requested by the **Fetch** element (section 2.2.3.63.1):

- Inline

- Multipart

The inline method of delivering binary content is including data encoded with **base64 encoding** inside the WBXML. The inline method generally requires the client to read the whole response into memory in order to parse it, thereby consuming a large amount of memory. The inline method is the default method for **ItemOperations** responses.
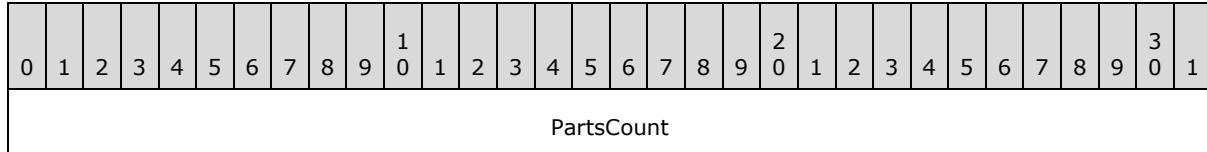
The multipart method of delivering content is a multipart structure with the WBXML being the first part, and the requested data populating the subsequent parts. This format enables a client to handle large files without consuming large amounts of memory, because a file is read in pieces, one piece at a time. Clients request multipart responses by including the MS-ASAcceptMultiPart header (specified in [MS-ASHTTP] section 2.2.1.1.2.3) set to "T" (**TRUE**) in the request.

Multipart responses have a Content-Type header with the value "application/vnd.ms-sync.multipart". The format of the body of the response is a **MultiPartResponse** structure, specified in section 2.2.2.8.1.1.

For an example of requesting and receiving a multipart response, see section 4.10.5.

2.2.2.8.1.1 MultiPartResponse

The **MultiPartResponse** structure represents the format of the body of a multipart **ItemOperations** command response.

0	1	2	3	4	5	6	7	8	9	1 0	1	2	3	4	5	6	7	8	9	2 0	1	2	3	4	5	6	7	8	9	3 0	1
PartsCount																															

PartsMetaData (variable)
...
Parts (variable)
...

PartsCount (4 bytes): This field is an integer that indicates the number of **PartMetaData** structures contained in the **PartsMetaData** field. It also indicates the number of binary parts contained in the **Parts** field.

PartsMetaData (variable): This field is an array of **PartMetaData** structures, as specified in section 2.2.2.8.1.1.1. The length of the array is equal to the value of the **PartsCount** field.

Parts (variable): This field is an array of bytes that contains the data for the parts of the multipart response.

The data in the **Parts** field is interpreted by using the offset and length information in the **PartsMetaData** field.

2.2.2.8.1.1.1 PartMetaData

The **PartMetaData** structure is used in the **PartsMetaData** field of the **MultiPartReponse** structure specified in section 2.2.2.8.1.1. It contains the offset and length of a single part of a multipart **ItemOperations** command response.

0	1	2	3	4	5	6	7	8	9	1 0	1	2	3	4	5	6	7	8	9	2 0	1	2	3	4	5	6	7	8	9	3 0	1
Offset																															
Length																															

Offset (4 bytes): This field is an integer that contains the number of bytes from the beginning of the **MultiPartResponse** structure where a part's data is located.

Length (4 bytes): This field is an integer that contains the number of bytes that comprise the part's data.

2.2.2.9 MeetingResponse

The **MeetingResponse** command is used to accept, tentatively accept, or decline a meeting request in the user's **Inbox folder** or **Calendar folder**.<3>

The **MeetingResponse** command can only be used when the **CollectionId** element (section 2.2.3.30.3) is being used to synchronize the folder that contains the meeting request item.

The **SendMail** command (section 2.2.2.15) can be used to send a message back to the meeting **organizer**, notifying him or her that the meeting request was accepted or declined.

The **MeetingResponse** namespace is the primary namespace for this section. Elements referenced in this section that are not defined in the **MeetingResponse** namespace use the namespace prefixes defined in section 2.2.1.

The following table lists the elements that are used in **MeetingResponse** command requests and responses.

Element name	Scope	Reference
MeetingResponse	Request and Response	section 2.2.3.96
Request	Request	section 2.2.3.137.2
UserResponse	Request	section 2.2.3.178
CollectionId	Request	section 2.2.3.30.3
RequestId	Request and Response	section 2.2.3.138
InstanceId	Request	section 2.2.3.83.1
Result	Response	section 2.2.3.142.1
Status	Response	section 2.2.3.162.8
CalendarId	Response	section 2.2.3.18
search:LongId	Request	section 2.2.3.89.2

The XML schema for the **MeetingResponse** command request is described in section 6.21. The XML schema for the **MeetingResponse** command response is described in section 6.22.

2.2.2.10 MoveItems

The **MoveItems** command moves an item or items from one folder on the server to another.

The item to be moved is identified by its server ID in the **MoveItems** command request (section 2.2.2.10). The source and destination folders are also identified by their server IDs in the command request. The **MoveItems** command response shows the status of the move, the message that was moved, and the new message ID.

When items are moved between folders on the server, the client receives **Delete** (section 2.2.3.42) and **Add** (section 2.2.3.7) operations the next time the client synchronizes the affected folders.

An item that has been successfully moved to a different folder can be assigned a new server ID by the server.

The **Move** namespace is the primary namespace for this section. Elements referenced in this section that are not defined in the **Move** namespace use the namespace prefixes defined in section 2.2.1.

The following table lists the elements that are used in **MoveItems** command requests and responses.

Element name	Scope	Reference
MoveItems	Request and Response	section 2.2.3.109
Move	Request	section 2.2.3.107.2

Element name	Scope	Reference
SrcMsgId	Request and Response	section 2.2.3.160
SrcFldId	Request	section 2.2.3.159
DstFldId	Request	section 2.2.3.49.2
Response	Response	section 2.2.3.140.4
Status	Response	section 2.2.3.162.9
DstMsgId	Response	section 2.2.3.50

The XML schema for the **MoveItems** command request is described in section 6.23. The XML schema for the **MoveItems** command response is described in section 6.24.

2.2.2.11 Ping

The **Ping** command is used to request that the server monitor specified folders for changes that would require the client to resynchronize.

The body of the request contains a list of folders on the server about which the client is requesting notifications and an interval of time that specifies how long the server SHOULD wait before responding if no new items are added to the specified folders. If the **Ping** element (section 2.2.3.126) is specified in an XML request body, either the **HeartbeatInterval** element or the **Folders** element (section 2.2.3.70), or both elements MUST be specified.

The **Ping** namespace is the primary namespace for this section. Elements referenced in this section that are not defined in the **Ping** namespace use the namespace prefixes defined in section 2.2.1.

The following table lists the elements that are used in **Ping** command requests and responses.

Element name	Scope	Reference
Ping	Request and Response	section 2.2.3.126
HeartbeatInterval	Request and Response	section 2.2.3.79.1
Folders	Request and Response	section 2.2.3.70
Folder	Request and Response	section 2.2.3.66
Id	Request	section 2.2.3.81
Class	Request	section 2.2.3.27.3
Status	Response	section 2.2.3.162.10
MaxFolders	Response	section 2.2.3.92

The XML schema for the **Ping** command request is described in section 6.25. A **Ping** command can be sent with no body, in which case the cached version is used. The XML schema is applied only to requests that have a body. The XML schema for the **Ping** command response is described in section 6.26.

2.2.2.12 Provision

The **Provision** command enables client devices to request from the server the security policy settings that the administrator sets, such as the minimum personal identification number (PIN) password length requirement. The **Provision** command is specified in [MS-ASPROV].

2.2.2.13 ResolveRecipients

The **ResolveRecipients** command is used by clients to resolve a list of supplied recipients, to retrieve their free/busy information, and optionally, to retrieve their S/MIME certificates so that clients can send encrypted S/MIME email messages.<4>

The **ResolveRecipients** namespace is the primary namespace for this section. Elements referenced in this section that are not defined in the **ResolveRecipients** namespace use the namespace prefixes defined in section 2.2.1.

The following table lists the elements that are used in **ResolveRecipients** command requests and responses.

Element name	Scope	Reference
ResolveRecipients	Request and Response	section 2.2.3.139
To	Request and Response	section 2.2.3.168
Options	Request	section 2.2.3.115.3
CertificateRetrieval	Request	section 2.2.3.22
MaxCertificates	Request	section 2.2.3.91
MaxAmbiguousRecipients	Request	section 2.2.3.90
Availability	Request and Response	section 2.2.3.16
StartTime	Request	section 2.2.3.161.1
EndTime	Request	section 2.2.3.58.1
Picture	Request and Response	section 2.2.3.125.1
MaxSize	Request	section 2.2.3.95.1
MaxPictures	Request	section 2.2.3.94.1
Response	Response	section 2.2.3.140.5
Status	Response	section 2.2.3.162.11
RecipientCount	Response	section 2.2.3.133
Recipient	Response	section 2.2.3.132
Type	Response	section 2.2.3.170.4
DisplayName	Response	section 2.2.3.47.4
EmailAddress	Response	section 2.2.3.53.1

Release: February 10, 2014

Element name	Scope	Reference
MergedFreeBusy	Response	section 2.2.3.97
Certificates	Response	section 2.2.3.23.1
CertificateCount	Response	section 2.2.3.21
Certificate	Response	section 2.2.3.19.1
MiniCertificate	Response	section 2.2.3.102
Data	Response	section 2.2.3.39.2

The XML schema for the **ResolveRecipients** command request is described in section 6.27. The XML schema for the **ResolveRecipients** command response is described in section 6.28.

2.2.2.14 Search

The **Search** command is used to find entries in an **address book**, mailbox, or document library (UNC or Windows SharePoint Services).

The **Search** namespace is the primary namespace for this section. Elements referenced in this section that are not defined in the **Search** namespace use the namespace prefixes defined in section 2.2.1.

The following table lists the elements that are used in **Search** command requests and responses.

Element name	Scope	Reference
Search	Request and Response	section 2.2.3.146
Store	Request and Response	section 2.2.3.163.2
Name	Request	section 2.2.3.110.2
Query	Request	section 2.2.3.129
And	Request	section 2.2.3.10
FreeText	Request	section 2.2.3.73
airsync:Class	Request and Response	section 2.2.3.27.4
airsync:CollectionId	Request and Response	section 2.2.3.30.4
ConversationId	Request	section 2.2.3.35.2
GreaterThan	Request	section 2.2.3.78
email:DateReceived	Request	section 2.2.3.51
Value	Request	section 2.2.3.180
LessThan	Request	section 2.2.3.87
EqualTo	Request	section 2.2.3.59
documentlibrary:LinkId	Request	section 2.2.3.48

Element name	Scope	Reference
Options	Request	section 2.2.3.115.4
airsync:MIMESupport	Request	section 2.2.3.100.2
Range	Request and Response	section 2.2.3.130.2
UserName	Request	section 2.2.3.177.2
Password	Request	section 2.2.3.122.2
DeepTraversal	Request	section 2.2.3.41
RebuildResults	Request	section 2.2.3.131
Picture	Request	section 2.2.3.125.2
MaxSize	Request	section 2.2.3.95.2
MaxPictures	Request	section 2.2.3.94.2
Status	Response	section 2.2.3.162.12
Response	Response	section 2.2.3.140.6
Result	Response	section 2.2.3.142.2
LongId	Response	section 2.2.3.89.3
Properties	Response	section 2.2.3.128.2
gal:Picture	Response	section 2.2.3.125.2
gal:Data	Response	section 2.2.3.39.3
Total	Response	section 2.2.3.169.2

The XML schema for the **Search** command request is described in section 6.30. The XML schema for the **Search** command response is described in section 6.31.

The Accept-Language header in a **Search** command request is used to define the **locale** of the client so that the search is relevant. If the accept language is not specified, the search is conducted by using the server language.

Searching the **Global Address List (GAL)**

The **Search** command is used to find contacts and recipients in the GAL, and to retrieve information about them. When a search query matches more than one GAL entry, the **Search** command MUST return as many entries as requested, up to a total of 100 entries by default.

For each GAL entry that is found, the **Search** command returns all the non-empty properties that are indexed by the online **ambiguous name resolution (ANR)** in the global catalog server—for example, email **alias**, display name, first and last names, company name, and so on.

The client can optionally specify the maximum number of entries to retrieve in the **Search** command request by specifying the range. The server MUST return entries up to the number that is requested, and MUST also indicate the total number of entries that are found.

The text query string that is provided to the **Search** command is used in a prefix-string match. For example, if the client performs a **Search** with a **Query** element value of "Michael A.", the command returns the entries that contain the search string in any text field, such as "Michael Alexander", "Michael Allen". Because the **Search** command matches the **Query** element value against all GAL text properties that are indexed by using ANR, the client can also search by email address, company name, and so on.

The ANR system indexes the following properties:

- Display name

- Alias

- FirstName

- LastName

- EmailAddress

The **Search** command results are sorted by the server according to their ordering in the GAL (that is, by the display name property). Because of how the search results are sorted, the client could have to sort the results to display results in a relevant manner to users. For example, a search for "123" might return all GAL entries that have mailing addresses or email addresses that begin with 123. The client can choose to display matching email addresses before mailing addresses, if they know their users use email addresses more frequently than mailing addresses, or mailing addresses before email addresses if mailing addresses are used more frequently.

The **Range** option is a zero-based index specifier in the form of "m-n". For more details about the meaning of the **Range** values, see section 2.2.3.130.2.
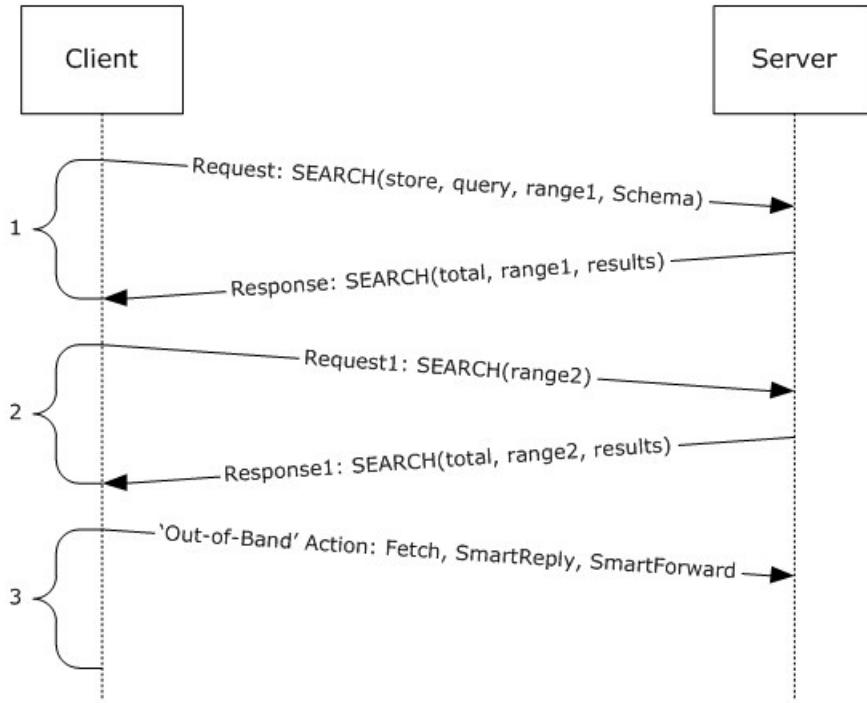
Searching Outside the GAL

Typical Usage

Essentially, search involves the following three phases:

1. The client issues a request for specific search results.

2. The client uses subsequent requests to retrieve more results by incrementing the range.

3. Any actions on the search results are carried out by using other protocol commands, such as **ItemOperations**, **SmartReply**, or **SmartForward**.

The following figure shows the typical usage of the **Search** command to retrieve successive result sets from the server and then perform some action based on those results (for example, retrieve the full **message body (2)** for an email search result).

Figure 1: Search command process

2.2.2.15   SendMail

The **SendMail** command is used by clients to send **MIME**-formatted email messages to the server. The body of the request differs depending on the value of the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.4.

If the MS-ASProtocolVersion header is set to "12.1", the body of the **SendMail** request is the MIME representation of the message to be sent. In this case, the client sets the Content-Type header to "message/rfc822".

If the MS-ASProtocolVersion header is set to "14.0" or "14.1", the body of the **SendMail** request is WBXML data generated by using the XML elements specified in the table below.

Clients SHOULD NOT save messages directly to the local **Sent Items folder**; instead, clients SHOULD use the **SaveInSentItems** element (section 2.2.3.144) to automatically save the messages on the server. It is not possible to reconcile the local Sent Items folder with the server's Sent Items folder by using the **Sync** command (section 2.2.2.19). Items in the server's Sent Items folder can be added to the client by using the **Sync** command, but it is not possible to add items that are in the local Sent Items folder to the server.

Note that the *From: MIME* header in the outgoing message is set on the server to the primary email address of the authenticated user.

The **ComposeMail** namespace is the primary namespace for this section. Elements referenced in this section that are not defined in the **ComposeMail** namespace use the namespace prefixes defined in section 2.2.1.

The following table lists the elements that are used in **SendMail** command requests and responses.

Element name	Scope	Reference
SendMail	Request and Response	section 2.2.3.148
ClientId	Request	section 2.2.3.28.1
AccountId	Request	section 2.2.3.3.1
SaveInSentItems	Request	section 2.2.3.144
Mime	Request	section 2.2.3.99
Status	Response	section 2.2.3.162.13

The XML schema for the **SendMail** command request is described in section 6.32. The XML schema for the **SendMail** command response is described in section 6.33. If the message was sent successfully, the server returns an empty response.

```
HTTP/1.1 200 OK
Date: Thu, 03 Sep 2009 21:05:44 GMT
Content-Length: 0
```

2.2.2.16 Settings

The **Settings** command supports get and set operations on global properties and **Out of Office (OOF)** settings for the user. The **Settings** command also sends device information to the server, implements the device password/personal identification number (PIN) recovery, and retrieves a list of the user's email addresses.

The **Get** (section 2.2.3.75) and **Set** (section 2.2.3.152) operations act on **named properties**. In the context of the **Get** and **Set** operations, each named property can contain a set of property-specific data nodes.

The **Settings** command can contain multiple **Get** and **Set** requests and responses in any order. The implication of this batching mechanism is that commands are executed in the order in which they are received and that the ordering of **Get** and **Set** responses will match the order of those commands in the request.

The following is the generic form of the **Settings** request, where "PropertyName" is a named property (that is, the actual name of the property).

```
<Settings>
    <PropertyName>
        Data nodes
    </PropertyName>
    ...
</Settings>
```

The **Settings** command can be used on the following named properties:

▪ **OOF** (section 2.2.3.112)

▪ **DevicePassword** (section 2.2.3.46)

▪ **DeviceInformation** (section 2.2.3.45)

- **UserInformation** (section 2.2.3.176)

- **RightsManagementInformation** (section 2.2.3.143)

Clients SHOULD send **DeviceInformation** parameters in a **Set** block to the server as soon as the client has been provisioned, and before the **FolderSync** command (section 2.2.2.4), so that the server can use this information to determine what the device has access to.<5>

The argument or data nodes are **Get** or **Set**, which can also have their own arguments. It is up to the individual property handlers to parse and interpret them as necessary.

It is possible to have between 0 (zero) and 5 **PropertyName** elements in a **Settings** request (that is, each of the four named properties can appear zero or one time in a request). Each property MUST be processed in order. There can be cases in which one property call affects another property call. The server will return responses in the same order in which they were requested.

Each response message contains a **Status** element (section 2.2.3.162.14) value for the command, which addresses the success or failure of the **Settings** command, followed by **Status** values for each of the changes made to the **Oof**, **DeviceInformation**, **DevicePassword** or **UserInformation** elements.

The **Status** node MUST indicate success if the **Settings** response contains property responses. If the command was not successful, the processing of the request cannot begin, no property responses are returned, and the **Status** node MUST indicate a protocol error.

Any error other than a protocol error is returned in the **Status** elements of the individual property responses. All property responses, regardless of the property, MUST contain a **Status** element to indicate success or failure. This **Status** node MUST be the first node in the property response.

The **Settings** namespace is the primary namespace for this section. Elements referenced in this section that are not defined in the **Settings** namespace use the namespace prefixes defined in section 2.2.1.

The following table lists the elements that are used in **Settings** command requests and responses.

Element name	Scope	Reference
Settings	Request and Response	section 2.2.3.153.2
RightsManagementInformation	Request and Response	section 2.2.3.143
Get	Request and Response	section 2.2.3.75
Oof	Request and Response	section 2.2.3.112
BodyType	Request and Response	section 2.2.3.17
Set	Request	section 2.2.3.152
OofState	Request and Response	section 2.2.3.114
StartTime	Request and Response	section 2.2.3.161.2
EndTime	Request and Response	section 2.2.3.58.2
OofMessage	Request and Response	section 2.2.3.113
AppliesToInternal	Request and Response	section 2.2.3.14

Element name	Scope	Reference
AppliesToExternalKnown	Request and Response	section 2.2.3.12
AppliesToExternalUnknown	Request and Response	section 2.2.3.13
Enabled	Request and Response	section 2.2.3.56
ReplyMessage	Request and Response	section 2.2.3.136
DevicePassword	Request and Response	section 2.2.3.46
Password	Request	section 2.2.3.122.3
DeviceInformation	Request and Response	section 2.2.3.45
Model	Request	section 2.2.3.105
IMEI	Request	section 2.2.3.82
FriendlyName	Request	section 2.2.3.74
OS	Request	section 2.2.3.117
OSLanguage	Request	section 2.2.3.118
PhoneNumber	Request	section 2.2.3.124
UserAgent	Request	section 2.2.3.174
EnableOutboundSMS	Request	section 2.2.3.57
MobileOperator	Request	section 2.2.3.103
UserInformation	Request and Response	section 2.2.3.176
Status	Response	section 2.2.3.162.14
Accounts	Response	section 2.2.3.5
Account	Response	section 2.2.3.2
AccountId	Response	section 2.2.3.3.2
AccountName	Response	section 2.2.3.4
UserDisplayName	Response	section 2.2.3.175
SendDisabled	Response	section 2.2.3.147
EmailAddresses	Response	section 2.2.3.54
SMTPAddress	Response	section 2.2.3.156
PrimarySmtpAddress	Response	section 2.2.3.127

The XML schema for the **Settings** command request is described in section 6.35. The XML schema for the **Settings** command response is described in section 6.36.

2.2.2.17 SmartForward

The **SmartForward** command is used by clients to forward messages without retrieving the full, original message from the server. The body of the request differs depending on the value of the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.4.

If the MS-ASProtocolVersion header is set to "12.1", the body of the **SmartForward** request is the MIME representation of the message to be sent. In this case, the client sets the Content-Type header to "message/rfc822".

If the MS-ASProtocolVersion header is set to "14.0" or "14.1", the body of the **SmartForward** request is WBXML data generated by using the XML elements specified in the table below.

Clients SHOULD NOT save messages directly to the local Sent Items folder; instead, clients SHOULD use the **SaveInSentItems** element (section 2.2.3.144) to automatically save the messages on the server. It is not possible to reconcile the local Sent Items folder with the server's Sent Items folder by using the **Sync** command (section 2.2.2.19). Items in the server's Sent Items folder can be added to the client by using the **Sync** command, but it is not possible to add items that are in the local Sent Items folder to the server.

The **SmartForward** command can be applied to a meeting. When **SmartForward** is applied to a recurring meeting, the **InstanceId** element (section 2.2.3.83.2) specifies the ID of a particular occurrence in the recurring meeting. If **SmartForward** is applied to a recurring meeting and the **InstanceId** element is absent, the server SHOULD forward the entire recurring meeting. If the value of the **InstanceId** element is invalid, the server responds with **Status** element (section 2.2.3.162.15) value 104, as specified in section 2.2.4.

When the **SmartForward** command is used for an appointment, the original message is included by the server as an attachment to the outgoing message. When the **SmartForward** command is used for a normal message or a meeting, the behavior of the **SmartForward** command is the same as that of the **SmartReply** command (section 2.2.2.18).

The **SmartForward** command is similar to the **SendMail** command (section 2.2.2.15), except that the outgoing message identifies the item being forwarded to and includes the text of the new message. The full text of the original message is retrieved and sent by the server. Using the server copy of the original message saves network bandwidth by not downloading the original message to the client and then uploading it again with the forward.

The **SmartForward** command lists the message recipients.

By default, because the original message and the forward messages can use different **character sets (2)**, this command will always send the outgoing message by using the **UTF-8** character set (2) for the body of the message.

The **ComposeMail** namespace is the primary namespace for this section. Elements referenced in this section that are not defined in the **ComposeMail** namespace use the namespace prefixes defined in section 2.2.1.

The following table lists the elements that are used in **SmartForward** command requests and responses.

Element name	Scope	Reference
SmartForward	Request and Response	section 2.2.3.154
ClientId	Request	section 2.2.3.28.1

Release: February 10, 2014

Element name	Scope	Reference
Source	Request	section 2.2.3.158
FolderId	Request	section 2.2.3.69
ItemId	Request	section 2.2.3.84
LongId	Request	section 2.2.3.89.4
InstanceId	Request	section 2.2.3.83.2
AccountId	Request	section 2.2.3.3.1
SaveInSentItems	Request	section 2.2.3.144
ReplaceMime	Request	section 2.2.3.135
Mime	Request	section 2.2.3.99
Status	Response	section 2.2.3.162.15

The XML schema for the **SmartForward** command request is described in section 6.37. The XML schema for the **SmartForward** command response is described in section 6.38. If the message was forwarded successfully, the server returns an empty response.

```
HTTP/1.1 200 OK
Date: Thu, 03 Sep 2009 21:05:44 GMT
Content-Length: 0
```

2.2.2.18   SmartReply

The **SmartReply** command is used by clients to reply to messages without retrieving the full, original message from the server. The body of the request differs depending on the value of the MS-ASProtocolVersion header, as specified in [MS-ASHTTP] section 2.2.1.1.2.4.

If the MS-ASProtocolVersion header is set to "12.1", the body of the **SmartReply** request is the MIME representation of the message to be sent. In this case, the client sets the Content-Type header to "message/rfc822".

If the MS-ASProtocolVersion header is set to "14.0" or "14.1", the body of the **SmartReply** request is WBXML data generated by using the XML elements specified in the table below.

Clients SHOULD NOT save messages directly to the local Sent Items folder; instead, clients SHOULD use the **SaveInSentItems** element (section 2.2.3.144) to automatically save the messages on the server. It is not possible to reconcile the local Sent Items folder with the server's Sent Items folder by using the **Sync** command (section 2.2.2.19). Items in the server's Sent Items folder can be added to the client by using the **Sync** command, but it is not possible to add items that are in the local Sent Items folder to the server.

The **SmartReply** command can be applied to a meeting. When **SmartReply** is applied to a recurring meeting, the **InstanceId** element (section 2.2.3.83.2) specifies the ID of a particular occurrence in the recurring meeting. If **SmartReply** is applied to a recurring meeting and the **InstanceId** element is absent, the server SHOULD reply for the entire recurring meeting. If the value of the **InstanceId** element is invalid, the server responds with **Status** element (section 2.2.3.162.15) value 104, as specified in section 2.2.4.

The **SmartReply** command is similar to the **SendMail** command (section 2.2.2.15), except that the outgoing message identifies the item being replied to and includes the text of the new message. The full text of the original message is retrieved and sent by the server. Using the server copy of the original message saves network bandwidth by not downloading the original message to the client and then uploading it again with the reply.

The **SmartReply** command lists the message recipients, so it is used to implement both Reply and Reply-to-All functionality. It is the responsibility of the client to implement Reply and Reply-to-All functionality.

By default, because the original message and the reply messages can use different character sets (2), this command will always send the outgoing message by using the UTF-8 character set (2) for the body of the message.

The **ComposeMail** namespace is the primary namespace for this section. Elements referenced in this section that are not defined in the **ComposeMail** namespace use the namespace prefixes defined in section 2.2.1.

The following table lists the elements that are used in **SmartReply** command requests and responses.

Element name	Scope	Reference
SmartReply	Request and Response	section 2.2.3.155
ClientId	Request	section 2.2.3.28.1
Source	Request	section 2.2.3.158
FolderId	Request	section 2.2.3.69
ItemId	Request	section 2.2.3.84
LongId	Request	section 2.2.3.89.4
InstanceId	Request	section 2.2.3.83.2
AccountId	Request	section 2.2.3.3.1
SaveInSentItems	Request	section 2.2.3.144
ReplaceMime	Request	section 2.2.3.135
Mime	Request	section 2.2.3.99
Status	Repsponse	section 2.2.3.162.15

The XML schema for the **SmartReply** command request is described in section 6.39. The XML schema for the **SmartReply** command response is described in section 6.40. If the message was sent successfully, the server returns an empty response.

```
HTTP/1.1 200 OK
Date: Thu, 03 Sep 2009 21:05:44 GMT
Content-Length: 0
```

2.2.2.19 Sync

The **Sync** command synchronizes changes in a collection between the client and the server.

The **AirSync** namespace is the primary namespace for this section. Elements referenced in this section that are not defined in the **AirSync** namespace use the namespace prefixes defined in section 2.2.1.

The following table lists the elements that are used in **Sync** command requests and responses.

Element name	Scope	Reference
Sync	Request and Response	section 2.2.3.165
Collections	Request and Response	section 2.2.3.31.2
Collection	Request and Response	section 2.2.3.29.2
SyncKey	Request and Response	section 2.2.3.166.4
CollectionId	Request and Response	section 2.2.3.30.5
Supported	Request	section 2.2.3.164
DeletesAsMoves	Request	section 2.2.3.43
GetChanges	Request	section 2.2.3.76
WindowSize	Request	section 2.2.3.183
ConversationMode	Request	section 2.2.3.36.2
Options	Request	section 2.2.3.115.5
FilterType	Request	section 2.2.3.64.2
Class	Request and Response	section 2.2.3.27.5
Conflict	Request	section 2.2.3.34
MIMESupport	Request	section 2.2.3.100.3
MIMETruncation	Request	section 2.2.3.101
MaxItems	Request	section 2.2.3.93.2
Commands	Request and Response	section 2.2.3.32
Change	Request and Response	section 2.2.3.24
ServerId	Request and Response	section 2.2.3.151.6
ApplicationData	Request and Response	section 2.2.3.11
Delete	Request and Response	section 2.2.3.42.2
Add	Request and Response	section 2.2.3.7.2
ClientId	Request and Response	section 2.2.3.28.2
Fetch	Request and Response	section 2.2.3.63.2

Element name	Scope	Reference
Wait	Request	section 2.2.3.182
HeartbeatInterval	Request	section 2.2.3.79.2
Partial	Request	section 2.2.3.121
Status	Response	section 2.2.3.162.16
Limit	Response	section 2.2.3.88
SoftDelete	Response	section 2.2.3.157
Responses	Response	section 2.2.3.141
MoreAvailable	Response	section 2.2.3.106

The XML schema for the **Sync** command request is described in section 6.41. The XML schema for the **Sync** command response is described in section 6.42.

For more details about the **AirSyncBase** elements that are used by this command, see [MS-ASAIRS] section 2.2.

Synchronization requires a priming of the system; therefore for each collection that the client wishes to synchronize, it MUST issue an initial **Sync** request by sending a synchronization key of 0 (zero). This request establishes a synchronization relationship with the server and initializes the synchronization state there. The server responds with an initial value of the synchronization key, which the client MUST then use to get the initial set of objects from the server. (From this point forward, client requests MUST always include the synchronization key that was received in the last response from the server.) The client then sends a **Sync** command request to the server with the response synchronization key and includes any changes that were made on the client.
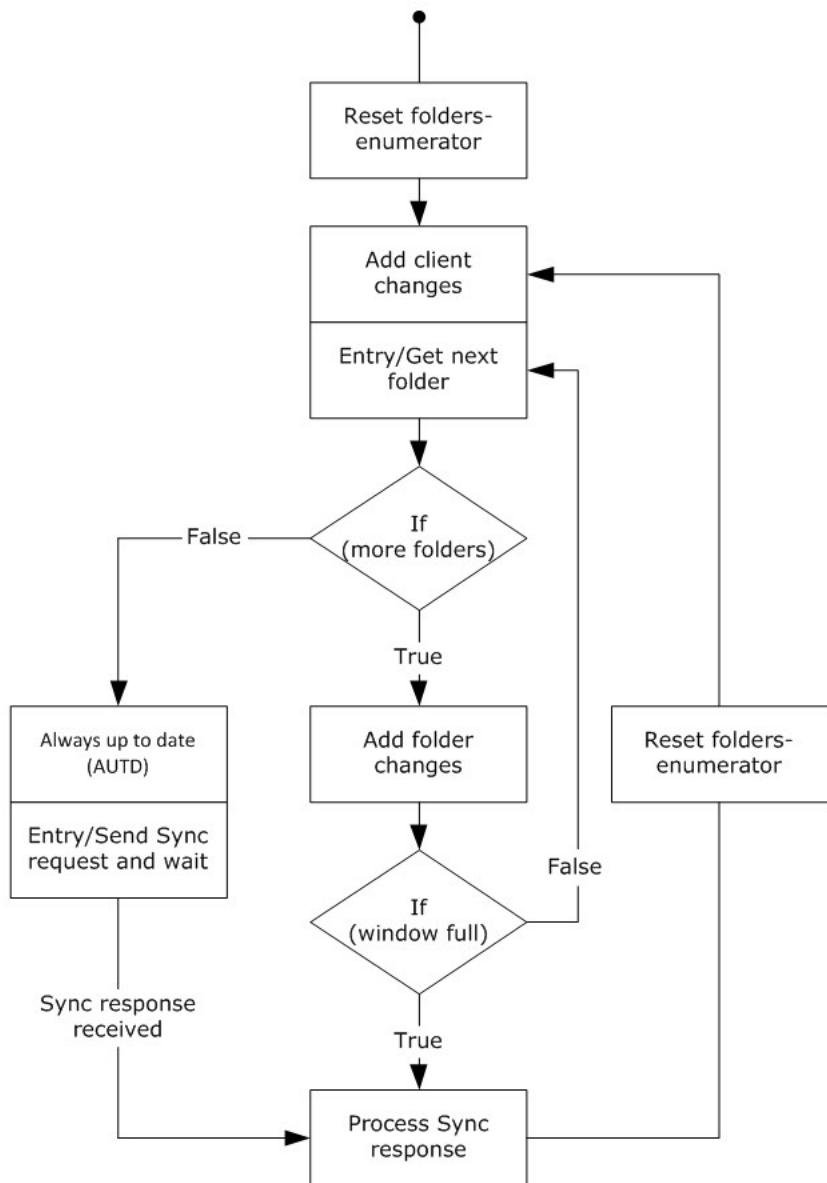
If the client device has not yet synchronized a folder, there SHOULD be no client-side changes. The device MUST synchronize the full contents of a given folder, and then have its changes, additions, and deletions applied.

The response from the server indicates whether the client's changes were accepted, and includes any changes that were made on the server. The server response also contains a synchronization key that is to be used for the next synchronization session for the folder.

This protocol has been optimized for the case in which there are no changes to any of the collections that are specified in the **Sync** request. In such a case, the client can receive an empty response from the server. After the client receives an empty response, the client can issue an empty **Sync** request. The server then re-executes the previous request, which it cached.

Certain ActiveSync classes support **ghosted** properties. A ghosted property whose value has not changed from the last **Sync** response can be excluded from the request body, and its value on the server will be preserved instead of being deleted. A client uses the **Supported** element to specify to the server which properties are managed by the client and not ghosted by the server. For more information, see section 2.2.3.164.

The following diagram shows request and response processing by the client.

Figure 2: Sync command client processing

2.2.2.19.1   Empty Sync Request

If the client sends a **Sync** request (section 2.2.2.19) with no client additions, changes, or deletes, the server caches the request. If no changes are detected on the server, the **Sync** response includes only HTTP headers, and no XML payload, and is referred to as an empty **Sync** response.

When the client receives the empty **Sync** response, if there are no pending client changes, the client in turn can send only the HTTP headers, and no XML payload in the **Sync** request to save bandwidth. This request is referred to as an empty **Sync** request. If bandwidth is not a concern, the client can send a **Sync** request with an XML payload.

Release: February 10, 2014

When the server receives the empty **Sync** request, the server assumes the request is identical to the cached request, so it retrieves the cached request and uses it. The cached request is discarded when the server receives a **Sync** request with an XML payload (a non-empty **Sync** request). This exchange of the empty **Sync** requests and responses continues until a change is detected on either the client or server. For an example empty **Sync** request and response, see section 4.5.10.

2.2.2.19.2 Empty Sync Response

The server sends a **Sync** response (section 2.2.2.19) with only HTTP headers, and no XML payload, if there are no pending server changes to report to the client. This **Sync** response is referred to as an empty **Sync** response. The client can respond to the empty **Sync** response with an empty **Sync** request if there are no pending client changes and if the client is required to conserve bandwidth; otherwise, a **Sync** request with an XML payload can be sent. For more information about empty **Sync** requests, see section 2.2.2.19.1. For an example of an empty **Sync** request and response, see section 4.5.10.

2.2.2.20 ValidateCert

The **ValidateCert** command is used by the client to validate a certificate that has been received via an S/MIME mail.

To validate a certificate, the server MUST verify that the certificate has not expired and has not been revoked. The server MUST walk up the certificate chain, verifying that each intermediate CA certificate has not expired and has not been revoked and that the root certificate is a trusted **certification authority (CA) (1)**. Certificate validation is particularly important for verifying signatures (for example, on S/MIME signed mail).

The **ValidateCert** namespace is the primary namespace for this section. Elements referenced in this section that are not defined in the **ValidateCert** namespace use the namespace prefixes defined in section 2.2.1.

The following table lists the elements that are used in **ValidateCert** command requests and responses.

Element name	Scope	Reference
ValidateCert	Request and Response	section 2.2.3.179
CertificateChain	Request	section 2.2.3.20
Certificate	Request and Response	section 2.2.3.19.2
Certificates	Request	section 2.2.3.23.2
CheckCRL	Request	section 2.2.3.26
Status	Response	section 2.2.3.162.17

The XML schema for the **ValidateCert** command request is described in section 6.43. The XML schema for the **ValidateCert** command response is described in section 6.44.

2.2.3 Elements

2.2.3.1 AcceptableResponseSchema

The **AcceptableResponseSchema** element is a required child element of the **Request** element in **Autodiscover** command requests that indicates the schema in which the server MUST send the response.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Autodiscover command request (section 2.2.2.1)	**Request** (section 2.2.3.137.1)	None	**string** ([MS-ASDTYPE] section 2.6)	1...1 (required)

The namespace MUST be "http://schemas.microsoft.com/exchange/autodiscover/mobilesync/responseschema/2006".

2.2.3.2 Account

The **Account** element<6> is a required child element of the **Accounts** element in **Settings** command responses that contains all account information associated with a single account.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command **UserInformation** response (section 2.2.2.16)	**Accounts** (section 2.2.3.5)	**AccountId** (section 2.2.3.3.2) **AccountName** (section 2.2.3.4) **UserDisplayName** (section 2.2.3.175) **SendDisabled** (section 2.2.3.147) **EmailAddresses** (section 2.2.3.54)	**container** ([MS-ASDTYPE] section 2.2)	1...N (required)

2.2.3.3 AccountId

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

- **SendMail** command requests (section 2.2.2.15)

- **SmartForward** command requests (section 2.2.2.17)

- **SmartReply** command requests (section 2.2.2.18)

- **Settings** command responses (section 2.2.2.16)

The definition of the **AccountId** element differs according to the context in which it is used. For more details, see section 2.2.3.3.1 and section 2.2.3.3.2.

2.2.3.3.1 AccountId (SendMail, SmartForward, SmartReply)

The **AccountId<7>** element is an optional child element of the **SendMail** element in **SendMail** command requests, an optional child element of the **SmartForward** element in **SmartForward** command requests, and an optional child element of the **SmartReply** element in **SmartReply** command requests. In all contexts, it identifies the account from which an email is sent.

Elements referenced in this section are defined in the **ComposeMail** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
SendMail command request (section 2.2.2.15)	**SendMail** (section 2.2.3.148)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)
SmartForward command request (section 2.2.2.17)	**SmartForward** (section 2.2.3.154)	None	**string**	0…1 (optional)
SmartReply command request (section 2.2.2.18)	**SmartReply** (section 2.2.3.155)	None	**string**	0…1 (optional)

If the **AccountId** element is not present in the request, the server sends the email using the account identified by the **settings:PrimarySmtpAddress** element (section 2.2.3.127) returned in the **Settings** command response (section 2.2.2.16).

If **AccountId** is included in the request, the value MUST equal one of the **settings:AccountId** element values included in the **Settings** command response.

The server MUST validate that the **AccountId** element value provided is valid for sending email. A **Status** element (section 2.2.3.162) value of 166 is returned if the **AccountId** element value is not valid. A **Status** element value of 167 is returned if the account does not support sending email.

Note The server sends the email using the account specified by the **AccountId** element, and not the account specified by the **From** element ([MS-ASEMAIL] section 2.2.2.29).

2.2.3.3.2 AccountId (Settings)

The **AccountId<8>** element is an optional child element of the **Account** element in **Settings** command responses (section 2.2.2.16). The value of this element identifies an account. The server MAY return this element in **Settings** command responses.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command response (section 2.2.2.16)	**Account** (section 2.2.3.2)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

The primary account, as identified by the **PrimarySmtpAddress** element (section 2.2.3.127), does not have an **AccountId** value.

2.2.3.4 AccountName

The **AccountName** element<9> is an optional child element of the **Account** element in **Settings** command responses (section 2.2.2.16). The value of this element specifies the friendly name for the given account. The server MAY return this element in **Settings** command responses.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command **UserInformation** response	**Account** (section 2.2.3.2)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

The **AccountName** element value is a **string** value up to 512 characters in length.

2.2.3.5 Accounts

The **Accounts** element<10> is an optional child element of the **Get** element in **Settings** command responses that contains all aggregate accounts that the user subscribes to.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command **UserInformation** response (section 2.2.2.16)	**Get** (section 2.2.3.75)	**Account** (section 2.2.3.2)	**container** ([MS-ASDTYPE] section 2.2)	0…1 (optional)

2.2.3.6 Action

The **Action** element is an optional child element of the **Response** element in **Autodiscover** command responses that encapsulates the server action type for the request.

All elements referenced in this section are defined in the http://schemas.microsoft.com/exchange/autodiscover/mobilesync/requestschema/2006 namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Autodiscover command response (section 2.2.2.1)	**Response** (section 2.2.3.140.1)	**Redirect** (section 2.2.3.134) **Settings** (section 2.2.3.153.1) **Error** (section 2.2.3.60)	**container** ([MS-ASDTYPE] section 2.2)	0…1 (optional)

2.2.3.7 Add

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

- **FolderSync** command responses (section 2.2.2.4)

- **Sync** command requests and responses (section 2.2.2.19)

The definition of the **Add** element differs according to the context in which it is used. For more details, see section 2.2.3.7.1 and section 2.2.3.7.2.

2.2.3.7.1 Add (FolderSync)

The **Add** element is an optional child element of the **Changes** element in **FolderSync** command responses that creates a new folder on the client. Child elements of the **Add** element specify the server ID of the folder, the server ID of the parent folder, the display name of the folder, and the type of folder.

All elements referenced in this section are defined in the **FolderHierarchy** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
FolderSync command response (section 2.2.2.4)	**Changes** (section 2.2.3.25)	**ServerId** (section 2.2.3.151.3) **ParentId** (section 2.2.3.119.2) **DisplayName** (section 2.2.3.47.3) **Type** (section 2.2.3.170.3)	**container** ([MS-ASDTYPE] section 2.2)	0...N (optional)

2.2.3.7.2 Add (Sync)

The **Add** element is an optional child element of the **Commands** element in **Sync** command requests and an optional child element of the **Commands** element and the **Responses** element in **Sync** command responses. It creates a new object in a collection on the client or on the server.

All elements referenced in this section are defined in the **AirSync** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Sync command request (section 2.2.2.19)	**Commands** (section 2.2.3.32)	**ClientId** (section 2.2.3.28.2) **ApplicationData** (section 2.2.3.11) **Class** (section 2.2.3.27.5)	**container** ([MS-ASDTYPE] section 2.2)	0...N (optional)
Sync command response	**Commands**	**ServerId** (section 2.2.3.151.6) **ApplicationData** (section 2.2.3.11)	**container**	0...N (optional)
Sync command response	**Responses** (section 2.2.3.141)	**ServerId** **ClientId** **Class** **Status** (section	**container**	0...N (optional)

Command request/response	Parent elements	Child elements	Data type	Number allowed
		2.2.3.162.16)		

When a new item is being sent from the client to the server, the **ClientId** element specifies a temporary ID for the item, which is unique on the client. The **ApplicationData** element specifies the item data. The server then responds with an **Add** element in a **Responses** element, which specifies the client ID and the server ID that was assigned to the new item.

When the client sends a **Sync** command request to the server and a new item has been added to the server collection since the last synchronization, the server responds with an **Add** element in a **Commands** element. This **Add** element specifies the server ID and data of the item to be added to the collection on the client.

When the client adds a calendar item, the **EndTime** element (as specified in [MS-ASCAL] section 2.2.2.18) MUST be present in the **ApplicationData** element. A **Status** element value of 6 is returned in the **Sync** response if the **EndTime** element is not included.

One or more **Add** elements can appear as a child of the **Commands** and **Responses** elements for a particular collection.

The **Add** element cannot be used to add any email items from the client to the server, or to modify the contents of the recipient information cache. If a client attempts to add emails to the server, a **Status** element with a value of 6 is returned as a child of the **Add** element. If a client attempts to add an item to the recipient information cache, a **Status** element with a value of 4 is returned as a child of the **Sync** element.

If the server ID in an **Add** element from the server matches the server ID for an item on the client, the client treats the addition as a change to the client item.

The server is not required to send an individual response for every operation that is sent by the client. The client only receives responses for successful additions and fetches, and failed changes and deletions. When the client does not receive a response, the client MUST assume that the operation succeeded unless informed otherwise.

2.2.3.8   airsyncbase:FileReference

The **airsyncbase:FileReference** element is an optional child element of the **Fetch** element in **ItemOperations** command requests that specifies a unique identifier that is assigned by the server to each attachment to a given item.

Elements referenced in this section are defined in the **ItemOperations** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
ItemOperations command request (section 2.2.2.8) fetch operation	**Fetch** (section 2.2.3.63.1)	None	**string** ([MS-ASDTYPE] section 2.6)	0...1 (optional)

The client MUST store the file reference for any item that is retrieved by means of the **Sync** command (section 2.2.2.19) or **Search** command (section 2.2.2.14). In an **ItemOperations** request, only one **airsyncbase:FileReference** identifier can exist per **Fetch** node. Violation of this constraint results in a **Status** element (section 2.2.3.162.7) value of 2 being returned from the

server. The client can, however, retrieve multiple attachments by using one **Fetch** node per attachment.

If the **airsyncbase:FileReference** element is present, then the **Range** element (section 2.2.3.130.1) is the only valid child element of the **Options** element (section 2.2.3.115.2).

2.2.3.9   Alias

The **Alias** element in the **GAL** namespace is an optional child element of the **search:Properties** element in **Search** command responses. The **Alias** element is only present in response to **Search** command requests that contain a **search:Name** element with a value of "GAL", as specified in section 2.2.3.110.2. It contains the alias of a recipient in the GAL that matched the search criteria from the corresponding **Search** command request.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command response (section 2.2.2.14)	**Properties** (section 2.2.3.128.2)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

2.2.3.10   And

The **And** element is an optional child element of the **Query** element in **Search** command requests. It contains elements that specify items on which to perform an AND operation.

Elements referenced in this section are defined in the **Search** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command request (section 2.2.2.14)	**Query** (section 2.2.3.129)	**FreeText** (section 2.2.3.73) **airsync:Class** (section 2.2.3.27.4) **airsync:CollectionId** (section 2.2.3.30.4) **ConversationId** (section 2.2.3.35.2) **GreaterThan** (section 2.2.3.78) **LessThan** (section 2.2.3.87)	**container** ([MS-ASDTYPE] section 2.2)	0…1 (optional)

If multiple **And** elements are included in the request, the server responds with a **Status** element (section 2.2.3.162.12) value of 8 (SearchTooComplex).

2.2.3.11   ApplicationData

The **ApplicationData** element is a required child element of the **Change** element and the **Add** element in **Sync** command requests, and a required child element of the **Change** element, the **Add** element, and the **Fetch** element in **Sync** command responses. It contains data for a particular object, such as a contact, email message, calendar appointment, or task item. The **ApplicationData** element can be used to change items, add items, or fetch items on the client device or server. The format of this data is determined by the schema for the object.

Elements referenced in this section are defined in the **AirSync** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
Sync command request (section 2.2.2.19)	**Change** (section 2.2.3.24) **Add** (section 2.2.3.7.2)	Data elements from the content classes. For details about the content classes, see [MS-ASCAL], [MS-ASCNTC], [MS-ASEMAIL], [MS-ASMS], and [MS-ASTASK].	**container** ([MS-ASDTYPE] section 2.2)	1…1 (required)
Sync command response	**Change** (section 2.2.3.24) **Add** (section 2.2.3.7.2)	Data elements from the content classes. For details about the content classes, see [MS-ASCAL], [MS-ASCNTC], [MS-ASDOC], [MS-ASEMAIL], [MS-ASMS], and [MS-ASTASK]. **rm:RightsManagementLicense** ([MS-ASRM] section 2.2.2.14)	**container**	1…1 (required)
Sync command response	**Fetch** (section 2.2.3.63.2)	**airsyncbase:Body** ([MS-ASAIRS] section 2.2.2.4) **airsyncbase:BodyPart** ([MS-ASAIRS] section 2.2.2.5) **rm:RightsManagementLicense** Data elements from the content classes. For details about the content classes, see [MS-ASCAL], [MS-ASCNTC], [MS-ASDOC], [MS-ASEMAIL], [MS-ASMS], and [MS-ASTASK].	**container**	1…1 (required)

2.2.3.12   AppliesToExternalKnown

The **AppliesToExternalKnown** element is an optional child element of the **OofMessage** element in **Settings** command requests and responses that indicates that the **OOF message** applies to known external users. A known external user is a user who is outside the sending user's organization, but is represented in the sending user's contacts.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command **Oof** request and response (section 2.2.2.16)	**OofMessage** (section 2.2.3.113)	None	None	0…1 (Choice of **AppliesToInternal** (section 2.2.3.14), **AppliesToExternalKnown**, and **AppliesToExternalUnknown** (section 2.2.3.13))

The **AppliesToExternalKnown** 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 <AppliesToExternalKnown/> tag.

When the **AppliesToExternalKnown** element is present, its peer elements (that is, the other elements within the **OofMessage** element) specify the OOF settings with regard to known external users.

The following are the peer elements of the **AppliesToExternalKnown** element:

- **Enabled** (section 2.2.3.56)—Specifies whether an OOF message is sent to this audience while the sending user is OOF.

- **ReplyMessage** (section 2.2.3.136)—Specifies the OOF reply message.

- **BodyType** (section 2.2.3.17)—Specifies the format of the OOF message.

2.2.3.13   AppliesToExternalUnknown

The **AppliesToExternalUnknown** element is an optional child element of the **OofMessage** element in **Settings** command requests and responses that indicates that the OOF message applies to unknown external users. An unknown external user is a user who is outside the sending user's organization and is not represented in the sending user's contacts.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command **Oof** request and response (section 2.2.2.16)	**OofMessage** (section 2.2.3.113)	None	None	0...1 (Choice of **AppliesToInternal** (section 2.2.3.14), **AppliesToExternalKnown** (section 2.2.3.12), and **AppliesToExternalUnknown**)

The **AppliesToExternalUnknown** 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 <AppliesToExternalUnknown/> tag.

When the **AppliesToExternalUnknown** element is present, its peer elements (that is, the other elements within the **OofMessage** element) specify the OOF settings with regard to unknown external users.

The following are the peer elements of the **AppliesToExternalUnknown** element:

- **Enabled** (section 2.2.3.56)—Specifies whether an OOF message is sent to this audience while the sending user is OOF.

- **ReplyMessage** (section 2.2.3.136)—Specifies the OOF reply message.

- **BodyType** (section 2.2.3.17)—Specifies the format of the OOF message.

2.2.3.14   AppliesToInternal

The **AppliesToInternal** element is an optional child element of the **OofMessge** element in **Settings** command requests and responses that indicates that the OOF message applies to internal users. An internal user is a user who is in the same organization as the sending user.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command **Oof** request and response (section 2.2.2.16)	**OofMessage** (section 2.2.3.113)	None	None	0…1 (Choice of **AppliesToInternal**, **AppliesToExternalKnown** (section 2.2.3.12), and **AppliesToExternalUnknown** (section 2.2.3.13))

The **AppliesToInternal** 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 <AppliesToInternal/> tag.

When the **AppliesToInternal** element is present, its peer elements (that is, the other elements within the **OofMessage** element) specify the OOF settings with regard to internal users.

The following are the peer elements of the **AppliesToInternal** element:

- **Enabled** (section 2.2.3.56)—Specifies whether an OOF message is sent to this audience while the sending user is OOF.

- **ReplyMessage** (section 2.2.3.136)—Specifies the OOF message itself.

- **BodyType** (section 2.2.3.17)—Specifies the format of the OOF message.

2.2.3.15   Autodiscover

The **Autodiscover** element is a required element in **Autodiscover** command requests responses that identifies the body of the HTTP **POST** as containing an **Autodiscover** command (section 2.2.2.1). It is the top-level element in the XML stream.

All elements referenced in this section are defined in the http://schemas.microsoft.com/exchange/autodiscover/mobilesync/requestschema/2006 namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Autodiscover command request (section 2.2.2.1)	None	**Request** (section 2.2.3.137.1)	container ([MS-ASDTYPE] section 2.2)	1…1 (required)
Autodiscover command response	None	**Response** (section 2.2.3.140.1)	container	1…1 (required)

2.2.3.16   Availability

The **Availability** element is an optional child element of the **Options** element in **ResolveRecipients** command requests that indicates to the server that free/busy data is being requested by the client and identifies the start time and end time of the free/busy data to retrieve.<11> The **Availability** element is an optional child element of the **Recipient** element in **ResolveRecipient** command responses that returns the status and free/busy data of the users or **distribution lists** identified in the request for the time identified by the **StartTime** and **EndTime** elements.

All elements referenced in this section are defined in the **ResolveRecipients** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ResolveRecipients command request (section 2.2.2.13)	**Options** (section 2.2.3.115.3)	**StartTime** (section 2.2.3.161.1) **EndTime** (section 2.2.3.58.1)	**container** ([MS-ASDTYPE] section 2.2)	0…1 (optional)
ResolveRecipients command response	**Recipient** (section 2.2.3.132)	**Status** (section 2.2.3.162.11) **MergedFreeBusy** (section 2.2.3.97)	**container**	0…1 (optional)

When the **Availability** element is included in a **ResolveRecipients** request, the server retrieves free/busy information for the users identified in the **To** elements included in the request, and returns the free/busy information in the **MergedFreeBusy** element in the response. If the **Availability** element is included in the **ResolveRecipients** request, the request MUST also include a valid **StartTime** element value and **EndTime** element value. When the server parses the request, the server first resolves the recipients identified by the **To** elements, and then determines the users free/busy information for the specified time span, before returning the free/busy data in the **MergedFreeBusy** element.

2.2.3.17 BodyType

The **BodyType** element is a required child element of the **Get** element in **Settings** command **Oof** requests. It is a required child element of the **OofMessage** element in **Settings** command requests and responses if a **ReplyMessage** element (section 2.2.3.136) is present as a child of the **OofMessage** element. In all contexts, it specifies the format of the OOF message.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command **Oof Get** request (section 2.2.2.16)	**Get** (section 2.2.3.75)	None	**string** ([MS-ASDTYPE] section 2.6)	1…1 (required)
Settings command **Oof Set** request	**OofMessage** (section 2.2.3.113)	None	**string**	1…1 (required) if a **ReplyMessage** element is present; otherwise, 0…1 (optional)
Settings command **Oof** response	**OofMessage**	None	**string**	1…1 (required) if a **ReplyMessage** element is present; otherwise, 0…1 (optional)

The following are the permitted values for the **BodyType** element:

- Text

- HTML

If the **BodyType** element has the value "HTML", all message strings are sent in the **HTML** format. If the **BodyType** element has the value "Text", the message strings are sent in **plain text**. Because there is no default value, the **BodyType** element MUST be present (with a non-NULL value) in a

Settings command **Oof Get** operation request and MUST be present (with a non-NULL value) in a **Settings** command **Oof Set** operation request if the **ReplyMessage** element (section 2.2.3.136) has been set, and MUST always be present (with a non-NULL value) in a **Settings** command **Oof Get** operation response if the **ReplyMessage** element is returned.

2.2.3.18 CalendarId

The **CalendarId** element is an optional child element of the **Result** element in **MeetingResponse** command responses that specifies the server ID of the calendar item. The **CalendarId** element value can be up to 64 characters in length.

All elements referenced in this section are defined in the **MeetingResponse** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
MeetingResponse command response (section 2.2.2.9)	**Result** (section 2.2.3.142.1)	None	**string** ([MS-ASDTYPE] section 2.6)	0...1 (optional)

The **CalendarId** element is included in the **MeetingResponse** command response that is sent to the client if the meeting request was not declined. If the meeting is accepted or tentatively accepted, the server adds a new item to the calendar and returns its server ID in the **CalendarId** element in the response. If the client created a tentative meeting calendar item, the client updates that item with the new server ID. The client also changes the busy status from tentative to busy. When a meeting is accepted, the server also creates a new calendar item with the same server ID. This means there is a conflict that will be resolved the next time the calendar is synchronized.

If the meeting is declined, the response does not contain a **CalendarId** element.

2.2.3.19 Certificate

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

▪ **ResolveRecipients** command responses (section 2.2.2.13)

▪ **ValidateCert** command requests and responses (section 2.2.2.20)

The definition of the **Certificate** element differs according to the context in which it is used. For more details, see section 2.2.3.19.1 and section 2.2.3.19.2.

2.2.3.19.1 Certificate (ResolveRecipients)

The **Certificate** element is an optional child element of the **Certificates** element in **ResolveRecipients** command responses that contains the X509 certificate **binary large object (BLOB)** that is encoded with base64 encoding.

All elements referenced in this section are defined in the **ResolveRecipients** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ResolveRecipients command response (section 2.2.2.13)	**Certificates** (section 2.2.3.23.1)	None	**string** ([MS-ASDTYPE] section 2.6) (encoded with base64 encoding)	0...N (optional)

This element is returned by the server only if the client specifies a value of 2 in the **CertificateRetrieval** element (section 2.2.3.22) in the **ResolveRecipients** command request.

2.2.3.19.2 Certificate (ValidateCert)

The **Certificate** element is a required child element of the **Certificates** element and the **CertificateChain** element in **ValidateCert** command requests that contains the X509 certificate BLOB that is encoded with base64 encoding. The **Certificate** element is an optional child element of the **ValidateCert** element in **ValidateCert** command responses that contains the status of the certificate validation.

All elements referenced in this section are defined in the **ValidateCert** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ValidateCert command request (section 2.2.2.20)	**Certificates** (section 2.2.3.23.2) **CertificateChain** (section 2.2.3.20)	None.	**string** ([MS-ASDTYPE] section 2.6) (encoded with base64 encoding)	1...N (required)
ValidateCert command response	**ValidateCert** (section 2.2.3.179)	**Status** (section 2.2.3.162.17)	**container** ([MS-ASDTYPE] section 2.2)	0...N (optional)

2.2.3.20 CertificateChain

The **CertificateChain** element is an optional child element of the **ValidateCert** element in **ValidateCert** command requests that contains the list of certificates to be validated.

All elements referenced in this section are defined in the **ValidateCert** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ValidateCert command request (section 2.2.2.20)	**ValidateCert** (section 2.2.3.179)	**Certificate** (section 2.2.3.19.2)	**container** ([MS-ASDTYPE] section 2.2)	0...1 (optional)

2.2.3.21 CertificateCount

The **CertificateCount** element is an optional child element of the **Certificates** element in **ResolveRecipients** command responses that specifies the number of valid certificates that were found for the recipient.

All elements referenced in this section are defined in the **ResolveRecipients** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ResolveRecipients command response (section 2.2.2.13)	**Certificates** (section 2.2.3.23.1)	None	**integer** ([MS-ASDTYPE] section 2.5)	0...1 (optional) per **Certificates** parent element

If a **Status** element (section 2.2.3.162.11) value of 8 is returned with the **Certificates** element, the **CertificateCount** element specifies the number of recipient certificates that were not returned.

2.2.3.22 CertificateRetrieval

The **CertificateRetrieval** element is an optional child element of the **Options** element in **ResolveRecipients** command requests that specifies whether S/MIME certificates are returned by the server for each resolved recipient.

All elements referenced in this section are defined in the **ResolveRecipients** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ResolveRecipients command request (section 2.2.2.13)	**Options** (section 2.2.3.115.3)	None	**integer** ([MS-ASDTYPE] section 2.5)	0...1 (optional)

The following table shows valid values for the **CertificateRetrieval** element.

Value	Meaning
1	Do not retrieve certificates for the recipient (default).
2	Retrieve the full certificate for each resolved recipient.
3	Retrieve the mini certificate for each resolved recipient.

2.2.3.23 Certificates

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

- **ResolveRecipients** command responses (section 2.2.2.13)

- **ValidateCert** command requests (section 2.2.2.20)

The definition of the **Certificates** element differs according to the context in which it is used. For more details, see section 2.2.3.23.1 and section 2.2.3.23.2.

2.2.3.23.1 Certificates (ResolveRecipients)

The **Certificates** element is an optional child element of the **Recipient** element in **ResolveRecipients** command responses that contains information about the certificates for a recipient.

All elements referenced in this section are defined in the **ResolveRecipients** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ResolveRecipients command response (section 2.2.2.13)	**Recipient** (section 2.2.3.132)	**Status** (section 2.2.3.162.11) **CertificateCount** (section 2.2.3.21) **RecipientCount**	**container** ([MS-ASDTYPE] section 2.2)	0...1 (optional)

Command request/response	Parent elements	Child elements	Data type	Number allowed
		(section 2.2.3.133) **Certificate** (section 2.2.3.19) **MiniCertificate** (section 2.2.3.102)		

Although multiple **Certificates** elements can be included in a response, only one **Certificates** element is allowed per **Recipient** parent element.

2.2.3.23.2 Certificates (ValidateCert)

The **Certificates** element is a required child element of the **ValidateCert** element in **ValidateCert** command requests that contains the list of certificates to be validated.

All elements referenced in this section are defined in the **ValidateCert** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ValidateCert command request (section 2.2.2.20)	**ValidateCert** (section 2.2.3.179)	**Certificate** (section 2.2.3.19.2)	**container** ([MS-ASDTYPE] section 2.2)	1…1 (required)

2.2.3.24 Change

The **Change** element is an optional child element of the **Commands** element in **Sync** command requests that modifies properties of an existing object on the client device or the server. The **Change** element is an optional child element of the **Commands** element and the **Responses** element in **Sync** command responses that contains properties of an existing object on the client device or the server that were modified. The changed object is identified by its **ServerId** element.

Elements referenced in this section are defined in the **AirSync** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
Sync command request (section 2.2.2.19)	**Commands** (section 2.2.3.32)	**ServerId** (section 2.2.3.151.6) **ApplicationData** (section 2.2.3.11)	**container** ([MS-ASDTYPE] section 2.2)	0…N (optional)
Sync command response	**Commands**	**ServerId** **ApplicationData** **Class** (section 2.2.3.27.5)	**container**	0…N (optional)
Sync command response	**Responses** (section 2.2.3.141)	**ServerId** **Class** **Status** (section 2.2.3.162.16)	**container**	0…N (optional)

In a **Sync** command request or response, one or more **Change** elements can appear as a child element of the **Commands** element for a particular collection.

Certain in-schema properties remain untouched in the following three cases:

▪ If there is only an **email:Flag** ([MS-ASEMAIL] section 2.2.2.27), **email:Read** ([MS-ASEMAIL] section 2.2.2.47), or **email:Categories** ([MS-ASEMAIL] section 2.2.2.9) change (that is, if only an **email:Flag**, **email:Categories** or **email:Read** element is present), all other properties will remain unchanged and the client SHOULD NOT send the other elements in the request. If all the other elements are sent, extra bandwidth is used, but no errors occur.

▪ If a **calendar:Exceptions** ([MS-ASCAL] section 2.2.2.20) node is not specified, the properties for that **calendar:Exceptions** node will remain unchanged. If a **calendar:Exception** ([MS-ASCAL] section 2.2.2.19) node within the **calendar:Exceptions** node is not present, that particular exception will remain unchanged.

▪ If the **airsyncbase:Body**, **airsyncbase:Data**, or **contacts:Picture** elements are not present, the corresponding properties will remain unchanged.

In all other cases, if an in-schema property is not specified in a change request, the property is actively deleted from the item on the server. A client MUST be aware of this when it is sending **Sync** requests; otherwise, data can be unintentionally removed.

2.2.3.25 Changes

The **Changes** element is an optional child element of the **FolderSync** element in **FolderSync** command responses that contains changes to the folder hierarchy. It is used to update the client with folder additions, deletions, and updates on the server.

All elements referenced in this section are defined in the **FolderHierarchy** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
FolderSync command response (section 2.2.2.4)	**FolderSync** (section 2.2.3.71)	**Count** (section 2.2.3.37) **Update** (section 2.2.3.171) **Delete** (section 2.2.3.42.1) **Add** (section 2.2.3.7.1)	**container** ([MS-ASDTYPE] section 2.2)	0…1 (optional)

The server SHOULD maintain the same set of folder data being returned across synchronization key 0 (zero), in terms of **ServerId** (section 2.2.3.151.3) and **DisplayName** (section 2.2.3.47.3) mapping. While the server SHOULD maintain this mapping, clients MUST be able to handle having the server return a completely different set of mapped data.

2.2.3.26 CheckCRL

The **CheckCRL** element is an optional child element of the **ValidateCert** element in **ValidateCert** command requests that specifies whether the server ignores an unverifiable revocation status.

All elements referenced in this section are defined in the **ValidateCert** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ValidateCert command request (section 2.2.2.20)	**ValidateCert** (section 2.2.3.179)	None	**integer** ([MS-ASDTYPE] section 2.5)	0…1 (optional)

The revocation status of a certificate cannot be verified when the **certificate revocation lists (CRLs)** cannot be retrieved.

When the **CheckCRL** element value is set to 1 (**TRUE**), the server MUST NOT ignore an unverifiable revocation status. When the **CheckCRL** element value is set to 0 (**FALSE**), the server SHOULD ignore an unverifiable revocation status. The default value is 0 (zero).

2.2.3.27   Class

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

- **GetItemEstimate** command requests (section 2.2.2.7)

- **ItemOperations** command responses (section 2.2.2.8)

- **Ping** command requests (section 2.2.2.11)

- **Search** command requests (section 2.2.2.14) and responses

- **Sync** command requests and responses (section 2.2.2.19)

The definition of the **Class** element differs according to the context in which it is used. For more details, see section 2.2.3.27.1, section 2.2.3.27.2, section 2.2.3.27.3, section 2.2.3.27.4, and section 2.2.3.27.5.

2.2.3.27.1   Class (GetItemEstimate)

The **airsync:Class** element<12> is an optional child element of the **airsync:Options** element in **GetItemEstimate** command requests that assigns the filters within the **airsync:Options** container to a given class.

Command request/response	Parent elements	Child elements	Data type	Number allowed
GetItemEstimate command request (section 2.2.2.7)	**airsync:Options** (section 2.2.3.115.1)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

Options for the same class within the same collection MUST NOT be redefined. The **airsync:Class** element is not necessary for the default items contained within the collection (contacts in a Contacts folder for example).

For example, to sync SMS messages, include an **airsync:Options** node with class "SMS". To also sync email messages at the same time, include another **airsync:Options** node with class "Email". Only SMS messages and email messages can be synchronized at the same time; a request for any other combination of classes will fail with a status value of 4.The valid **airsync:Class** element values are:

- Tasks

- Email

- Calendar

- Contacts

- SMS

2.2.3.27.2   Class (ItemOperations)

The **airsync:Class** element is an optional child element of the **Fetch** element in **ItemOperations** command responses that indicates the class of the content of the fetched item. This element is defined in the **AirSync** namespace.

Elements referenced in this section are defined in the **ItemOperations** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
ItemOperations command response (section 2.2.2.8) fetch operation	**Fetch** (section 2.2.3.63.1)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

The valid **airsync:Class** element values are:

- Email

- Contacts

- Calendar

- Tasks

2.2.3.27.3   Class (Ping)

The **Class** element is a required child element of the **Folder** element in **Ping** command requests that specifies the content class of the folder to be monitored.

All elements referenced in this section are defined in the **Ping** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Ping command request (section 2.2.2.11)	**Folder** (section 2.2.3.66)	None	**string** ([MS-ASDTYPE] section 2.6)	1…1 (required)

The valid **Class** element values are:

- Email

- Calendar

- Contacts

- Tasks

▪ Notes<13>

2.2.3.27.4   Class (Search)

The **airsync:Class** element is an optional child element of the **And** element and the **Schema** element in **Search** command requests and a required child element of the **Result** element in **Search** command responses. In all contexts, it specifies the classes that the client wants returned for a given collection.

Elements referenced in this section are defined in the **Search** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command request (section 2.2.2.14)	**And** (section 2.2.3.10) **Schema**	None	**string** ([MS-ASDTYPE] section 2.6)	0…N (optional)
Search command response	**Result** (section 2.2.3.142.2)	None	**string**	1…1 (required, 1 per **Result** element)

The valid **airsync:Class** element values are:

▪ Tasks

▪ Email

▪ Calendar

▪ Contacts

▪ Notes

▪ SMS<14>

The **Search** request can include one or more **airsync:Class** elements in the request to limit the type of data included in the **Search** response.

If one or more **airsync:Class** elements are not included in the **Search** request, the server will return all supported classes.

If the **airsync:Class** element is included as a child of any element other than the **And** element, the server responds with a **Status** element (section 2.2.3.162.12) value of 8 (SearchTooComplex).

2.2.3.27.5   Class (Sync)

The **Class** element is an optional child element of the **Add** element or the **Options** element in **Sync** command requests. The **Class** element is an optional child element of the **Delete** element, the **Change** element, or the **Add** element in **Sync** command responses.

All elements referenced in this section are defined in the **AirSync** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Sync command request (section 2.2.2.19)	**Add** (section 2.2.3.7.2) **Options** (section 2.2.3.115.5)	None	**string** ([MS-ASDTYPE] section 2.6)	0...1 (optional)
Sync command response	**Delete**<15> (section 2.2.3.42.2) **Change** (section 2.2.3.24) **Add** (section 2.2.3.7.2)	None	**string**	0...1 (optional)

As a child element of the **Add** element in the **Sync** command request, the **Class** element<16> identifies the class of the item being added to the collection.

As a child element of the **Options** element in the **Sync** command request, the **Class** element<17> assigns the filters within the **Options** container to a given class. Options for the same **Class** within the same **Collection** (section 2.2.3.29.2) MUST NOT be redefined. A **Status** element (section 2.2.3.162.16) value of 4 is returned if options for the same **Class** within the same **Collection** are redefined. The **Class** element is not necessary for the default items contained within the collection (contacts in a contacts folder for example). For example, to sync SMS messages, include an **Options** node with class "SMS". To also sync email messages at the same time, include another **Options** node with class "Email". Only SMS messages and email messages can be synchronized at the same time; a request for any other combination of classes will fail with a status value of 4. The result of including more than one **Class** element as child elements of the **Options** element is undefined. The server MAY return a protocol status error in response to such a command request.

As a child element of the **Delete** element in the **Sync** command response, the **Class** element<18> identifies the class of the item being deleted from the collection.

As a child element of the **Change** element in the **Sync** command response, the **Class** element<19> identifies the class of the item being changed in the collection.

As a child element of the **Add** element in the **Sync** command response, the **Class** element<20> identifies the class of the item being added to the collection. The **Class** element is not included in **Sync Add** responses when the class of the collection matches the item class. For example, a **Sync** command response for an email added to the Inbox folder would not include a **Class** element value of "Email" as the Inbox contains email class content by default. However, a **Sync** command response for an SMS message added to the Inbox does include the **Class** value as SMS items are not the default class type of the Inbox.

In all contexts of a **Sync** command request or **Sync** command response, the valid **Class** element values are:

- Tasks

- Email

- Calendar

- Contacts

- SMS

▪ Notes

2.2.3.28 ClientId

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

▪ **SendMail** command requests (section 2.2.2.15)

▪ **SmartForward** command requests (section 2.2.2.17)

▪ **SmartReply** command requests (section 2.2.2.18)

▪ **Sync** command requests and responses (section 2.2.2.19)

The definition of the **ClientId** element differs according to the context in which it is used. For more details, see section 2.2.3.28.1 and section 2.2.3.28.2.

2.2.3.28.1 ClientId (SendMail, SmartForward, and SmartReply)

The **ClientId** element is a required child element of the **SendMail** element in **SendMail** command requests, a required child element of the **SmartForward** element in **SmartForward** command requests, and a required child element of the **SmartReply** element in **SmartReply** command requests. In all contexts, it specifies the client's unique message ID (MID).

All elements referenced in this section are defined in the **ComposeMail** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
SendMail command request (section 2.2.2.15)	**SendMail** (section 2.2.3.148)	None	**string** ([MS-ASDTYPE] section 2.6)	1…1 (required)
SmartForward command request (section 2.2.2.17)	**SmartForward** (section 2.2.3.154)	None	**string**	1…1 (required)
SmartReply command request (section 2.2.2.18)	**SmartReply** (section 2.2.3.155)	None	**string**	1…1 (required)

The **ClientId** element value can be up to 40 characters in length and MUST be unique for each message, as the server will use the **ClientId** element value to identify duplicate messages. The **ClientId** element value can be a simple counter incremented for each new message.

2.2.3.28.2 ClientId (Sync)

The **ClientId** element is a required child element of the **Add** element in **Sync** command requests and responses that contains a unique identifier (typically an integer) that is generated by the client to temporarily identify a new object that is being created by using the **Add** element. The client includes the **ClientId** element in the **Add** element request that it sends to the server. The server response contains an **Add** element that contains the original client ID and a new server ID that was assigned for the object, which replaces the client ID as the permanent object identifier.

All elements referenced in this section are defined in the **AirSync** namespace.

Release: February 10, 2014

Command request/response	Parent elements	Child elements	Data type	Number allowed
Sync command request (section 2.2.2.19)	**Add** (section 2.2.3.7.2)	None	**string** ([MS-ASDTYPE] section 2.6)	1…1 (required)
Sync command response	**Add** (section 2.2.3.7.2)	None	**string**	1…1 (required)

The **ClientId** element is a unique identifier that consists of up to 64 digits and letters. The client generates this ID. The value only has to be unique for the device during the duration of the Sync request that adds the object to the server. The client stores the client IDs until the synchronization session is completed successfully, to make recovery easier if the synchronization process fails.

An easy way to implement the client ID is to use a counter that is incremented for each new object that is created on the client.

2.2.3.29 Collection

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

- **GetItemEstimate** command requests and responses (section 2.2.2.7)

- **Sync** command requests and responses (section 2.2.2.19)

The definition of the **Collection** element differs according to the context in which it is used. For more details, see section 2.2.3.29.1 and section 2.2.3.29.2.

2.2.3.29.1 Collection (GetItemEstimate)

The **Collection** element is a required child element of the **Collections** element in **GetItemEstimate** command requests and an optional child element of the **Response** element in **GetItemEstimate** command responses. In both contexts, it contains elements that apply to a particular collection. A maximum of 300 **Collection** elements can be included in a single **Collections** element.

Elements referenced in this section are defined in the **GetItemEstimate** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
GetItemEstimate command request (section 2.2.2.7)	**Collections** (section 2.2.3.31.1)	**airsync:SyncKey** (section 2.2.3.166.3) **CollectionId** (section 2.2.3.29.1) **airsync:ConversationMode** (section 2.2.3.36.1) **airsync:Options** (section 2.2.3.115.1)	**container** ([MS-ASDTYPE] section 2.2)	1…300 (required)
GetItemEstimate command response	**Response** (section 2.2.3.140.2)	**CollectionId** (section 2.2.3.30.1) **Estimate** (section 2.2.3.62)<21>	**container**	0…1 (optional)

Release: February 10, 2014

2.2.3.29.2 Collection (Sync)

The **Collection** element is a required child element of the **Collections** element in **Sync** command requests, and an optional child element in **Sync** command responses. In both contexts, it contains commands and options that apply to a particular collection.

All elements referenced in this section are defined in the **AirSync** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Sync command request (section 2.2.2.19)	**Collections** (section 2.2.3.31.2)	**SyncKey** (section 2.2.3.166.4) **Supported** (section 2.2.3.164) **CollectionId** (section 2.2.3.30.5) **DeletesAsMoves** (section 2.2.3.43) **GetChanges** (section 2.2.3.76) **WindowSize** (section 2.2.3.183) **Options** (section 2.2.3.115.5) **ConversationMode** (section 2.2.3.36.2) **Commands** (section 2.2.3.32)	**container** ([MS-ASDTYPE] section 2.2)	1…N (required)
Sync command response	**Collections**	**SyncKey** **CollectionId** **Status** (section 2.2.3.162.16) **MoreAvailable** (section 2.2.3.106) **Commands** **Responses** (section 2.2.3.141)	**container**	0…N (optional)

The **Collection** element contains identification information (**CollectionId**), synchronization state (**SyncKey**), commands (**GetChanges**, **Commands**), and options (**WindowSize**, **Options**, **DeletesAsMoves**, **MoreAvailable**).

There is a strict ordering of the XML elements within a **Collection** node in a **Sync** command request. The order is as follows:

1. **SyncKey**

2. **CollectionId**

3. **Supported**

4. **DeletesAsMoves**

5. **GetChanges**

6. **WindowSize**

7. **ConversationMode**

8. **Options**

9. **Commands**

A single **Collections** element can contain multiple **Collection** elements. Therefore, each **Collection** does not require its own **Sync** command. That is, a **Sync** command request can specify multiple collections to be synchronized.

2.2.3.30   CollectionId

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

- **GetItemEstimate** command requests and responses (section 2.2.2.7)

- **ItemOperations** command requests and responses (section 2.2.2.8)

- **MeetingResponse** command requests (section 2.2.2.9)

- **Search** command requests and responses (section 2.2.2.14)

- **Sync** command requests and responses (section 2.2.2.19).

The definition of the **CollectionId** element differs according to the context in which it is used. For more details, see section 2.2.3.30.1, section 2.2.3.30.2, section 2.2.3.30.3, section 2.2.3.30.4, and section 2.2.3.30.5.

2.2.3.30.1   CollectionId (GetItemEstimate)

The **CollectionId** element is a required child element of the **Collection** element in **GetItemEstimate** command requests and responses that specifies the server ID of the collection from which the item estimate is being obtained.

Elements referenced in this section are defined in the **GetItemEstimate** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
GetItemEstimate command request (section 2.2.2.7)	**Collection** (section 2.2.3.29.1)	None	**string** ([MS-ASDTYPE] section 2.6)	1...1 (required)
GetItemEstimate command response	**Collection**	None	**string**	1...1 (required)

The **CollectionId** element value can be up to 64 characters in length. The **CollectionId** element value sent in the **GetItemEstimate** command request corresponds to the Server ID value assigned to the folder containing the item to retrieve. The collection ID is obtained from the **folderhierarchy:ServerId** element (section 2.2.3.151) of a previous **FolderSync** command response (section 2.2.2.4) or **FolderCreate** command response (section 2.2.2.2). (The client

SHOULD store **folderhierarchy:ServerId** values as they are returned by the server in **FolderSync** and **FolderCreate** command responses.)

2.2.3.30.2 CollectionId (ItemOperations)

The **airsync:CollectionId** element is a child element of the **EmptyFolderContents** element and the **Fetch** element in **ItemOperations** command requests that enables a client to specify the folder to be emptied or the item to be fetched. The **airsync:CollectionId** element is a child element of the **EmptyFolderContents** element and the **Fetch** element in **ItemOperations** command responses that identifies the folder that was emptied or the folder that was fetched.

Elements referenced in this section are defined in the **ItemOperations** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
ItemOperations command request (section 2.2.2.8)	**EmptyFolderContents** (section 2.2.3.55) **Fetch** (section 2.2.3.63.1)	None	**string** ([MS-ASDTYPE] section 2.6)	1…1 (required in **EmptyFolderContents** operation) 0…1 (optional in fetch operation)
ItemOperations command response	**EmptyFolderContents** **Fetch**	None	**string**	1…1 (required in **EmptyFolderContents** operation) 0…1 (optional in fetch operation)

2.2.3.30.3 CollectionId (MeetingResponse)

The **CollectionId** element is a child element of the **Request** element in **MeetingResponse** command requests that specifies the folder that contains the meeting request.

Elements referenced in this section are defined in the **MeetingResponse** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
MeetingResponse command request (section 2.2.2.9)	**Request** (section 2.2.3.137.2)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (Required, or optional if **search:LongId** (section 2.2.3.89.2) is included)

Because meeting requests are most commonly sent to the Inbox folder, the **CollectionId** element value that specifies the Inbox folder is the most common value for this element.

The **CollectionId** element value can be up to 64 characters in length and is obtained from the **folderhierarchy:ServerId** element (section 2.2.3.151) of a previous **FolderSync** command response (section 2.2.2.4) or **FolderCreate** command response (section 2.2.2.2).

2.2.3.30.4 CollectionId (Search)

The **airsync:CollectionId** element is an optional child element of the **And** element in **Search** command requests that specifies the folder in which to search. The **airsync:CollectionId** element

is an optional child element of the **Result** element in **Search** command responses that specifies the folder in which the item was found.

Elements referenced in this section are defined in the **Search** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command request (section 2.2.2.14)	**And** (section 2.2.3.10)	None	**string** ([MS-ASDTYPE] section 2.6)	0...N (optional)
Search command response	**Result** (section 2.2.3.142.2)	None	**string**	0...N (optional)

In a **Search** command request, multiple folders can be specified by including multiple **airsync:CollectionId** elements. If the **DeepTraversal** element (section 2.2.3.41) is present, it applies to all folders under each **airsync:CollectionId** element. If the **airsync:CollectionId** element is included as a child of any element other than **And**, the server responds with a **Status** element (section 2.2.3.162.12) value of 8 (SearchTooComplex).

2.2.3.30.5   CollectionId (Sync)

The **CollectionId** element is a required child element of the **Collection** element in **Sync** command requests and responses that specifies the server ID of the folder to be synchronized.

Elements referenced in this section are defined in the **AirSync** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
Sync command request (section 2.2.2.19)	**Collection** (section 2.2.3.29.2)	None	**string** ([MS-ASDTYPE] section 2.6)	1...1 (required)
Sync command response	**Collection** (section 2.2.3.29.2)	None	**string**	1...1 (required)

The server ID of the folder is obtained from the **folderhierarcy:ServerId** element (section 2.2.3.151) of a previous **FolderSync** command response (section 2.2.2.4) or **FolderCreate** command response (section 2.2.2.2). The **CollectionId** element value can be up to 64 characters in length.

The **CollectionId** value "RI" specifies the recipient information cache, which contains a list of contacts that the user has interacted with most often in the near term, and with whom the user is likely to interact again. The recipient information cache is a **special folder** (similar to the Inbox folder), which limits the operations that can be performed on it.

2.2.3.31   Collections

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

▪   **GetItemEstimate** command requests (section 2.2.2.7)

- **Sync** command requests and responses (section 2.2.2.19)

The definition of the **Collections** element differs according to the context in which it is used. For more details, see section 2.2.3.31.1 and section 2.2.3.31.2.

2.2.3.31.1 Collections (GetItemEstimate)

The **Collections** element is a required child element of the **GetItemEstimate** element in **GetItemEstimate** command requests that serves as a container for one to 300 **Collection** elements.

All elements referenced in this section are defined in the **GetItemEstimate** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
GetItemEstimate command request (section 2.2.2.7)	**GetItemEstimate** (section 2.2.3.77)	**Collection** (section 2.2.3.29.1)	container ([MS-ASDTYPE] section 2.2)	1…1 (required)

2.2.3.31.2 Collections (Sync)

The **Collections** element is an optional child element of the **Sync** element in **Sync** command requests and responses that serves as a container for the **Collection** element.

All elements referenced in this section are defined in the **AirSync** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Sync command request (section 2.2.2.19)	**Sync** (section 2.2.3.165)	**Collection** (section 2.2.3.29.2)	**container** ([MS-ASDTYPE] section 2.2)	0…1 (optional)
Sync command response	**Sync**	**Collection**	container	0…1 (optional)

The **Collections** element is optional. If the **Collections** element is present, it can contain multiple **Collection** elements.

2.2.3.32 Commands

The **Commands** element is an optional child element of the **Collection** element in **Sync** command requests and responses that contains operations that apply to a collection. Available operations are add, delete, change, fetch, and soft delete. Client operations are sent in the **POST** request; server commands are sent in the **POST** response.

This element is optional. If it is present, it MUST include at least one operation. It is a child of the **Collection** element.

All elements referenced in this section are defined in the **AirSync** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Sync command request	**Collection**	**Add** (section	**container** ([MS-	0…1

Command request/response	Parent elements	Child elements	Data type	Number allowed
(section 2.2.2.19)	(section 2.2.3.29.2)	2.2.3.7.2) **Delete** (section 2.2.3.42.2) **Change** (section 2.2.3.24) **Fetch** (section 2.2.3.63.2)	ASDTYPE] section 2.2)	(optional)
Sync command response	**Collection**	**Add** **Delete** **Change** **SoftDelete** (section 2.2.3.157)	**container**	0…1 (optional)

The result of including no child elements for the **Commands** element is undefined. The server MAY return a protocol status error in response to such a command request. See section 3.1.5.4 for information regarding the number of child elements contained in a **Commands** element.

The **Commands** element can appear in both Sync requests and responses.

2.2.3.33   Company

The **Company** element in the **GAL** namespace is an optional child element of the **search:Properties** element in **Search** command responses. The **Company** element is only present in response to **Search** command requests that contain a **search:Name** element with a value of "GAL", as specified in section 2.2.3.110.2. It contains the company of a recipient in the GAL that matched the search criteria from the corresponding **Search** command request.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command response (section 2.2.2.14)	**Properties** (section 2.2.3.128.2)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

2.2.3.34   Conflict

The **Conflict** element is an optional child element of the **Options** element in **Sync** command requests that specifies how to resolve the conflict that occurs when an object has been changed on both the client and the server. The value specifies which object—the client object or the server object—to keep if there is a conflict.

All elements referenced in this section are defined in the **AirSync** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Sync command request (section 2.2.2.19)	**Options** (section 2.2.3.115.5)	None	**unsignedByte** ([MS-ASDTYPE] section 2.7)	0…1 (optional)

The following table lists valid values for the element.

Value	Meaning
0	Client object replaces server object.
1	Server object replaces client object.

If the **Conflict** element is not present, the server object will replace the client object when a conflict occurs.

A value of 0 (zero) means to keep the client object; a value of 1 means to keep the server object. If the value is 1 and there is a conflict, a **Status** element (section 2.2.3.162.16) value of 7 is returned to inform the client that the object that the client sent to the server was discarded.

The **Conflict** element applies to the entire collection; therefore, it is not possible to apply the **Conflict** element value to individual items within the collection.

The **Conflict** element is a child of the **Options** element, and therefore the **Conflict** element appears only in requests to the server from the client.

The result of including more than one **Conflict** element as the child of an **Options** element is undefined. The server MAY return a protocol status error in response to such a command request.

If a **Delete** element (section 2.2.3.42.2) conflicts with an **Add** (section 2.2.3.7.2) or **Change** element (section 2.2.3.24), the **Delete** element takes precedence.

2.2.3.35   ConversationId

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

- **ItemOperations** command requests and responses (section 2.2.2.8)

- **Search** command requests (section 2.2.2.14)

The definition of the **ConversationId** element differs according to the context in which it is used. For more details, see section 2.2.3.35.1 and section 2.2.3.35.2.

2.2.3.35.1   ConversationId (ItemOperations)

The **ConversationId** element is a required child element of the **Move** element in **ItemOperations** command requests and responses that specifies the conversation to be moved.<22>

All elements referenced in this section are defined in the **ItemOperations** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ItemOperations command request (section 2.2.2.8) **Move** operation	**Move** (section 2.2.3.107.1)	None	**string** ([MS-ASDTYPE] section 2.6)	1…1 (required)
ItemOperations command response **Move** operation	**Move**	None	**string**	1…1 (required)

2.2.3.35.2 ConversationId (Search)

The **ConversationId** element is an optional child element of the **And** element in **Search** command requests that specifies the conversation for which to search.<23> The value is a **GUID**. For details, see [MS-ASCON].

All elements referenced in this section are defined in the **Search** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command request (section 2.2.2.14)	**And** (section 2.2.3.10)	None	**string** ([MS-ASDTYPE] section 2.6)	0...1 (optional)

If the **ConversationId** element is included as a child of any element other than the **And** element, the server responds with a **Status** element (section 2.2.3.162.12) value of 8 (SearchTooComplex).

The result of including more than one **ConversationId** element in a **Search** command request is undefined. The server SHOULD return a protocol status error in response to such a command request.

2.2.3.36 ConversationMode

The **ConversationMode** element is used in the following command requests:

▪ **GetItemEstimate** command requests (section 2.2.2.7)

▪ **Sync** command requests (section 2.2.2.19)

The definition of the **ConversationId** element differs according to the context in which it is used. For more details, see section 2.2.3.36.1 and section 2.2.3.36.2.

2.2.3.36.1 ConversationMode (GetItemEstimate)

The **airsync:ConversationMode** element<24> is an optional child element of the **Collection** element in **GetItemEstimate** command requests that specifies whether to include items that are included within the conversation modality within the results of the **GetItemEstimate** command response (section 2.2.2.7). A single conversation MAY span multiple collections and therefore **airsync:ConversationMode** is a child of the **Collection** element, rather than the **airsync:Options** element (section 2.2.3.115.1).

Elements referenced in this section are defined in the **GetItemEstimate** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
GetItemEstimate command request	**Collection** (section 2.2.3.29.1)	None	**boolean** ([MS-ASDTYPE] section 2.1)	0...1 (optional)

Setting the **airsync:ConversationMode** element to 0 (**FALSE**) in a **GetItemEstimate** request results in an **Estimate** element (section 2.2.3.62) value that only includes items that meet the **airsync:FilterType** element (section 2.2.3.64.1) value. Setting the **airsync:ConversationMode** element to 1 (**TRUE**) expands the result set to also include items with identical

email2:ConversationId element (as specified in [MS-ASEMAIL] section 2.2.2.14) values to those in the **airsync:FilterType** result set. The **airsync:ConversationMode** element value has no impact on items outside the collection specified by the **CollectionId** element (section 2.2.3.30.1), the result set is always limited to items in the specified collection. The **airsync:ConversationMode** element value only limits or expands the results determined by the **airsync:FilterType** element value.

Specifying the **airsync:ConversationMode** element for collections that do not store email results in an invalid XML error, **Status** element (section 2.2.3.162.6) value 103.

2.2.3.36.2 ConversationMode (Sync)

The **ConversationMode** element<25> is an optional child element of the **Collection** element in **Sync** command requests that specifies whether to include items that are included within the conversation modality within the results of the **Sync** response. A single conversation MAY span multiple classes, and therefore **ConversationMode** is a child of the **Collection** element as opposed to the **Options** element (section 2.2.3.115.5).

Elements referenced in this section are defined in the **AirSync** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
Sync command request (section 2.2.2.19)	**Collection** (section 2.2.3.29.2)	None	**boolean** ([MS-ASDTYPE] section 2.1)	0...1 (optional)

Setting the **ConversationMode** element value to 1 (**TRUE**) results in retrieving all emails that match the conversations received within the date filter specified. However, although the body of the emails outside of that time based filter will not be retrieved, the text previews will be retrieved if the previews were requested.

Setting the **ConversationMode** element value to 0 (**FALSE**) in a **Sync** request results in the synchronization of items that meet the criteria of the **FilterType** element (section 2.2.3.64) value. Setting the **ConversationMode** element value to 1 (**TRUE**) expands the result set to also include any items with identical **email2:ConversationId** ([MS-ASEMAIL] section 2.2.2.14) values to those in the **FilterType** result set. The **ConversationMode** element value has no impact on items outside the collection specified by the **CollectionId** element (section 2.2.3.30.5); the result set is always limited to items in the specified collection. The **ConversationMode** element value only limits or expands the results determined by the **FilterType** element value.

Specifying the **ConversationMode** element for collections that do not store emails results in an invalid XML error, **Status** element (section 2.2.3.162.16) value 4.

2.2.3.37 Count

The **Count** element is an optional child element of the **Changes** element in **FolderSync** command responses that specifies the number of added, deleted, and updated folders on the server since the last folder synchronization. These changes are listed in the **Changes** element. If there are no changes since the last folder synchronization, a **Count** element value of 0 (zero) is returned.

All elements referenced in this section are defined in the **FolderHierarchy** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
FolderSync command response (section 2.2.2.4)	**Changes** (section 2.2.3.25)	None	unsigned **integer** ([MS-ASDTYPE] section 2.5)	0…1 (optional)

2.2.3.38   Culture

The **Culture** element is an optional child element of the **Response** element in **Autodiscover** command responses that specifies the client culture, which is used to localize error messages.

All elements referenced in this section are defined in the http://schemas.microsoft.com/exchange/autodiscover/mobilesync/requestschema/2006 namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Autodiscover command response (section 2.2.2.1)	**Response** (section 2.2.3.140.1)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

The **string** value MUST be of the form "en:us".<26>

2.2.3.39   Data

The **Data** element is used in the following command responses:

- **ItemOperations** command responses (section 2.2.2.8)

- **ResolveRecipients** command responses (section 2.2.2.13)

- **Search** command response (section 2.2.2.14)

The definition of the **Data** element differs according to the context in which it is used. For more details, see section 2.2.3.39.1, section 2.2.3.39.2, and section 2.2.3.39.3.

2.2.3.39.1   Data (ItemOperations)

The **Data** element is an optional child element of the **Properties** element in **ItemOperations** command responses that contains the item content for inline content responses.

All elements referenced in this section are defined in the **ItemOperations** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ItemOperations command response (section 2.2.2.8) fetch operation	**Properties** (section 2.2.3.128.1)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

The content of the **Data** element is a base64 encoding of the binary document, attachment, or body data. The size of the data (in bytes) that is contained within the **Data** element is indicated by the **Range** element (section 2.2.3.130.1). The total size of the item (in bytes) is indicated by the **Total** element (section 2.2.3.169.1).

2.2.3.39.2 Data (ResolveRecipients)

The **Data** element<27> is an optional child element of the **Picture** element in **ResolveRecipients** command responses that contains the binary data of the contact photo.

All elements referenced in this section are defined in the **ResolveRecipients** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ResolveRecipients command response (section 2.2.2.13)	**Picture** (section 2.2.3.125.1)	None	**Byte array**	0…1 (optional)

2.2.3.39.3 Data (Search)

The **gal:Data** element<28> is an optional child element of the **gal:Picture** element in **Search** command responses that contains the binary data of the contact photo.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command response (section 2.2.2.14)	**gal:Picture** (section 2.2.3.125.2)	None	**Byte array**	0…1 (optional)

2.2.3.40 DebugData

The **DebugData** element is an optional child element of the **Error** element in **Autodiscover** command responses that represents information about the failure that can help the system administrator debug the source of the problem.

All elements referenced in this section are defined in the http://schemas.microsoft.com/exchange/autodiscover/mobilesync/requestschema/2006 namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Autodiscover command response (section 2.2.2.1)	**Error** (section 2.2.3.60)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

This element is not intended for use by developers debugging their own data.

2.2.3.41 DeepTraversal

The **DeepTraversal** element is an optional child element of the **Options** element in **Search** command requests that indicates that the client wants the server to search all subfolders for the folders that are specified in the query.

All elements referenced in this section are defined in the **Search** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command request (section 2.2.2.14)	**Options** (section 2.2.3.115.4)	None	None	0…1 (optional)

The result of including more than one **DeepTraversal** element in a **Search** command request is undefined. The server MAY return a protocol status error in response to such a command request.

The **DeepTraversal** 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 <DeepTraversal/> tag.

If the **DeepTraversal** element is not present, the subfolders are not searched.

2.2.3.42 Delete

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

- **FolderSync** command responses (section 2.2.2.4)

- **Sync** command requests and responses (section 2.2.2.19)

The definition of the **Delete** element differs according to the context in which it is used. For more details, see section 2.2.3.42.1 and section 2.2.3.42.2.

2.2.3.42.1 Delete (FolderSync)

The **Delete** element is an optional child element of the **Changes** element in **FolderSync** command responses that specifies that a folder on the server was deleted since the last folder synchronization.

All elements referenced in this section are defined in the **FolderHierarchy** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
FolderSync command response (section 2.2.2.4)	**Changes** (section 2.2.3.25)	**ServerId** (section 2.2.3.151.3)	container ([MS-ASDTYPE] section 2.2)	0…N (optional)

2.2.3.42.2 Delete (Sync)

The **Delete** element is an optional child element of the **Commands** element in **Sync** command requests and responses that deletes an object on the client device or the server. The object is identified by its **ServerId** element.

All elements referenced in this section are defined in the **AirSync** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Sync command request (section 2.2.2.19)	**Commands** (section 2.2.3.32)	**ServerId** (section 2.2.3.151.6)	container ([MS-ASDTYPE] section 2.2)	0…N (optional)
Sync command response	**Commands**	**ServerId** **Class** (section 2.2.3.27.5)<29>	container	0…N (optional)

2.2.3.43 DeletesAsMoves

The **DeletesAsMoves** element is an optional child element of the **Collection** element in **Sync** command requests that indicates that any deleted items SHOULD be moved to the Deleted Items folder.

All elements referenced in this section are defined in the **AirSync** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Sync command request (section 2.2.2.19)	**Collection** (section 2.2.3.29.2)	None	**boolean** ([MS-ASDTYPE] section 2.1)	0...1 (optional)

The **DeletesAsMoves** element appears only in requests sent to the server from the client. If the **DeletesAsMoves** element is set to false, the deletion is permanent.

If the client wants to permanently delete items, the request MUST include the **DeletesAsMoves** element with a value of 0 (**FALSE**). A value of 1 (**TRUE**), which is the default, indicates that any deleted items are moved to the Deleted Items folder. The default is assumed when the **DeletesAsMoves** element is either empty or not present.

2.2.3.44 DeleteSubFolders

The **DeleteSubFolders** element is an optional child element of the **Options** element in **ItemOperations** command requests for the **EmptyFolderContents** (section 2.2.3.55) operation. It is a flag that indicates whether to delete the subfolders of the specified folder.

All elements referenced in this section are defined in the **ItemOperations** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ItemOperations command request (section 2.2.2.8) **EmptyFolderContents** operation	**Options** (section 2.2.3.115.2)	None	None	0...1 (optional)

If the **DeleteSubFolders** element is not present in the request, the default behavior is to not delete subfolders.

2.2.3.45 DeviceInformation

The **DeviceInformation** element is an optional child element of the **Settings** element in **Settings** command requests and responses, and an optional child element of the **Provision** element in **Provision** command requests. In **Settings** and **Provision** command requests, it is a container node that is used for sending the client device's properties to the server. In **Settings** command responses, it is a container node that is used to return the status of the request.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command request	**Settings** (section	**Set** (section	**container** ([MS-ASDTYPE]	0...1

Command request/response	Parent elements	Child elements	Data type	Number allowed
(section 2.2.2.16)	2.2.3.153.2)	2.2.3.152)	section 2.2)	(optional)
Settings command response	**Settings**	**Status** (section 2.2.3.162.14)	**container**	0...1 (optional)
Provision command request ([MS-ASPROV] section 2.2)	**Provision** ([MS-ASPROV] section 2.2.2.43)	**Set**	**container**	0...1 (optional)

When the value of the MS-ASProtocolVersion header ([MS-ASHTTP] section 2.2.1.1.2.4) is 14.1, the **DeviceInformation** element MUST be present as a child element of the **Provision** element in the initial **Provision** command (section 2.2.2.12). When the value of the MS-ASProtocolVersion header is 14.0 or 12.1, the **DeviceInformation** element MUST NOT be present as a child element of the **Provision** element. In this case, the client uses the **Settings** command to send **DeviceInformation** to the server, as specified in section 3.1.5.2.

When the **Settings** command (section 2.2.2.16) is used to send the **DeviceInformation** element, it sends the following information about a client device to the server:

- Device model

- International Mobile Equipment Identity (IMEI)

- Device friendly name

- Device operating system

- Telephone number

- Device operating system language

- User Agent

- Whether to enable outbound SMS (see [MS-ASMS])

- Mobile operator name

The device information is represented as a list of settings under the **DeviceInformation** node in the **Settings** command. The **DeviceInformation** element has only one child element, **Set**, which contains the list of device information items in the request and the status in the response. The **DeviceInformation** element supports only the **Set** element because this information is write-only from the device.

2.2.3.46 DevicePassword

The **DevicePassword** element is an optional child element of the **Settings** element in **Settings** command requests and responses. In **Settings** command requests, the **DevicePassword** element is a container node that is used to send the recovery password of the client device to the server. In **Settings** command responses, the **DevicePassword** element is a container node that is used to specify the status of the **Settings** command **DevicePassword Set** operation.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command request (section 2.2.2.16)	**Settings** (section 2.2.3.153.2)	**Set** (section 2.2.3.152)	**container** ([MS-ASDTYPE] section 2.2)	0…1 (optional)
Settings command response	**Settings**	**Status** (section 2.2.3.162.14)	**container**	0…1 (optional)

Clients use the **Set** operation within the **DevicePassword** element to enable the device to send or store a recovery password on the server. The recovery password is stored in the user's mailbox and can be retrieved by the administrator or the end-user if the user forgets his or her password.

2.2.3.47 DisplayName

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

- **Autodiscover** command responses (section 2.2.2.1)

- **FolderCreate** command requests (section 2.2.2.2)

- **FolderSync** command responses (section 2.2.2.4)

- **FolderUpdate** command requests (section 2.2.2.5)

- **ResolveRecipients** command responses (section 2.2.2.13)

- **Search** command responses (section 2.2.2.14)

The definition of the **DisplayName** element differs according to the context in which it is used. For more details, see section 2.2.3.47.1, section 2.2.3.47.2, section 2.2.3.47.3, section 2.2.3.47.4, and section 2.2.3.47.5.

2.2.3.47.1 DisplayName (Autodiscover)

The **DisplayName** element is an optional child element of the **User** element in **Autodiscover** command responses that contains the user's display name in the directory service.

All elements referenced in this section are defined in the http://schemas.microsoft.com/exchange/autodiscover/mobilesync/requestschema/2006 namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Autodiscover command response (section 2.2.2.1)	**User** (section 2.2.3.173)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

The client can choose to display or store this value on the device.

2.2.3.47.2 DisplayName (FolderCreate and FolderUpdate)

The **DisplayName** element is a required child element of the **FolderCreate** element in **FolderCreate** command requests and a required child element of the **FolderUpdate** element in

FolderUpdate command requests. In both contexts, it specifies the name of the folder that is shown to the user.

All elements referenced in this section are defined in the **FolderHierarchy** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
FolderCreate command request (section 2.2.2.2)	**FolderCreate** (section 2.2.3.67)	None	**string** ([MS-ASDTYPE] section 2.6)	1…1 (required)
FolderUpdate command request (section 2.2.2.5)	**FolderUpdate** (section 2.2.3.72)	None	**string**	1…1 (required)

The **DisplayName** element value MUST be between 1 and 256 characters in length.

2.2.3.47.3   DisplayName (FolderSync)

The **DisplayName** element is a required child element of the **Update** element and the **Add** element in **FolderSync** command responses that specifies the name of the folder that is shown to the user.

All elements referenced in this section are defined in the **FolderHierarchy** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
FolderSync command response (section 2.2.2.4)	**Update** (section 2.2.3.171)	None	**string** ([MS-ASDTYPE] section 2.6)	1…1 (required)
FolderSync command response	**Add** (section 2.2.3.7.1)	None	**string**	1…1 (required)

One **DisplayName** element is used in each **Update** element and in each **Add** element included in a **FolderSync** command response when a folder has been updated or added on the server. Subfolder display names MUST be unique within a folder.

2.2.3.47.4   DisplayName (ResolveRecipients)

The **DisplayName** element is a required child element of the **Recipient** element in **ResolveRecipients** command responses that contains the display name of the recipient.

All elements referenced in this section are defined in the **ResolveRecipients** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ResolveRecipients command response (section 2.2.2.13)	**Recipient** (section 2.2.3.132)	None	**string** ([MS-ASDTYPE] section 2.6)	1…1 (required)

2.2.3.47.5   DisplayName (Search)

The **DisplayName** element in the **GAL** namespace is an optional child element of the **search:Properties** element in **Search** command responses. The **DisplayName** element is only

present in response to **Search** command requests that contain a **search:Name** element with a value of "GAL", as specified in section 2.2.3.110.2. It contains the display name of a recipient in the GAL that matched the search criteria from the corresponding **Search** command request.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command response (section 2.2.2.14)	**Properties** (section 2.2.3.128.2)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

2.2.3.48 documentlibrary:LinkId

The **documentlibrary:LinkId** element is an optional child element of the **itemoperations:Fetch** element in **ItemOperations** command requests and responses. The **documentlibrary:LinkId** element is a required child element of the **search:EqualTo** element in **Search** command requests and an optional child element of the **search:Properties** element in **Search** command responses. In all contexts, the **documentlibrary:LinkId** element specifies a **Uniform Resource Identifier (URI)** that is assigned by the server to certain **resources**, such as Windows SharePoint Services or UNC documents.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ItemOperations command request (section 2.2.2.8) fetch operation	**itemoperations:Fetch** (section 2.2.3.63.1)	None	**URI** (as specified in [RFC3986])	0…1 (optional)
ItemOperations command response fetch operation	**itemoperations:Fetch**	None	**URI**	0…1 (optional)
Search command request (section 2.2.2.14)	**search:EqualTo** (section 2.2.3.59)	None	**URI**	1…1 (required)
Search command response	**search:Properties** (section 2.2.3.128.2)	None	**URI**	0…1 (optional)

The client MUST store the **documentlibrary:LinkId** element value that is retrieved by the **Search** command (section 2.2.2.14) if the client will send requests using the **documentlibrary:LinkId** element value in the future. In an **ItemOperations** command request, the **documentlibrary:LinkId** element can be used by the **itemoperations:Fetch** element to refer to the location of an item.

2.2.3.49 DstFldId

The **DstFldId** element is used in the following command requests:

- **ItemOperations** command requests (section 2.2.2.8)

- **MoveItems** command requests (section 2.2.2.10)

The definition of the **DstFldId** element differs according to the context in which it is used. For more details, see section 2.2.3.49.1 and section 2.2.3.49.2.

2.2.3.49.1 DstFldId (ItemOperations)

The **DstFldId** element is a required child element of the **Move** element in **ItemOperations** command requests that specifies the server ID of the destination folder (that is, the folder to which the items are moved).<30>

Elements referenced in this section are defined in the **ItemOperations** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
ItemOperations command request (section 2.2.2.8) **Move** operation	**Move** (section 2.2.3.107.1)	None	**string** ([MS-ASDTYPE] section 2.6)	1…1 (required)

The server ID of the destination folder is obtained from the **folderhierarchy:ServerId** element (section 2.2.3.151) of a previous **FolderSync** command response (section 2.2.2.4) or **FolderCreate** command response (section 2.2.2.2). The **DstFldId** element value can be up to 64 characters in length.

2.2.3.49.2 DstFldId (MoveItems)

The **DstFldId** element is a required child element of the **Move** element in **MoveItems** command requests that specifies the server ID of the destination folder (that is, the folder to which the items are moved).<31>

Elements referenced in this section are defined in the **Move** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
MoveItems command request (section 2.2.2.10)	**Move** (section 2.2.3.107.2)	None	**String** ([MS-ASDTYPE] section 2.6)	1…1 (required)

The server ID of the destination folder is obtained from the **folderhierarchy:ServerId** element (section 2.2.3.151) of a previous **FolderSync** command response (section 2.2.2.4) or **FolderCreate** command response (section 2.2.2.2). The **DstFldId** element value can be up to 64 characters in length.

2.2.3.50 DstMsgId

The **DstMsgId** element is an optional child element of the **Response** element in **MoveItems** command responses that specifies the new server ID of the item after the item is moved to the destination folder.

All elements referenced in this section are defined in the **Move** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
MoveItems command response (section 2.2.2.10)	**Response** (section 2.2.3.140.4)	None	**string** ([MS-ASDTYPE] section	0…1 (optional)

Command request/response	Parent elements	Child elements	Data type	Number allowed
			2.6)	

The **DstMsgId** element value can be up to 64 characters in length.

2.2.3.51   email:DateReceived

The **email:DateReceived** element is a required child element of the **GreaterThan** element and the **LessThan** element in **Search** command requests. The corresponding value that is to be compared to the **email:DateReceived** element on items in the mailbox is specified in the **Value** element, specified in section 2.2.3.180. The format and usage of the **email:DateReceived** element for command requests or responses other than the **Search** command is specified in [MS-ASEMAIL] section 2.2.2.17.

All elements referenced in this section are defined in the **Search** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command request (section 2.2.2.14)	**GreaterThan** (section 2.2.3.78) **LessThan** (section 2.2.3.87)	None	None	1…1 (required)

2.2.3.52   EMailAddress

The **EMailAddress** element is a required child element of the **Request** element in **Autodiscover** command requests and a required child element of the **User** element in **Autodiscover** command responses. In both contexts, it contains the SMTP email address of the user and is used to identify the user's mailbox in the network.

All elements referenced in this section are defined in the http://schemas.microsoft.com/exchange/autodiscover/mobilesync/requestschema/2006 namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Autodiscover command request (section 2.2.2.1)	**Request** (section 2.2.3.137.1)	None	**string** ([MS-ASDTYPE] section 2.6)	1…1 (required)
Autodiscover command response	**User** (section 2.2.3.173)	None	**string**	1…1 (required)

If the user has multiple addresses, then the primary email address SHOULD be returned in the **Autodiscover** command response. The primary email address is the address that appears on the From line when a user sends an email message. This address can be the same as the email address that was sent in the request. The client device SHOULD record the address returned by the **Autodiscover** command response and SHOULD use this **string** value for all additional communication.

Release: February 10, 2014

2.2.3.53 EmailAddress

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

▪ **ResolveRecipients** command responses (section 2.2.2.13)

▪ **Search** command responses (section 2.2.2.14)

The definition of the **EmailAddress** element differs according to the context in which it is used. For more details, see section 2.2.3.53.1 and section 2.2.3.53.2.

2.2.3.53.1 EmailAddress (ResolveRecipients)

The **EmailAddress** element is a required child element of the **Recipient** element in **ResolveRecipients** command responses that contains the email address of the recipient, in SMTP format.

All elements referenced in this section are defined in the **ResolveRecipients** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ResolveRecipients command response (section 2.2.2.13)	**Recipient** (section 2.2.3.132)	None	**string** ([MS-ASDTYPE] section 2.6)	1…1 (required)

2.2.3.53.2 EmailAddress (Search)

The **EmailAddress** element in the **GAL** namespace is an optional child element of the **search:Properties** element in **Search** command responses. The **EmailAddress** element is only present in response to **Search** command requests that contain a **search:Name** element with a value of "GAL", as specified in section 2.2.3.110.2. It contains the email address of a recipient in the GAL that matched the search criteria from the corresponding **Search** command request.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command response (section 2.2.2.14)	**Properties** (section 2.2.3.128.2)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

2.2.3.54 EmailAddresses

The **EmailAddresses** element is an optional child element of the **Account** element and the **Get** element in **Settings** command responses that contains one or more email addresses for the user.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command **UserInformation** response (section 2.2.2.16)	**Account**<32> (section 2.2.3.2) **Get**<33>	**SMTPAddress** (section 2.2.3.156) **PrimarySmtpAddress**<34> (section 2.2.3.127)	**container** ([MS-ASDTYPE] section 2.2)	0…1 (optional)

Command request/response	Parent elements	Child elements	Data type	Number allowed
	(section 2.2.3.75)			

2.2.3.55 EmptyFolderContents

The **EmptyFolderContents** element is an optional child element of the **ItemOperations** element in **ItemOperations** command requests and an optional child element of the **Response** element in **ItemOperations** responses. In both contexts, the **EmptyFolderContents** element identifies the body of the request or response as containing the operation that deletes the contents of a folder.

Elements referenced in this section are defined in the **ItemOperations** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
ItemOperations command request (section 2.2.2.8)	**ItemOperations** (section 2.2.3.85)	**airsync:CollectionId** (section 2.2.3.30.2) **Options** (section 2.2.3.115.2)	container ([MS-ASDTYPE] section 2.2)	0...N (optional)
ItemOperations command response	**Response** (section 2.2.3.140.3)	**airsync:CollectionId** (section 2.2.3.30.2) **Status** (section 2.2.3.162.7)	container	0...N (optional)

The **EmptyFolderContents** element enables the client to empty a folder of all its items. The element supports a single child element of the **Options** element, the **DeleteSubFolders** element (section 2.2.3.44), which determines whether subfolders contained in the folder are deleted. If the **DeleteSubFolders** option is not included in the request, the subfolders of the specified **airsync:CollectionId** are not deleted.

Specifically, clients use **EmptyFolderContents** to empty the Deleted Items folder. The client can clear out all items in the Deleted Items folder when the user runs out of storage quota (indicated by the return of an HTTP 507 status code from the server).

2.2.3.56 Enabled

The **Enabled** element is an optional child element of the **OofMessage** element in **Settings** command requests and responses that specifies whether an OOF message is sent to this audience while the sending user is OOF.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command **Oof** request and response (section 2.2.2.16)	**OofMessage** (section 2.2.3.113)	None	string ([MS-ASDTYPE] section 2.6)	0...1 (optional)

The **Enabled** element is used in the **Settings** command **Oof Get** (section 2.2.3.75) response to retrieve the current value. The **Enabled** element is used in the **Settings** command **Oof Set** request to set the value.

The value of the **Enabled** element is 1 if an OOF message is sent while the sending user is OOF; otherwise, the value is 0 (zero).

2.2.3.57 EnableOutboundSMS

The **EnableOutboundSMS** element is an optional<35> child element of the **Set** element in **Settings** command requests that specifies whether the server will send outbound SMS messages through the mobile device. For more details, see [MS-ASMS] section 2.2.2.2.2.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command **DeviceInformation** request (section 2.2.2.16)	**Set** (section 2.2.3.152)	None	**integer** ([MS-ASDTYPE] section 2.5)	0…1 (optional)

2.2.3.58 EndTime

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

- **ResolveRecipients** command requests (section 2.2.2.13)

- **Settings** command requests and responses (section 2.2.2.16)

The definition of the **EndTime** element differs according to the context in which it is used. For more details, see section 2.2.3.58.1 and section 2.2.3.58.2.

2.2.3.58.1 EndTime (ResolveRecipients)

The **EndTime** element is an optional child element of the **Availability** element in **ResolveRecipients** command requests that identifies the end of the span of free/busy time requested by the client.<36>

All elements referenced in this section are defined in the **ResolveRecipients** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ResolveRecipients command request (section 2.2.2.13)	**Availability** (section 2.2.3.16)	None	**datetime** ([MS-ASDTYPE] section 2.3)	0…N (optional)

If the **Availability** element is included in the request, the request MUST also include the **StartTime** (section 2.2.3.161.1) element.

If the client sends an invalid **EndTime** element value, then the server returns a **Status** element (section 2.2.3.162.11) value of 5 in the **ResolveRecipients** command response.

If no **EndTime** element is included in an **Availablity** request, the server uses a default end time value of seven days after the **StartTime** value.

If the **EndTime** element value specified in the request is smaller than the **StartTime** element value plus 30 minutes, or the duration spanned by the **StartTime** element value and the **EndTime** element value is greater than a specific number<37> of days, then the server returns a **Status** element value of 5 in the **ResolveRecipients** command response.

2.2.3.58.2 EndTime (Settings)

The **EndTime** element is an optional child element of the **Set** element in **Settings** command requests and an optional child element of the **Get** element in **Settings** command responses. It is used with the **StartTime** element to specify the range of time during which the user is out of office.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command **Oof** request (section 2.2.2.16)	**Set** (section 2.2.3.152)	None	**datetime** ([MS-ASDTYPE] section 2.3)	0…1 (optional)
Settings command **Oof** response	**Get** (section 2.2.3.75)	None	**datetime**	0…1 (optional)

The **EndTime** element can be present within the **Get** element of the **Settings** response for the **Oof** property (section 2.2.3.112) or within the **Set** element of the **Settings** request for the **Oof** property.

In a **Set Oof** node, both the **StartTime** element and the **EndTime** element MUST be included in the **Settings** request, or neither the **StartTime** element nor the **EndTime** element MUST be included in the **Settings** request. If either the **StartTime** element or the **EndTime** element is included in the request without the other, a **Status** element (section 2.2.3.162.14) value of 2 is returned as a child of the **Oof** element.

2.2.3.59 EqualTo

The **EqualTo** element is an optional child element of the **Query** element in **Search** command requests that contains a property and a value that are compared for equality during a search.

Elements referenced in this section are defined in the **Search** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command request (section 2.2.2.14)	**Query** (section 2.2.3.129)	**documentlibrary:LinkId** (section 2.2.3.48) **Value** (section 2.2.3.180)	**container** ([MS-ASDTYPE] section 2.2)	0…1 (optional)

The result of including more than one **EqualTo** element in a **Search** command request is undefined. The server SHOULD return a protocol status error in response to such a command request.

The **Query** element is only supported as a parent element in a document library search.

The comparison is made between the value of the **Value** element and the **documentlibrary:LinkId** element value.

2.2.3.60 Error

The **Error** element is an optional child element of the **Action** element and an optional child element of the **Response** element in **Autodiscover** command responses. In both contexts, it specifies information about the error that was encountered while processing the request.

All elements referenced in this section are defined in the http://schemas.microsoft.com/exchange/autodiscover/mobilesync/requestschema/2006 namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Autodiscover command response (section 2.2.2.1)	**Action** (section 2.2.3.6)	**Status** (section 2.2.3.162.1) **Message** (section 2.2.3.98) **DebugData** (section 2.2.3.40) **ErrorCode** (section 2.2.3.61)	**container** ([MS-ASDTYPE] section 2.2)	0...1 (optional)
Autodiscover command response	**Response** (section 2.2.3.140.1)	**ErrorCode** **Message** **DebugData**	**container**	0...1 (optional)

When the **Error** element is a child of the **Response** element, it has two optional attributes as specified in the following table.

Attribute	Value
Time	The value is a string that represents the time of day in that the request that generated the response was submitted. The format of this string follows the **Augmented Backus-Naur Form (ABNF)** syntax specified below.
Id	The value is an unsigned integer that uniquely identifies the server (within its domain) that generated the response.

The value of the **Time** attribute is specified here using ABNF syntax, as specified in [RFC5234].

```
time_val             = hours ":" minutes ":" seconds ["." fractional_seconds]

hours                = 2*DIGIT  ; 00 - 23, representing a 24-hour clock
minutes              = 2*DIGIT  ; 00 - 59
seconds              = 2*DIGIT  ; 00 - 59
fractional_seconds   = 7*DIGIT  ; fractional seconds, always to 7 decimal places
```

2.2.3.61 ErrorCode

The **ErrorCode** element is an optional child element of the **Error** element in **Autodiscover** command responses that specifies an error number that indicates the cause of the error.

All elements referenced in this section are defined in the
http://schemas.microsoft.com/exchange/autodiscover/mobilesync/requestschema/2006 namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Autodiscover command response (section 2.2.2.1)	**Error** (section 2.2.3.60)	None	**integer** ([MS-ASDTYPE] section 2.5)	0…1 (optional)

If the provider cannot be found, or if the **AcceptableResponseSchema** element (section 2.2.3.1) value cannot be matched, then the **ErrorCode** element is included in the command response. A value of 600 means an invalid request was sent to the server; a value of 601 means that a provider could not be found to handle the **AcceptableResponseSchema** element value that was specified.

2.2.3.62 Estimate

The **Estimate** element is a required child element of the **Collection** element in **GetItemEstimate** command responses that specifies the estimated number of items in the collection or folder that have to be synchronized.

All elements referenced in this section are defined in the **GetItemEstimate** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
GetItemEstimate command response (section 2.2.2.7)	**Collection** (section 2.2.3.29.1)	None	**integer** ([MS-ASDTYPE] section 2.5)	1…1 (required)

2.2.3.63 Fetch

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

▪ **ItemOperations** command requests and responses (section 2.2.2.8)

▪ **Sync** command requests and responses (section 2.2.2.19)

The definition of the **Fetch** element differs according to the context in which it is used. For more details, see section 2.2.3.63.1 and section 2.2.3.63.2.

2.2.3.63.1 Fetch (ItemOperations)

The **Fetch** element is an optional child element of the **ItemOperations** element in **ItemOperations** command requests and an optional child element of the **Response** element in **ItemOperations** responses. In both contexts, it retrieves an item from the server.

Elements referenced in this section are defined in the **ItemOperations** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
ItemOperations command request	**ItemOperations** (section	**Store** (section 2.2.3.163.1)	**container** ([MS-	0…N (optional

Command request/response	Parent elements	Child elements	Data type	Number allowed
(section 2.2.2.8)	2.2.3.85)	**documentlibrary:LinkId** (section 2.2.3.48) (optional) **search:LongId** (section 2.2.3.89.1) (optional) **airsync:CollectionId** (optional) (section 2.2.3.30.2) **airsync:ServerId** (optional) (section 2.2.3.151.5) **Options** (optional) (section 2.2.3.115.2) **airsyncbase:FileReference** (optional) (section 2.2.3.8) **rm:RemoveRightsManagementProtection** ([MS-ASRM] section 2.2.2.11)	ASDTYPE] section 2.2)	)
ItemOperations command response	**Response** (section 2.2.3.140.3)	**documentlibrary:LinkId** (optional) **search:LongId** (optional) **airsync:CollectionId** (optional) **airsync:ServerId** (optional) **Status** (section 2.2.3.162.7) **airsync:Class** (optional) (section 2.2.3.27.2) **Properties** (optional) (section 2.2.3.128.1)	container	0…N (optional)

The fetch operation is intended to be used on Windows SharePoint Services technology or UNC document metadata, search results, and items and attachments.

Because the **ItemOperations** response potentially contains large amounts of binary data, this protocol enables the client to choose a delivery method that is most efficient for its implementation by providing the following two methods to deliver content that is requested by the **Fetch** element:

- Inline—The binary content is encoded with base64 encoding and is included inside the WBXML.

- Multipart—This method involves a multipart structure in which the WBXML is the first part, and the requested data populates the subsequent parts. This format enables a client to handle large files without consuming large amounts of memory.

The inline approach generally requires the client to read the WBXML part into memory in order to parse it, thereby consuming a large amount of memory. The multipart approach enables the client to parse the small WBXML part, obtain references to the binary parts, and handle the binary parts as necessary, without reading the whole response into memory.

In the request, the client specifies the location and a byte range for the item. The location is indicated by either a link ID (**documentlibrary:LinkId** element) if the target item is identified by a URI, or a file reference (**airsyncbase:FileReference** element (section 2.2.3.8)) if the client is retrieving an email attachment. The location is indicated by a server ID (**airsync:ServerId** element) if an ActiveSync ID is being used to identify the item.

The **Fetch** element supports several options, such as byte ranges, body preference, and schema, as specified in section 2.2.3.115.2.

The **ItemOperations** command response contains either the requested byte range of the item, or an error code that indicates why the fetch failed.

Multiple fetch operations can be included within one **ItemOperations** request. In this case, the fetch operations are executed in the order that is specified.

For more information about the **Fetch** element, see section 2.2.2.8.1.

2.2.3.63.2 Fetch (Sync)

The **Fetch** element is an optional child element of the **Commands** element in **Sync** command requests and an optional child element of the **Responses** element in **Sync** command responses. In a request, the **Fetch** element is used to request the application data of an item that was truncated in a synchronization response from the server. The complete item is then returned to the client in a server response.

The **ItemOperations** command (section 2.2.2.8) is the preferred way to fetch items.

All elements referenced in this section are defined in the **AirSync** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Sync command request (section 2.2.2.19)	**Commands** (section 2.2.3.32)	**ServerId** (section 2.2.3.151.6)	**container** ([MS-ASDTYPE] section 2.2)	0...N (optional)
Sync command response	**Responses** (section 2.2.3.141)	**ServerId** **Status** (section 2.2.3.162.16) **ApplicationData** (section 2.2.3.11)	**container**	0...N (optional)

The **Fetch** element cannot be used to get truncated calendar, contact, note, recipient information, or task items from the server.

2.2.3.64 FilterType

The **FilterType** element is used in the following command requests:

▪ **GetItemEstimate** command requests (section 2.2.2.7)

▪ **Sync** command requests (section 2.2.2.19)

The definition of the **FilterType** element differs according to the context in which it is used. For more details, see section 2.2.3.64.1 and section 2.2.3.64.2.

2.2.3.64.1 FilterType (GetItemEstimate)

The **airsync:FilterType** element is an optional child element of the **airsync:Options** element and the **Collection** element in **GetItemEstimate** command requests. In both contexts, it specifies a time window for the objects sent from the server to the client.

Elements referenced in this section are defined in the **GetItemEstimate** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
GetItemEstimate command request (section 2.2.2.7)	**airsync:Options** (section 2.2.3.115.1) **Collection** (section 2.2.3.29.1)<38>	None	**integer** ([MS-ASDTYPE] section 2.5)	0...1 (optional)

The **airsync:FilterType** element applies to email, calendar, and task collections. If a filter type is specified, then the server sends an estimate of the items within the filter specifications.

If the **airsync:FilterType** element is present in the request, then the server manages objects on the client to maintain the specified time window. New objects are added to the client when they are within the time window. The start time of the time window is determined by subtracting the period of time represented by the value of the **FilterType** element from the current time. Any items that are dated between the start time of the window and the current time are considered within the time window. For example, if the value of the **FilterType** element is 2, then the start time of the time window is three days prior to the current time. If the **airsync:FilterType** element is omitted, then all objects are sent from the server.

Calendar items that are in the future or that have recurrence, but no end date, are sent to the client regardless of the **airsync:FilterType** element value.

The **airsync:FilterType** element does not support contact collections. However, if the **airsync:FilterType** element is included in a **GetItemEstimate** request for a contact collection, no error is returned.

The valid values for each collections type are listed in the following table.

Value	Meaning	Applies to email	Applies to calendar	Applies to tasks
0	No filter	Yes	Yes	Yes
1	1 day	Yes	No, **Status** element (section 2.2.3.162.6) value 110	No, **Status** element value 110
2	3 days	Yes	No, **Status** element value 110	No, **Status** element value 110
3	1 week	Yes	No, **Status** element value 110	No, **Status** element value 110
4	2 weeks	Yes	Yes	No, **Status** element value 110
5	1 month	Yes	Yes	No, **Status** element value 110
6	3 months	No, **Status** element value 110	Yes	No, **Status** element value 110
7	6 months	No, **Status** element value 110	Yes	No, **Status** element value 110
8	Incomplete tasks	No, **Status** element value 110	No, **Status** element value 110	Yes

Specifying a **airsync:FilterType** of 9 or above for when the **CollectionId** element (section 2.2.3.30.1) identifies any email, contact, calendar or task collection results in a **Status** element value of 103.

2.2.3.64.2 FilterType (Sync)

The **FilterType** element is an optional child element of the **Options** element in **Sync** command requests that specifies an optional time window for the objects that are sent from the server to the client. It applies to email and calendar collections. If a **FilterType** element is specified, the server sends only objects that are dated within the specified time window.

All elements referenced in this section are defined in the **AirSync** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Sync command request (section 2.2.2.19)	**Options** (section 2.2.3.115.5)	None	**unsignedByte** ([MS-ASDTYPE] section 2.7)	0...1 (optional)

The following table lists valid values for the element.

Value	Meaning	Email	Calendar	Tasks
0	No filter- synchronize all items	Yes	Yes	Yes
1	1 day	Yes	No	No
2	3 days	Yes	No	No
3	1 week	Yes	No	No
4	2 weeks	Yes	Yes	No
5	1 month	Yes	Yes	No
6	3 months	No	Yes	No
7	6 months	No	Yes	No
8	Filter by incomplete tasks	No	No	Yes

When the **FilterType** element is specified, the server manages objects on the client to maintain the time window. New objects are added when they are within the time window. The server sends **SoftDelete** (section 2.2.3.157) operations for objects on the client when they become older than the window. The start time of the time window is determined by subtracting the period of time represented by the value of the **FilterType** element from the current time. Any items that are dated between the start time of the window and the current time are considered within the time window. For example, if the value of the **FilterType** element is 2, then the start time of the time window is three days prior to the current time.

Calendar items that are in the future or that have recurrence but no end date are sent to the client regardless of the **FilterType** element value. Calendar items that fall between the **FilterType** element value and the present time are returned to the client. For example, an appointment that occurred two weeks ago is returned when the **FilterType** element value is set to 5 (one month back). The result of including a **FilterType** element value of 8 for a Calendar item is undefined. The server MAY return a protocol status error in response to such a command request.

Release: February 10, 2014

The **FilterType** element is a child of the **Options** element. Therefore, it appears only in requests to the server from the client.

The result of including more than one **FilterType** element as the child of the **Options** element is undefined. The server MAY return a protocol status error in response to such a command request.

If the **FilterType** element is omitted, all objects are sent from the server without regard for their age. Clients MUST send a maximum of one **FilterType** element.

The server ignores the **FilterType** element if it is included in a contact **Sync** request, no error is thrown.

The server returns a **Status** element (section 2.2.3.162.16) value of 4 if a **FilterType** element value of 8 is included in an email **Sync** request.

2.2.3.65 FirstName

The **FirstName** element in the **GAL** namespace is an optional child element of the **search:Properties** element in **Search** command responses. The **FirstName** element is only present in response to **Search** command requests that contain a **search:Name** element with a value of "GAL", as specified in section 2.2.3.110.2. It contains the first name of a recipient in the GAL that matched the search criteria from the corresponding **Search** command request.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command response (section 2.2.2.14)	**Properties** (section 2.2.3.128.2)	None	string ([MS-ASDTYPE] section 2.6)	0...1 (optional)

2.2.3.66 Folder

The **Folder** element is a required child element of the **Folders** element in **Ping** command requests that identifies the folder and folder type to be monitored by the client. The **Folder** element is a required child element of the **Folders** element in **Ping** command responses that identifies the folder that is being described by the returned status code.

All elements referenced in this section are defined in the **Ping** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Ping command request (section 2.2.2.11)	**Folders** (section 2.2.3.70)	**Id** (section 2.2.3.81) **Class** (section 2.2.3.27.3)	container ([MS-ASDTYPE] section 2.2)	1...N (required)
Ping command response	**Folders** (section 2.2.3.70)	None	string ([MS-ASDTYPE] section 2.6)	1...N (required)

2.2.3.67 FolderCreate

The **FolderCreate** element is a required element in **FolderCreate** command requests and **FolderCreate** command responses that identifies the body of the HTTP **POST** as containing a **FolderCreate** command (section 2.2.2.2). It is the top-level element in the XML stream.

All elements referenced in this section are defined in the **FolderHierarchy** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
FolderCreate command request (section 2.2.2.2)	None	**SyncKey** (section 2.2.3.166.1) **ParentId** (section 2.2.3.119.1) **DisplayName** (section 2.2.3.47.2) **Type** (section 2.2.3.170.2)	**container** ([MS-ASDTYPE] section 2.2)	1…1 (required)
FolderCreate command response	None	**SyncKey** **ServerId** (section 2.2.3.151.1) **Status** (section 2.2.3.162.2)	**container**	1…1 (required)

2.2.3.68 FolderDelete

The **FolderDelete** element is a required element in **FolderDelete** command requests and **FolderDelete** command responses that identifies the body of the HTTP **POST** as containing a **FolderDelete** command (section 2.2.2.3). It is the top-level element in the XML stream.

All elements referenced in this section are defined in the **FolderHierarchy** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
FolderDelete command request	None	**SyncKey** (section 2.2.3.166.1) **ServerId** (section 2.2.3.151.2)	**container** ([MS-ASDTYPE] section 2.2)	1…1 (required)
FolderDelete command response	None	**SyncKey** **Status** (section 2.2.3.162.3)	**container**	1…1 (required)

2.2.3.69 FolderId

The **FolderId** element is an optional child element of the **Source** element in **SmartForward** command requests and **SmartReply** command requests that specifies the folder ID for the source message, which is returned in the **FolderSync** command response message (section 2.2.2.4). If the **FolderId** element is present, the **ItemId** element (section 2.2.3.84) MUST also be present. The **FolderId** element value can be up to 64 characters in length.

All elements referenced in this section are defined in the **ComposeMail** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
SmartForward command	**Source** (section	None	**string** ([MS-ASDTYPE] section	0…1

Command request/response	Parent elements	Child elements	Data type	Number allowed
request (section 2.2.2.17)	2.2.3.158)		2.6)	(optional)
SmartReply command request (section 2.2.2.18)	**Source**	None	**string**	0…1 (optional)

2.2.3.70   Folders

The **Folders** element is an optional child element of the **Ping** element in **Ping** command requests and responses that contains the **Folder** element.

All elements referenced in this section are defined in the **Ping** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Ping command request (section 2.2.2.11)	**Ping** (section 2.2.3.126)	**Folder** (section 2.2.3.66)	**container** ([MS-ASDTYPE] section 2.2)	0…1 (optional)
Ping command response	**Ping**	**Folder**	**container**	0…1 (optional)

2.2.3.71   FolderSync

The **FolderSync** element is a required element in **FolderSync** command requests and **FolderSync** command responses that identifies the body of the HTTP **POST** as containing a **FolderSync** command (section 2.2.2.4). It is the top-level element in the XML stream.

All elements referenced in this section are defined in the **FolderHierarchy** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
FolderSync command request (section 2.2.2.4)	None	**SyncKey** (section 2.2.3.166.2)	**container** ([MS-ASDTYPE] section 2.2)	1…1 (required)
FolderSync command response	None	**SyncKey** **Status** (section 2.2.3.162.4) **Changes** (section 2.2.3.25)	**container**	1…1 (required)

2.2.3.72   FolderUpdate

The **FolderUpdate** element is a required element in **FolderUpdate** command requests and **FolderUpdate** command responses that identifies the body of the HTTP **POST** as containing a **FolderUpdate** command (section 2.2.2.5). It is the top-level element in the XML stream.

All elements referenced in this section are defined in the **FolderHierarchy** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
FolderUpdate command request	None	**SyncKey** (section 2.2.3.166.1) **ServerId** (section 2.2.3.151.4) **ParentId** (section 2.2.3.119.3) **DisplayName** (section 2.2.3.47.2)	**container** ([MS-ASDTYPE] section 2.2)	1…1 (required)
FolderUpdate command response	None	**SyncKey** **Status** (section 2.2.3.162.5)	**container**	1…1 (required)

Including the **Status** element in a **FolderUpdate** request results in a **Status** element value of 10 being returned in the response.

2.2.3.73 FreeText

The **FreeText** element is an optional child element of the **And** element in **Search** command requests that specifies a **string** value for which to search.

All elements referenced in this section are defined in the **Search** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command request (section 2.2.2.14)	**And** (section 2.2.3.10)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

If the **FreeText** element is included as a child element of any element other than the **And** element, the server responds with a **Status** element (section 2.2.3.162.12) value of 8 (SearchTooComplex).

2.2.3.74 FriendlyName

The **FriendlyName** element is an optional child element of the **Set** element in **Settings** command requests that specifies a name that MUST uniquely describe the client device.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command **DeviceInformation** request (section 2.2.2.16)	**Set** (section 2.2.3.152)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

The friendly name of the device is a string that is meaningful to the user. The server does not validate this value. The **FriendlyName** element value can be up to 1024 characters in length.

The friendly name of the device is specified during partnership creation if the user creates a desktop-device partnership with a desktop. For more information about creating a desktop-device partnership, see [MSDN-ADDP].

2.2.3.75 Get

The **Get** element is a required child element of the **RightsManagementInformation** element in **Settings** command **RightsManagementInformation** requests and responses, an optional child element of the **Oof** element in **Settings** command **Oof** requests and responses, and a required child element of the **UserInformation** element in **Settings** command **UserInformation** requests and responses.

In requests, the **Get** element enables the client to retrieve rights management information settings, OOF settings, or user information settings from the server. In responses, the **Get** element contains rights management information settings, **OOF** information, or user information settings retrieved from the server.

Elements referenced in this section are defined in the **Settings** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command **RightsManagementInformation** request (section 2.2.2.16)	**RightsManagementInformation** (section 2.2.3.143)	None	None	1…1 (required)
Settings command **RightsManagementInformation** response	**RightsManagementInformation**	**rm:RightsManagementTemplates** ([MS-ASRM] section 2.2.2.17)	container ([MS-ASDTYPE] section 2.2)	1…1 (required)
Settings command **Oof** request	**Oof** (section 2.2.3.112)	**BodyType** (section 2.2.3.17)	container	0…1 (optional)
Settings command **Oof** response	**Oof**	**OofState** (section 2.2.3.114) **StartTime** (section 2.2.3.161.2) **EndTime** (section 2.2.3.58.2) **OofMessage** (section 2.2.3.113)	container	0…1 (optional)
Settings command **UserInformation** request	**UserInformation** (section 2.2.3.176)	None	None	1…1 (required)
Settings command **UserInformation** response	**UserInformation**	**EmailAddresses**<39> (section 2.2.3.54)	container	1…1 (required)

Command request/response	Parent elements	Child elements	Data type	Number allowed
		Accounts (section 2.2.3.5)		

In a **Settings** command **Oof** request, the client specifies the body type (**BodyType** element) to be retrieved and the server will return all OOF settings and messages for that body type.

2.2.3.76   GetChanges

The **GetChanges** element is an optional child element of the **Collection** element in **Sync** command requests that requests that the server include in its response any pending changes to the collection that is specified by the **ServerId** element (section 2.2.3.151.6). If there have been changes since the last synchronization, the server response includes a **Commands** element (section 2.2.3.32) that contains additions, deletions, and changes.

All elements referenced in this section are defined in the **AirSync** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Sync command request (section 2.2.2.19)	**Collection** (section 2.2.3.29.2)	None	**boolean** ([MS-ASDTYPE] section 2.1)	0...1 (optional)

The **GetChanges** element appears only in requests to the server from the client.

If the client does not want the server changes returned, the request MUST include the **GetChanges** element with a value of 0 (**FALSE**). A value of 1 (**TRUE**) indicates that the client wants the server changes to be returned. A value of 1 (**TRUE**) is assumed when the **GetChanges** element is empty.

When the **GetChanges** element is not present in the request, the behavior depends on the value of the **SyncKey** element, as specified in section 2.2.3.166.4. If the **SyncKey** element has a value of 0, then the request is handled as if the **GetChanges** element were set to 0 (**FALSE**). If the **SyncKey** element has a non-zero value, then the request is handled as if the **GetChanges** element were set to 1 (**TRUE**).

A **Status** element (section 2.2.3.162.16) value of 4 is returned if the **GetChanges** element is present and empty or set to 1 (**TRUE**) when the **SyncKey** element value is 0 (zero). No error is returned if the **GetChanges** element is absent or set to 0 (**FALSE**) when the **SyncKey** value is 0 (zero).

2.2.3.77   GetItemEstimate

The **GetItemEstimate** element is a required element in **GetItemEstimate** command requests and **GetItemEstimate** command responses that identifies the body of the HTTP **POST** as containing a **GetItemEstimate** command (section 2.2.2.7). It is the top-level element in the XML stream.

All elements referenced in this section are defined in the **GetItemEstimate** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
GetItemEstimate command request	None	**Collections** (section 2.2.3.31.1)	**container** ([MS-ASDTYPE] section 2.2)	1…1 (required)
GetItemEstimate command response	None	**Response** (section 2.2.3.140.2)	**container**	1…1 (required)

2.2.3.78 GreaterThan

The **GreaterThan** element is an optional child element of the **And** element in **Search** command requests that contains elements that specify a property and a value that are compared for a "greater than" **condition** during a search.

Elements referenced in this section are defined in the **Search** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command request (section 2.2.2.14)	**And** (section 2.2.3.10)	**email:DateReceived** (section 2.2.3.51) **Value** (section 2.2.3.180)	**container** ([MS-ASDTYPE] section 2.2)	0…1 (optional)

The result of including more than one **GreaterThan** element in a **Search** command request is undefined. The server SHOULD return a protocol status error in response to such a command request.

The **GreaterThan** element is supported only in mailbox searches. It is not supported for document library searches. The comparison is made between the value of the **Value** element and the date that a mailbox item was received. The **email:DateReceived** element MUST be present before the **Value** element.

Typically, this element is used to filter results by the date on which they were received so that the date received is greater than the specified value.

If the **GreaterThan** element is included as a child element of any element other than the **And** element (section 2.2.3.10), the server responds with a **Status** element (section 2.2.3.162.12) value of 8 (SearchTooComplex).

2.2.3.79 HeartbeatInterval

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

- **Ping** command requests and responses (section 2.2.2.11)

- **Sync** command requests (section 2.2.2.19)

The definition of the **HeartbeatInterval** element differs according to the context in which it is used. For more details, section 2.2.3.79.1 and section 2.2.3.79.2.

2.2.3.79.1   HeartbeatInterval (Ping)

The **HeartbeatInterval** element is a child element of the **Ping** element in **Ping** command requests and responses. In **Ping** command requests, it specifies the length of time, in seconds, that the server SHOULD wait before sending a response if no new items are added to the specified set of folders, as specified in section 3.1.5.6. The **HeartbeatInterval** element is also returned by the server with a status code of 5 and specifies either the minimum or maximum interval that is allowed when the client has requested a heartbeat interval that is outside the acceptable range.

All elements referenced in this section are defined in the **Ping** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Ping command request (section 2.2.2.11)	**Ping** (section 2.2.3.126)	None	**integer** ([MS-ASDTYPE] section 2.5)	1 (required in first request only)
Ping command response	**Ping**	None	**integer**	0…1 (optional)

The **HeartbeatInterval** element is only required in the first **Ping** command request from a device by a given user. The server then caches the heartbeat interval value so that in later requests the **HeartbeatInterval** element is necessary only if the client is changing the interval.

2.2.3.79.2   HeartbeatInterval (Sync)

The **HeartbeatInterval** is an optional child element of the **Sync** element in **Sync** command requests that specifies the number of seconds that the server SHOULD delay a response if no new items are added to the included folders, as specified in section 3.1.5.4.

All elements referenced in this section are defined in the **AirSync** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Sync command request (section 2.2.2.19)	**Sync** (section 2.2.3.165)	None	**integer** ([MS-ASDTYPE] section 2.5)	0…1 (optional)

Valid values for **HeartbeatInterval** are 60 through 3540 seconds (59 minutes). When the client requests an interval that is outside the acceptable range, the server will send a response that includes a **Status** element (section 2.2.3.162.16) value of 14 and a **Limit** element.

Either the **HeartbeatInterval** element or the **Wait** element (section 2.2.3.182) can be included in a **Sync** command request, but not both. If both elements are included, the server response will contain a **Status** element value of 4.

2.2.3.80   HomePhone

The **HomePhone** element in the **GAL** namespace is an optional child element of the **search:Properties** element in **Search** command responses. The **HomePhone** element is only present in response to **Search** command requests that contain a **search:Name** element with a value of "GAL", as specified in section 2.2.3.110.2. It contains the home phone number of a recipient in the GAL that matched the search criteria from the corresponding **Search** command request.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command response (section 2.2.2.14)	**Properties** (section 2.2.3.128.2)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

2.2.3.81 Id

The **Id** element is a required child element of the **Folder** element in **Ping** command requests that specifies the server ID of the folder to be monitored.

Elements referenced in this section are defined in the **Ping** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
Ping command request (section 2.2.2.11)	**Folder** (section 2.2.3.66)	None	**string** ([MS-ASDTYPE] section 2.6)	1…1 (required)

The server ID of the folder is obtained from the **folderhierarchy:ServerId** element (section 2.2.3.151) of a previous **FolderSync** command response (section 2.2.2.4) or **FolderCreate** command response (section 2.2.2.2). The **Id** element value can be up to 64 characters in length.

2.2.3.82 IMEI

The **IMEI** element is an optional child element of the **Set** element in **Settings** command requests that specifies a 15-character code that MUST uniquely identify a device.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command **DeviceInformation** request (section 2.2.2.16)	**Set** (section 2.2.3.152)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

The **IMEI** element value can be up to 1024 characters in length. The server does not validate the format of the **IMEI** element value.

The device ID parameter that is currently included in the request URL is not precisely defined; protocol implementers are free to populate the field as they want. To enable workable inventory-type report generation, an ID that uniquely identifies a device in the space of all devices is required. The **IMEI** element satisfies this requirement.

2.2.3.83 InstanceId

The **InstanceId** element is used in the following command requests:

- **MeetingResponse** command requests (section 2.2.2.9)

- **SmartForward** command requests (section 2.2.2.17)

- **SmartReply** command requests (section 2.2.2.18)

The definition of the **InstanceId** element differs according to the context in which it is used. For more details, see section 2.2.3.83.1 and section 2.2.3.83.2.

2.2.3.83.1 InstanceId (MeetingResponse)

The **InstanceId** element<40> is an optional child element of the **Request** element in **MeetingResponse** command requests that specifies the instance of the recurring meeting to be modified.

Elements referenced in this section are defined in the **MeetingResponse** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
MeetingResponse command request (section 2.2.2.9)	**Request** (section 2.2.3.137.2)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

The **InstanceId** element contains the start time of the appointment or meeting instance to be modified. If the **InstanceId** element is not included in the **MeetingResponse** request, then the action is to be taken on every instance of the recurring item. The **InstanceId** element can specify the start time of an exception to a recurring appointment or meeting. The **InstanceId** element can be used with the **search:LongId** element (section 2.2.3.89.2) to identify a calendar item, or it can be used with the **CollectionId** (section 2.2.3.30.3) and **RequestId** (section 2.2.3.138) elements to identify a calendar item.

The format of the **InstanceId** value is a **string** in **dateTime** ([MS-ASDTYPE] section 2.3) format with the punctuation separators, for example, 2010-04-08T18:16:00.000Z. If the **InstanceId** element value specified is not in the proper format, the server responds with a **Status** element (section 2.2.3.162.8) value of 104. If the **InstanceId** element value specifies a non-recurring meeting, the server responds with a **Status** element value of 146.

2.2.3.83.2 InstanceId (SmartForward and SmartReply)

The **InstanceId** element is an optional child element of the **Source** element in **SmartForward** command requests and **SmartReply** command requests that specifies the instance of a recurrence for the source item. If the **InstanceId** element is present, both the **FolderId** (section 2.2.3.69) and **ItemId** (section 2.2.3.84) elements SHOULD be present.

All elements referenced in this section are defined in the **ComposeMail** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
SmartForward command request (section 2.2.2.17)	**Source** (section 2.2.3.158)	None	**datetime** ([MS-ASDTYPE] section 2.3)	0…1 (optional)
SmartReply command request (section 2.2.2.18)	**Source**	None	**datetime**	0…1 (optional)

The format of the **InstanceId** element is a **dateTime** value that includes the punctuation separators. For example, 2010-03-20T22:40:00.000Z.

2.2.3.84 ItemId

The **ItemId** element is an optional child element of the **Source** element in **SmartForward** command requests and **SmartReply** command requests that specifies the item ID for the source message, which is returned in the **Sync** command response message (section 2.2.2.19). If the **ItemId** element is present, the **FolderId** element (section 2.2.3.69) MUST also be present if the message being forwarded or replied to is stored in a folder other than the Inbox folder. The **ItemId** element value can be up to 64 characters in length.

All elements referenced in this section are defined in the **ComposeMail** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
SmartForward command request (section 2.2.2.17)	**Source** (section 2.2.3.158)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)
SmartReply command request (section 2.2.2.18)	**Source**	None	**string**	0…1 (optional)

2.2.3.85 ItemOperations

The **ItemOperations** element is a required element in **ItemOperations** command requests and **ItemOperations** command responses that identifies the body of the HTTP **POST** as containing an **ItemOperations** command (section 2.2.2.8). It is the top-level element in the XML stream.

All elements referenced in this section are defined in the **ItemOperations** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ItemOperations command request	None	**EmptyFolderContents** (section 2.2.3.55) **Fetch** (section 2.2.3.63.1) **Move** (section 2.2.3.107.1)	**container** ([MS-ASDTYPE] section 2.2)	1…1 (required)
ItemOperations command response	None	**Status** (section 2.2.3.162.7) **Response** (section 2.2.3.140.3)	**container**	1…1 (required)

2.2.3.86 LastName

The **LastName** element in the **GAL** namespace is an optional child element of the **search:Properties** element in **Search** command responses. The **LastName** element is only present in response to **Search** command requests that contain a **search:Name** element with a value of "GAL", as specified in section 2.2.3.110.2. It contains the last name of a recipient in the GAL that matched the search criteria from the corresponding **Search** command request.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command response (section 2.2.2.14)	**Properties** (section	None	**string** ([MS-ASDTYPE] section	0…1 (optional)

Command request/response	Parent elements	Child elements	Data type	Number allowed
	2.2.3.128.2)		2.6)	

2.2.3.87 LessThan

The **LessThan** element is an optional child element of the **And** element in **Search** command requests that contains elements that specify a property and a value that are compared for a "less than" condition during a search.

Elements referenced in this section are defined in the **Search** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command request (section 2.2.2.14)	**And** (section 2.2.3.10)	**email:DateReceived** (section 2.2.3.51) **Value** (section 2.2.3.180)	**container** ([MS-ASDTYPE] section 2.2)	0...1 (optional)

The result of including more than one **LessThan** element in a **Search** command request is undefined. The server SHOULD return a protocol status error in response to such a command request.

The **LessThan** element is supported only in mailbox searches. It is not supported for document library searches. The comparison is made between the value of the **Value** element and the date that a mailbox item was received. The **email:DateReceived** element MUST be present before the **Value** element.

Typically, this element is used to filter results by the date on which they were received so that the date received is less than the specified value.

If the **LessThan** element is included as a child element of any element other than the **And** element (section 2.2.3.10), the server responds with a **Status** element (section 2.2.3.162.12) value of 8 (SearchTooComplex).

2.2.3.88 Limit

The **Limit** element is an optional child element of the **Sync** element in **Sync** command responses that specifies either the maximum number of collections that can be synchronized or the maximum/minimum value that is allowed for the **Wait** interval (section 2.2.3.182) or **HeartbeatInterval** interval (section 2.2.3.79.2).

All elements referenced in this section are defined in the **AirSync** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Sync command response (section 2.2.2.19)	**Sync** (section 2.2.3.165)	None	**integer** ([MS-ASDTYPE] section 2.5)	0...1 (optional)

The **Limit** element is returned in a response with a status code of 14 or 15. The value of the **Status** element (section 2.2.3.162.16) indicates whether the limit applies to the **Wait** interval or **HeartbeatInterval** interval or the number of collections, as follows:

- A status code 14 indicates that the **Limit** element specifies the minimum or maximum wait-interval that is acceptable. When the value of the **Wait** element or **HeartbeatInterval** element is outside of the acceptable range, the server responds with the closest acceptable value. If a **Wait** element value of less than 1 is sent, the server returns a **Limit** element value of 1, indicating the minimum value of the **Wait** element is 1. If a **Wait** element value greater than 59 is sent, the server returns a **Limit** element value of 59, indicating the maximum value of the **Wait** element is 59. If a **HeartbeatInterval** element value of less than 60 is sent, the server returns a **Limit** element value of 60, indicating the minimum value of the **HeartbeatInterval** element is 60. If a **HeartbeatInterval** element value greater than 3540 is sent, the server returns a **Limit** element value of 3540, indicating the maximum value of **HeartbeatInterval** element is 3540.

- A status code 15 indicates that the **Limit** element specifies the maximum number of collections that can be synchronized.

2.2.3.89 LongId

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

- **ItemOperations** command requests (section 2.2.2.8)

- **MeetingResponse** command requests (section 2.2.2.9)

- **Search** command responses (section 2.2.2.14)

- **SmartForward** command requests (section 2.2.2.17)

- **SmartReply** command requests (section 2.2.2.18)

The definition of the **LongId** element differs according to the context in which it is used. For more details, see section 2.2.3.89.1, section 2.2.3.89.3, and section 2.2.3.89.4.

2.2.3.89.1 LongId (ItemOperations)

The **search:LongId** element is an optional child element of the **Fetch** element in **ItemOperations** command requests that specifies a unique identifier that was assigned by the server to each result returned by a previous **Search** response. The **search:LongId** element value can be up to 256 characters in length.

Elements referenced in this section are defined in the **ItemOperations** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
ItemOperations command request (section 2.2.2.8) fetch operation	**Fetch** (section 2.2.3.63.1)	None	**string** ([MS-ASDTYPE] section 2.6)	0...1 (optional)

2.2.3.89.2 LongId (MeetingResponse)

The **LongId** element is an optional child element of the **Source** element in **MeetingResponse** command requests. This element specifies the long ID for the source meeting request, which is returned in the **Search** command response message (section 2.2.2.14). If the **LongId** element is present, the **CollectionId** (section 2.2.3.30.3), **InstanceId** (section 2.2.3.83.1), and **RequestId** (section 2.2.3.138) elements are not present. The **LongId** element value can be up to 256 characters in length.

All elements referenced in this section are defined in the **ComposeMail** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
MeetingResponse command request (section 2.2.2.9)	**Request** (section 2.2.3.137.2)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

2.2.3.89.3 LongId (Search)

The **LongId** element is an optional child element of the **Result** element in **Search** command responses that specifies a unique identifier that is assigned by the server to each result set that is returned.

All elements referenced in this section are defined in the **Search** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command response (section 2.2.2.14)	**Result** (section 2.2.3.142.2)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

The value of the **LongId** element can be used as the long ID specified in the **ItemOperations** command request (section 2.2.2.8), the **SmartReply** command request (section 2.2.2.18), the **SmartForward** command request (section 2.2.2.17), or the **MeetingResponse** command request (section 2.2.2.9) to reference the result set.

The client MUST store the value of **LongId** as an opaque **string** of up to 256 characters.

2.2.3.89.4 LongId (SmartForward and SmartReply)

The **LongId** element is an optional child element of the **Source** element in **SmartForward** command requests and **SmartReply** command requests that specifies the long ID for the source message, which is returned in the **Search** command response message (section 2.2.2.14). If the **LongId** element is present, the **FolderId** (section 2.2.3.69), **ItemId** (section 2.2.3.84), and **InstanceId** (section 2.2.3.83.2) elements are not present. The **LongId** element value can be up to 256 characters in length.

All elements referenced in this section are defined in the **ComposeMail** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
SmartForward command	**Source** (section	None	**string** ([MS-	0…1

Release: February 10, 2014

Command request/response	Parent elements	Child elements	Data type	Number allowed
request (section 2.2.2.17)	2.2.3.158)		ASDTYPE] section 2.6)	(optional)
SmartReply command request (section 2.2.2.18)	**Source**	None	**string**	0…1 (optional)

2.2.3.90 MaxAmbiguousRecipients

The **MaxAmbiguousRecipients** element is an optional child element of the **Options** element in **ResolveRecipients** command requests that limits the number of suggestions that are returned for each ambiguous recipient node in the response.

All elements referenced in this section are defined in the **ResolveRecipients** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ResolveRecipients command request (section 2.2.2.13)	**Options** (section 2.2.3.115.3)	None	**integer** ([MS-ASDTYPE] section 2.5)	0…1 (optional)

The value of the **MaxAmbiguousRecipients** element is limited to a range of 0–9999. Each ambiguous recipient node receives only this many suggestions and no more. The recipient count, returned in the **RecipientCount** element (section 2.2.3.133), can be used by the client to determine the total number of suggestions available for that recipient.

2.2.3.91 MaxCertificates

The **MaxCertificates** element is an optional child element of the **Options** element in **ResolveRecipients** command requests that limits the total number of certificates that are returned by the server.

All elements referenced in this section are defined in the **ResolveRecipients** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ResolveRecipients command request (section 2.2.2.13)	**Options** (section 2.2.3.115.3)	None	**integer** ([MS-ASDTYPE] section 2.5)	0…1 (optional)

The value of the **MaxCertificates** element is limited to a range of 0–9999. This limit ensures that no individual recipient receives an incomplete set of certificates. If the **MaxCertificates** limit is reached while enumerating certificates for an **address list**, that address list will not get back any certificates and a **Status** element (section 2.2.3.162.11) value of 8 is returned. The client can then use the certificate count returned to determine the number of certificates that are available for that recipient node.

2.2.3.92 MaxFolders

The **MaxFolders** element is an optional child element of the **Ping** element in **Ping** command responses that specifies the maximum number of folders that can be monitored.

All elements referenced in this section are defined in the **Ping** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Ping command response (section 2.2.2.11)	**Ping** (section 2.2.3.126)	None	**integer** ([MS-ASDTYPE] section 2.5)	0...1 (optional)

The **MaxFolders** element is returned in a response with a status code of 6.

2.2.3.93 MaxItems

The **MaxItems** element is used in the following command requests:

- **GetItemEstimate** command requests (section 2.2.2.7)

- **Sync** command requests (section 2.2.2.19)

The definition of the **MaxItems** element differs according to the context in which it is used. For more details, see section 2.2.3.93.1 and section 2.2.3.93.2.

2.2.3.93.1 MaxItems (GetItemEstimate)

The **airsync:MaxItems** element<41> is an optional child element of the **airsync:Options** element in **GetItemEstimate** command requests that specifies the maximum number of items to include in the response. This element can only be included in a request when the **CollectionId** element (section 2.2.3.30.1) is set to "RI" to specify a recipient information store; otherwise, the server will respond with a status 2 error.

Elements referenced in this section are defined in the **GetItemEstimate** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
GetItemEstimate command request (section 2.2.2.7)	**airsync:Options** (section 2.2.3.115.1)	None	**integer** ([MS-ASDTYPE] section 2.5)	0...1 (optional)

Including **airsync:MaxItems** when the **CollectionId** element is set to anything other than "RI" results in an invalid XML error, **Status** element (section 2.2.3.162.6) value of 2.

The result of including more than one **airsync:MaxItems** element as the child of the **airsync:Options** element is undefined. The server MAY return a protocol status error in response to such a command request.

2.2.3.93.2 MaxItems (Sync)

The **MaxItems** element<42> is an optional child element of the **Options** element in **Sync** command requests that specifies the maximum number of recipients (that is, the top N most frequently used recipients) to keep synchronized from within the recipient information cache. This element MUST only be included in a request when the value of the **CollectionId** element (section 2.2.3.30.5) maps to the recipient information cache. The value of the **MaxItems** element does not specify the maximum estimate of additions and deletions to make to the recipient information cache, it only specifies the number of recipients to keep synchronized. A complete replacement of

each recipient would be double the number of **MaxItems** or items in the cache, as each recipient update requires a deletion and an addition.

All elements referenced in this section are defined in the **AirSync** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Sync command request (section 2.2.2.19)	**Options** (section 2.2.3.115.5)	None	**integer** ([MS-ASDTYPE] section 2.5)	0...1 (optional)

The result of including more than one **MaxItems** element as the child element of the **Options** element is undefined. The server MAY return a protocol status error in response to such a command request.

2.2.3.94   MaxPictures

The **MaxPictures** element<43> is used in the following command requests:

- **ResolveRecipients** command requests (section 2.2.2.13)

- **Search** command requests (section 2.2.2.14)

The definition of the **MaxPictures** element differs according to the context in which it is used. For more details, see section 2.2.3.93.1 and section 2.2.3.93.2.

2.2.3.94.1   MaxPictures (ResolveRecipients)

The **MaxPictures** element<44> is an optional child element of the **Picture** element in **ResolveRecipients** command requests that limits the number of contact photos returned in the server response.

All elements referenced in this section are defined in the **ResolveRecipients** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ResolveRecipients command request (section 2.2.2.13)	**Picture** (section 2.2.3.125.1)	None	**integer** ([MS-ASDTYPE] section 2.5)	0…..1 (optional)

The server returns the first N results that have contact photos, where N is the value of the **MaxPictures** element. After the **MaxPictures** limit is reached, the server returns **Status** element (section 2.2.3.162) value 173 (NoPicture) if the contact has no photo, **Status** element value 174 (PictureTooLarge) if the contact's photo is too large, or **Status** element value 175 (PictureLimitReached) if the contact has a photo but the **MaxPictures** limit was reached.

Note that the **MaxPictures** element identifies the number of contact photos returned per query. Therefore, if the client includes three recipients to resolve and sets the **MaxPictures** element value to 3 in the **ResolveRecipients** command request, a maximum of 9 contact photos can be returned.

2.2.3.94.2   MaxPictures (Search)

The **MaxPictures** element<45> is an optional child element of the **Picture** element in **Search** command requests that limits the number of contact photos returned in the server response.

All elements referenced in this section are defined in the **Search** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command request (section 2.2.2.14)	**Picture** (section 2.2.3.125.2)	None	**integer** ([MS-ASDTYPE] section 2.5)	0…..1 (optional)

The server returns the first N results that have contact photos, where N is the value of the **MaxPictures** element. After the **MaxPictures** limit is reached, the server returns **Status** element (section 2.2.3.162) value 173 (NoPicture) if the contact has no photo, **Status** element value 174 (PictureTooLarge) if the contact's photo is too large, or **Status** element value 175 (PictureLimitReached) if the contact has a photo but the **MaxPictures** limit was reached.

2.2.3.95 MaxSize

The **MaxSize** element<46> is used in the following command requests:

- **ResolveRecipients** command requests (section 2.2.2.13)

- **Search** command requests (section 2.2.2.14)

The definition of the **MaxSize** element differs according to the context in which it is used. For more details, see section 2.2.3.95.1 and section 2.2.3.95.2.

2.2.3.95.1 MaxSize (ResolveRecipients)

The **MaxSize** element<47> is an optional child element of the **Picture** element in **ResolveRecipients** command requests that limits the size of contact photos returned in the server response.

All elements referenced in this section are defined in the **ResolveRecipients** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ResolveRecipients command request (section 2.2.2.13)	**Picture** (section 2.2.3.125.1)	None	**integer** ([MS-ASDTYPE] section 2.5)	0…1 (optional)

The maximum value of the **MaxSize** element is 100 KB or 102400 bytes.

The **MaxSize** element specifies the maximum size of an individual contact photo that is returned in the response, in bytes. The **MaxPictures** element (section 2.2.3.94) specifies the maximum number of contact photos to return in the server response.

2.2.3.95.2 MaxSize (Search)

The **MaxSize** element<48> is an optional child element of the **Picture** element in **Search** command requests that limits the size of contact photos returned in the server response.

All elements referenced in this section are defined in the **Search** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command request (section 2.2.2.14)	**Picture** (section 2.2.3.125.2)	None	**integer** ([MS-ASDTYPE] section 2.5)	0…1 (optional)

The maximum value of the **MaxSize** element is 100 KB or 102400 bytes.

The **MaxSize** element specifies the maximum size of an individual contact photo that is returned in the response, in bytes. The **MaxPictures** element (section 2.2.3.94) specifies the maximum number of contact photos to return in the server response.

2.2.3.96 MeetingResponse

The **MeetingResponse** element is a required element in **MeetingResponse** command requests and **MeetingResponse** command responses that identifies the body of the HTTP **POST** as containing a **MeetingResponse** command (section 2.2.2.9). It is the top-level element in the XML stream.

All elements referenced in this section are defined in the **MeetingResponse** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
MeetingResponse command request	None	**Request** (section 2.2.3.137.2)	**container** ([MS-ASDTYPE] section 2.2)	1…1 (required)
MeetingResponse command response	None	**Result** (section 2.2.3.142.1)	**container**	1…1 (required)

2.2.3.97 MergedFreeBusy

The **MergedFreeBusy** element<49> is an optional child element of the **Availability** element in **ResolveRecipients** command responses that specifies the free/busy information for the users or distribution list identified in the request.

All elements referenced in this section are defined in the **ResolveRecipients** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ResolveRecipients command response (section 2.2.2.13)	**Availability** (section 2.2.3.16)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

The **MergedFreeBusy** element value string has a maximum length of 32 KB. To retrieve more than 32 KB of availability data, the client MUST issue a new request with the appropriate start time and end time.

Each digit in the **MergedFreeBusy** element value string indicates the free/busy status for the user or distribution list for every 30 minute interval. The following table lists the valid values.

Value	Meaning
0	Free
1	Tentative
2	Busy
3	Out of Office (OOF)
4	No data

A string value of "32201" would represent that this user or group of users is out of the office for the first 30 minutes, busy for the next hour, free for 30 minutes, and then has a tentative meeting for the last 30 minutes. If the user or group of users has a change in availability that lasts less than the interval value of 30 minutes, the availability value with the higher digit value is assigned to the whole interval period. For example, if a user has a 25 minutes of free time (value 0) followed by 5 minutes of busy time (value 2), the 30 minute interval is assigned a value of 2 in the server response.

The server determines the number of digits to include in the **MergedFreeBusy** element by dividing the time interval specified by the **StartTime** element (section 2.2.3.161.1) value and the **EndTime** element (section 2.2.3.58.1) value by 30 minutes, and rounding the result up to the next integer.

The **MergedFreeBusy** element value string is populated from the **StartTime** element value onwards, therefore the last digit represents between a millisecond and 30 minutes. A query for data from 13:00:00 to 13:30:00 returns a single digit but a query from 12:59:59 to 13:30:00 or 13:00:00 to 13:30:01 returns two digits.

Any appointment that ends inside a second of the interval requested shall impact the digit representing that timeframe. For example, given a calendar that contains a 5 minute OOF appointment from 12:00 to 12:05, and is free the rest of the day, queries would result in the following:

▪ If a query is made for 12:00:00 to 13:00:00, the result is "30", where each digit represents exactly 30 minutes.

▪ If a query is made for 12:04:59 to 13:00:00, the result is "30", where the "0" maps to 12:34:59 to 13:00:00.

▪ If a query is made for 12:05:00 to 13:00:00, the result is "00" where the second 0 maps the last 25 minutes of the interval.

The client MUST consider daylight saving time transitions and might need to add or remove time intervals from the **MergedFreeBusy** element value string, as there are days that have more or less than 24 hours.

If the **Availability** element is included in the response, the response MUST also include the **Status** element (section 2.2.3.162.11). The **MergedFreeBusy** element is also included if the **Status** element value indicates success.

2.2.3.98   Message

The **Message** element is an optional child element of the **Error** element in **Autodiscover** command responses that contains the error string localized using the **Culture** (section 2.2.3.38) value specified in the **Response** element, enabling the client to display error status to the end-user.

All elements referenced in this section are defined in the http://schemas.microsoft.com/exchange/autodiscover/mobilesync/requestschema/2006 namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Autodiscover command response (section 2.2.2.1)	**Error** (section 2.2.3.60)	None	**string** ([MS-ASDTYPE] section 2.6)	0...1 (optional)

2.2.3.99 Mime

The **Mime** element is a required child element of the **SendMail** element in **SendMail** command requests, a required child element of the **SmartForward** element in **SmartForward** command requests, and a required child element of the **SmartReply** element in **SmartReply** command requests. In all contexts, it contains the MIME-encoded message.

All elements referenced in this section are defined in the **ComposeMail** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
SendMail command request (section 2.2.2.15)	**SendMail** (section 2.2.3.148)	None	**Byte array**	1...1 (required)
SmartForward command request (section 2.2.2.17)	**SmartForward** (section 2.2.3.154)	None	**Byte array**	1...1 (required)
SmartReply command request (section 2.2.2.18)	**SmartReply** (section 2.2.3.155)	None	**Byte array**	1...1 (required)

The **Mime** element content is transferred as an opaque BLOB within the WBXML tags, as specified in [WBXML1.2].

If the message contains a meeting request, the **Mime** element contains the details of meeting in iCalendar format [MS-OXCICAL] or **Transport Neutral Encapsulation Format (TNEF)** format [MS-OXTNEF]. As specified in [RFC2447] section 3.4, iCalendar meeting requests have a content type of "text/calendar" with the **method** parameter set to "REQUEST".

2.2.3.100 MIMESupport

The **MIMESupport** element is used in the following command requests:

- **ItemOperations** command requests (section 2.2.2.8)

- **Search** command requests (section 2.2.2.14)

- **Sync** command requests (section 2.2.2.19)

The definition of the **MIMESupport** element differs according to the context in which it is used. For more details, see section 2.2.3.100.1, section 2.2.3.100.2, and section 2.2.3.100.3.

2.2.3.100.1   MIMESupport (ItemOperations)

The **airsync:MIMESupport** element is an optional child element of the the **Options** element in **ItemOperations** command requests that enables MIME support for email items that are sent from the server to the client. For an example, see section 4.10.2.

Elements referenced in this section are defined in the **ItemOperations** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
ItemOperations command request (section 2.2.2.8) fetch operation	**Options** (section 2.2.3.115.2)	None	**unsignedByte** ([MS-ASDTYPE] section 2.7)	0...1 (optional)

The following table lists the valid values for this element. If the **MIMESupport** element is not present, the default value of 0 is used.

Value	Meaning
0	Never send MIME data.
1	Send MIME data for S/MIME messages only. Send regular body for all other messages.
2	Send MIME data for all messages. This flag could be used by clients to build a more rich and complete Inbox solution.

To support fetching of the full S/MIME message, the **ItemOperations** command request MUST include the following elements in the **Options** element:

- The **airsync:MIMESupport** element (section 2.2.3.100.3) to indicate to the server to return MIME for S/MIME-only messages, all messages, or no messages.

- The **airsyncbase:BodyPreference** element ([MS-ASAIRS] section 2.2.2.7) with its child element, **Type** ([MS-ASAIRS] section 2.2.2.22.4), having a value of 4 to inform the server that the device can read the MIME binary large object (BLOB).

The server's response MUST include the **airsyncbase:Body** element ([MS-ASAIRS] section 2.2.2.4), which is a child element of the **Properties** element (section 2.2.3.128.1). The **airsyncbase:Body** element is a complex element and MUST contain the following child elements in an S/MIME response:

- The **airsyncbase:Type** element ([MS-ASAIRS] section 2.2.2.22.1) with a value of 4 to inform the device that the data is a MIME BLOB.

- The **airsyncbase:EstimatedDataSize** element ([MS-ASAIRS] section 2.2.2.12.2) to specify the rough total size of the data.

- The **airsyncbase:Data** element ([MS-ASAIRS] section 2.2.2.10.1) that contains the full MIME BLOB.

For more details about the **airsyncbase:Body** element or the **airsyncbase:BodyPreference** element, see [MS-ASAIRS] section 2.2.2.4 or [MS-ASAIRS] section 2.2.2.7, respectively.

2.2.3.100.2 MIMESupport (Search)

The **airsync:MIMESupport** element is an optional child element of the **Options** element in **Search** command requests that enables MIME support for email items that are sent from the server to the client.

Elements referenced in this section are defined in the **Search** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command request (section 2.2.2.14)	**Options** (section 2.2.3.115.4)	None	**integer** ([MS-ASDTYPE] section 2.5)	0...1 (optional)

The result of including more than one **airsync:MIMESupport** element in a **Search** command request is undefined. The server MAY return a protocol status error in response to such a command request.

The following table lists the valid values for the element. If the **MIMESupport** element is not present, the default value of 0 is used.

Value	Meaning
0	Never send MIME data.
1	Send MIME data for S/MIME [RFC5751] messages only. Send regular body (non S/MIME) data for all other messages.
2	Send MIME data for all messages. This flag could be used by clients to build a more rich and complete Inbox solution.

The **Search** response can include the S/MIME BLOB of a signed/encrypted message.

If the **airsync:MIMESupport** element is set to 1 or 2 in the **Search** request:

- The child element of the **airsyncbase:BodyPreference** element ([MS-ASAIRS] section 2.2.2.7), the **Type** element ([MS-ASAIRS] section 2.2.2.22.4), SHOULD be included in the **Search** request, containing a value of 4 to inform the server that the device can read the MIME BLOB.

- The response from the server MUST include the **airsyncbase:Body** element ([MS-ASAIRS] section 2.2.2.4), which is a child element of the **Properties** element (section 2.2.3.128.2). The **airsyncbase:Body** element is a complex element and MUST contain the following child elements in an S/MIME **Search** response:

  - The **airsyncbase:Type** element ([MS-ASAIRS] section 2.2.2.22.1) with a value of 4 to inform the device that the data is a MIME BLOB.

  - The **airsyncbase:EstimatedDataSize** element ([MS-ASAIRS] section 2.2.2.12.2) to specify the rough total size of the data.

  - The **airsyncbase:Truncated** element ([MS-ASAIRS] section 2.2.2.20.1) to indicate whether the MIME BLOB is truncated.

  - The **airsyncbase:Data** element ([MS-ASAIRS] section 2.2.2.10.1) that contains the full MIME BLOB.

For more details about the **airsyncbase:Body** element or the **airsyncbase:BodyPreference** element, see [MS-ASAIRS] section 2.2.2.4 or [MS-ASAIRS] section 2.2.2.7, respectively.

2.2.3.100.3 MIMESupport (Sync)

The **MIMESupport** element is an optional child element of the **Options** element in **Sync** command requests that enables MIME support for email items that are sent from the server to the client.

Elements referenced in this section are defined in the **AirSync** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
Sync command request (section 2.2.2.19)	**Options** (section 2.2.3.115.5)	None	**unsignedByte** ([MS-ASDTYPE] section 2.7)	0…1 (optional)

The following table shows valid values for the element. If the **MIMESupport** element is not present, the default value of 0 is used.

Value	Meaning
0	Never send MIME data.
1	Send MIME data for S/MIME messages only. Send regular body for all other messages.
2	Send MIME data for all messages. This flag could be used by clients to build a more rich and complete Inbox solution.

The client MUST send a maximum of one **MIMESupport** element. The result of including more than one **MIMESupport** element as the child element of the **Options** element is undefined. The server MAY return a protocol status error in response to such a command request.

The **Sync** request MUST include the following elements in the **Options** element when handling S/MIME content:

▪ The **MIMESupport** element to tell the server to return MIME for S/MIME-only/All/None messages.

▪ The **airsyncbase:BodyPreference** element ([MS-ASAIRS] section 2.2.2.7) with its child element, **Type** ([MS-ASAIRS] section 2.2.2.22.4), which contains a value of 4 to inform the server that the device can read the MIME BLOB.

When handling S/MIME content in the response, the server MUST include the **airsyncbase:Body** element ([MS-ASAIRS] section 2.2.2.4), which is a child of the **ApplicationData** element (section 2.2.3.11). The **airsyncbase:Body** element is a complex element and MUST contain the following child elements in an S/MIME synchronization response:

▪ The **airsyncbase:Type** element ([MS-ASAIRS] section 2.2.2.22.1) with a value of 4 to inform the device that the data is a MIME BLOB.

▪ The **airsyncbase:EstimatedDataSize** element ([MS-ASAIRS] section 2.2.2.12.2) to specify the rough total size of the data.

▪ The **airsyncbase:Truncated** element ([MS-ASAIRS] section 2.2.2.20.1) to indicate whether the MIME BLOB is truncated.

- The **airsyncbase:Data** element ([MS-ASAIRS] section 2.2.2.10.1) that contains the full MIME BLOB.

For more details about the **airsyncbase:Body** element or the **airsyncbase:BodyPreference** element, see [MS-ASAIRS] section 2.2.2.4 or [MS-ASAIRS] section 2.2.2.7, respectively.

2.2.3.101 MIMETruncation

The **MIMETruncation** element is an optional child element of the **Options** element in **Sync** command requests that specifies whether the MIME data of the email item SHOULD be truncated when it is sent from the server to the client.

All elements referenced in this section are defined in the **AirSync** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Sync command request (section 2.2.2.19)	**Options** (section 2.2.3.115.5)	None	**unsignedByte** ([MS-ASDTYPE] section 2.7)	0…1 (optional)

The following table lists valid values for the element.

Value	Meaning
0	Truncate all body text.
1	Truncate text over 4,096 characters.
2	Truncate text over 5,120 characters.
3	Truncate text over 7,168 characters.
4	Truncate text over 10,240 characters.
5	Truncate text over 20,480 characters.
6	Truncate text over 51,200 characters.
7	Truncate text over 102,400 characters.
8	Do not truncate; send complete MIME data.

The size of the truncated message returned in the response is not the exact size of the **MIMETruncation** element value; the size specified by the **MIMETruncation** element is an approximate value. This is because line feeds are treated as one character locally, but are counted as two characters during truncation.

The result of including more than one **MIMETruncation** element as the child element of the **Options** element is undefined. The server MAY return a protocol status error in response to such a command request.

2.2.3.102 MiniCertificate

The **MiniCertificate** element is an optional child element of the **Certificates** element in **ResolveRecipients** command responses that contains the mini-certificate BLOB that is encoded with base64 encoding.

All elements referenced in this section are defined in the **ResolveRecipients** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ResolveRecipients command response (section 2.2.2.13)	**Certificates** (section 2.2.3.23.1)	None	**string** ([MS-ASDTYPE] section 2.6) (encoded with base64 encoding)	0...1 per **Certificates** parent element

This element is returned only if the client specifies a value of 3 in the **CertificateRetrieval** element (section 2.2.3.22) in the **ResolveRecipients** command request and the resolved recipient has a valid S/MIME certificate.

2.2.3.103   MobileOperator

The **MobileOperator** element is an optional<50> child element of the **Set** element in **Settings** command requests that specifies the name of the mobile operator to which a mobile device is connected. The **MobileOperator** element value can be up to 1024 characters in length.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command **DeviceInformation** request (section 2.2.2.16)	**Set** (section 2.2.3.152)	None	**string** ([MS-ASDTYPE] section 2.6)	0...1 (optional)

2.2.3.104   MobilePhone

The **MobilePhone** element in the **GAL** namespace is an optional child element of the **search:Properties** element in **Search** command responses. The **MobilePhone** element is only present in response to **Search** command requests that contain a **search:Name** element with a value of "GAL", as specified in section 2.2.3.110.2. It contains the mobile phone number of a recipient in the GAL that matched the search criteria from the corresponding **Search** command request.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command response (section 2.2.2.14)	**Properties** (section 2.2.3.128.2)	None	**string** ([MS-ASDTYPE] section 2.6)	0...1 (optional)

2.2.3.105   Model

The **Model** element is an optional child element of the **Set** element in **Settings** command requests that specifies a name that generally describes the device of the client. The **Model** element value can be up to 1024 characters in length.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command **DeviceInformation** request (section 2.2.2.16)	**Set** (section 2.2.3.152)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

The descriptive name of the device can be any string that the client chooses, typically a general description of the device. For example, the name of the manufacturer, the model name, or the model number can be used. The server does not perform any validation of this string, so the client can submit any string.

2.2.3.106 MoreAvailable

The **MoreAvailable** element is an optional child element of the **Collection** element in **Sync** command responses that indicates there are more changes than the number that are requested in the **WindowSize** element (section 2.2.3.183).

All elements referenced in this section are defined in the **AirSync** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Sync command response (section 2.2.2.19)	**Collection** (section 2.2.3.29.2)	None	None	0…1 (optional)

The **MoreAvailable** 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 <MoreAvailable/> tag.

The **MoreAvailable** element appears only in responses that are sent from the server to the client. It appears only if the client request contained a **WindowSize** element and there are still changes to be returned to the client.

The server includes the **MoreAvailable** element in **Sync** responses that contain no additions, changes, or deletions when the server encounters elements external to the protocol.<51> If the client receives multiple Sync responses that contain the MoreAvailable element and fewer changes than requested by the WindowSize element value included in the Sync request the client SHOULD continue to send **Sync** requests to retrieve additional changes until no additional results are sent by the server, but MAY stop sending **Sync** requests if the Sync request and response loop is affecting network performance and synchronizing the client is of less importance than network performance.

The **MoreAvailable** element has no body. It is omitted if no additional changes are available. The maximum value for the **WindowSize** element is 512. The server interprets **WindowSize** element values above 512 and 0 (zero) as 512.

If the **WindowSize** element is omitted, the server behaves as if a **WindowSize** element with a value of 100 was submitted. The **MoreAvailable** element is returned by the server if there are more than 512 changes, regardless of whether the **WindowSize** element is included in the request.

2.2.3.107 Move

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

- **ItemOperations** command requests and responses (section 2.2.2.8)

- **MoveItems** command request (section 2.2.2.10)

The definition of the **Move** element differs according to the context in which it is used. For more details, see section 2.2.3.107.1 and section 2.2.3.107.2.

2.2.3.107.1   Move (ItemOperations)

The **Move** element is an optional child element of the **ItemOperations** element in **ItemOperations** command requests and an optional child element of the **Response** element in **ItemOperations** command responses. It identifies the body of the request or response as containing the operation that moves a given conversation.<52>

All elements referenced in this section are defined in the **ItemOperations** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ItemOperations command request (section 2.2.2.8)	**ItemOperations** (section 2.2.3.85)	**ConversationId** (section 2.2.3.35.1) **DstFldId** (section 2.2.3.49.1) **Options** (section 2.2.3.115.2)	**container** ([MS-ASDTYPE] section 2.2)	0…N (optional)
ItemOperations command response	**Response** (section 2.2.3.140.3)	**ConversationId** **Status** (section 2.2.3.162.7)	**container**	0…N (optional)

2.2.3.107.2   Move (MoveItems)

The **Move** element is a required child element of the **MoveItems** element in **MoveItems** command requests that contains elements that describe details of the items to be moved.

All elements referenced in this section are defined in the **Move** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
MoveItems command request (section 2.2.2.10)	**MoveItems** (section 2.2.3.109)	**SrcMsgId** (section 2.2.3.160) **SrcFldId** (section 2.2.3.159) **DstFldId** (section 2.2.3.49.2)	**container** ([MS-ASDTYPE] section 2.2)	1…N (required)

The **Move** element's child elements specify the item to be moved, the folder it's currently located in, and the folder it will be moved to.

2.2.3.108   MoveAlways

The **MoveAlways** element is an optional child element of the **Options** element in **ItemOperations** command requests that indicates whether to always move the specified conversation, including all future emails in the conversation, to the folder specified by the **DstFldId** element (section 2.2.3.49.1) value.<53>

All elements referenced in this section are defined in the **ItemOperations** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ItemOperations command request (section 2.2.2.8) **Move** operation	**Options** (section 2.2.3.115.2)	None	None	0…1 (optional)

The **MoveAlways** 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.

The **MoveAlways** element MUST be included in an **ItemOperations** request when performing a move operation on a conversation. A **Status** element (section 2.2.3.162.7) value of 155 is returned if the **MoveAlways** element is not included in the **ItemOperations** request for a move operation.

2.2.3.109   MoveItems

The **MoveItems** element is a required element in **MoveItems** command requests and responses that identifies the body of the HTTP **POST** as containing a **MoveItems** command (section 2.2.2.10). It is the top-level element in the XML stream.

All elements referenced in this section are defined in the **Move** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
MoveItems command request	None	**Move** (section 2.2.3.107.1)	**container** ([MS-ASDTYPE] section 2.2)	1…1 (required)
MoveItems command response	None	**Response** (section 2.2.3.140.4)	**container**	1…1 (required)

2.2.3.110   Name

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

- **Autodiscover** command responses (section 2.2.2.1)

- **Search** command requests (section 2.2.2.14)

The definition of the **Name** element differs according to the context in which it is used. For more details, see section 2.2.3.110.1 and section 2.2.3.110.2.

2.2.3.110.1   Name (Autodiscover)

The **Name** element is an optional child element of the **Server** element in **Autodiscover** command responses that specifies a URL if the **Type** element (section 2.2.3.170.1) value is set to "MobileSync".

All elements referenced in this section are defined in the http://schemas.microsoft.com/exchange/autodiscover/mobilesync/requestschema/2006 namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Autodiscover command response (section 2.2.2.1)	**Server** (section 2.2.3.149)	None	**string** ([MS-ASDTYPE] section 2.6)	0...1 (optional)

If the **Type** element value is "MobileSync", then the **Name** element specifies the URL that conveys the protocol. If the **Type** element value is "CertEnroll", then the **Name** element value is NULL.

2.2.3.110.2 Name (Search)

The **Name** element is a required child element of the **Store** element in **Search** command requests that specifies the store to search.

All elements referenced in this section are defined in the **Search** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command request (section 2.2.2.14)	**Store** (section 2.2.3.163.2)	None	**string** ([MS-ASDTYPE] section 2.6)	1...1 (required)

The **Name** element value can be up to 256 characters in length. The **Name** element accepts the following values:

- Mailbox—The client specifies "Mailbox" when it intends to search the mailbox.

- Document Library—The client specifies "DocumentLibrary" when it intends to search a Windows SharePoint Services or UNC library.

- GAL—The client specifies "GAL" when it intends to search the Global Address List.

2.2.3.111 Office

The **Office** element in the **GAL** namespace is an optional child element of the **search:Properties** element in **Search** command responses. The **Office** element is only present in response to **Search** command requests that contain a **search:Name** element with a value of "GAL", as specified in section 2.2.3.110.2. It contains the office location or number of a recipient in the GAL that matched the search criteria from the corresponding **Search** command request.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command response (section 2.2.2.14)	**Properties** (section 2.2.3.128.2)	None	**string** ([MS-ASDTYPE] section 2.6)	0...1 (optional)

2.2.3.112 Oof

The **Oof** element is an optional child element of the **Setttings** element in **Settings** command requests and responses that specifies a named property node for retrieving and setting Out of Office (OOF) information.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command request (section 2.2.2.16)	**Settings** (section 2.2.3.153.2)	**Get** (section 2.2.3.75) **Set** (section 2.2.3.152)	**container** ([MS-ASDTYPE] section 2.2)	0…1 (optional)
Settings command response	**Settings**	**Get** **Status** (section 2.2.3.162.14)	**container**	0…1 (optional)

The **Settings** command request supports **Get** and **Set** operations for the **Oof** element. The **Oof** element enables a user to do the following:

- Specify whether the user is currently out of office.

- Schedule an out of office message to be sent between a particular start date and end date.

- Specify the message that is to be shown to various audiences when the mobile device user is out of office.

Oof Get Request and Response

The **Get** element within the **Oof** element of a **Settings** command request enables the client to retrieve OOF information from the server. The client specifies the body type (**BodyType** element (section 2.2.3.17)) to be retrieved and the server will return all OOF information and messages.

In a **Settings** command **Oof Get** response, there is one **OofMessage** node (section 2.2.3.113) per audience. If the sender group is allowed, but is disabled and has no reply message (specified by the **ReplyMessage** element (section 2.2.3.136)), an **OofMessage** node is still reported to the client.

If the client does not receive a group, it is presumably because the client does not have permission to enter settings for that group; in such a case, any attempt to set those properties results in an Access Denied status code.

Oof Set Request and Response

The **Set** element within the **Oof** element of a **Settings** command request enables the client to set the OOF status, time OOF, and OOF messages for one or more of the following groups:

- Internal

- External Known Senders (such as contacts)

- External Unknown Senders

2.2.3.113   OofMessage

The **OofMessage** element is an optional child element of the **Set** element in **Settings** command requests and an optional child element of the **Get** element in **Settings** command responses. It specifies the OOF message for a particular audience.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command **Oof** request (section 2.2.2.16)	**Set** (section 2.2.3.152)	**AppliesToInternal** (section 2.2.3.14) **AppliesToExternalKnown** (section 2.2.3.12) **AppliesToExternalUnknown** (section 2.2.3.13) **Enabled** (section 2.2.3.56) **ReplyMessage** (section 2.2.3.136) **BodyType** (section 2.2.3.17)	**container** ([MS-ASDTYPE] section 2.2)	0...3 (optional)
Settings command **Oof** response	**Get** (section 2.2.3.75)	**AppliesToInternal** **AppliesToExternalKnown** **AppliesToExternalUnknown** **Enabled** **ReplyMessage** **BodyType**	**container**	0...3 (optional)

The **Oof** property (section 2.2.3.112) supports the following three audiences for an OOF message:<54>

▪ Internal—A user who is in the same organization as the sending user.

▪ Known external—A user who is outside the sending user's organization, but is represented in the sending user's contacts.

▪ Unknown external—A user who is outside the sending user's organization and is not represented in the sending user's contacts.

The presence of one of the following elements, which are mutually exclusive, indicates the audience to which an **OOF** message pertains:

▪ **AppliesToInternal**—The OOF message is relevant to an internal audience.

▪ **AppliesToExternalKnown**—The OOF message is relevant to a known external audience.

▪ **AppliesToExternalUnknown**—The OOF message is relevant to an unknown external audience.

There is one **OofMessage** node per audience in a **Settings** command **Oof Get** response. If a sender group is allowed, but is disabled and has no reply message (specified by the **ReplyMessage** element), an **OofMessage** node is reported to the client. If **AppliesToExternalKnown** or **AppliesToExternalUnknown** are not allowed and are disabled by the administrator but are sent by the client in the **Settings** command **Oof Set** request, the response returns a successful **Status** element (section 2.2.3.162.14) value of 1 even though the user does not have access to these settings. Similarly, the **AppliesToExternalKnown** element and the **AppliesToExternalUnknown** element are returned to the client the response even if the sender group is not allowed and is disabled.

In a **Settings** command **Oof Set** request, the client MUST NOT include the same **AppliesTo*** element in more than one **OofMessage** element.

2.2.3.114 OofState

The **OofState** element is an optional child element of the **Set** element in **Settings** command requests and an optional child element of the **Get** element in **Settings** command responses. It specifies the availability of the **Oof** property (section 2.2.3.112).

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command **Oof** request (section 2.2.2.16)	**Set** (section 2.2.3.152)	None	**integer** ([MS-ASDTYPE] section 2.5)	0…1 (optional)
Settings command **Oof** response	**Get** (section 2.2.3.75)	None	**integer**	0…1 (optional)

The following table lists the valid values for **OofState**.

Value	Meaning
0	The **Oof** property is disabled.
1	The **Oof** property is global.
2	The **Oof** property is time-based.

The **OofState** element value MUST be set to 2 if the **StartTime** (section 2.2.3.161.1) and **EndTime** (section 2.2.3.58.2) elements are present. If the **OofState** element value is not set to 2 and the **StartTime** and **EndTime** elements are submitted in the request, the client does receive a successful response message, but the server does not store the **StartTime** and **EndTime** element values.

2.2.3.115 Options

The **Options** element is used in the following command requests:

- **GetItemEstimate** command requests (section 2.2.2.7)

- **ItemOperations** command requests (section 2.2.2.8)

- **ResolveRecipients** command requests (section 2.2.2.13)

- **Search** command requests (section 2.2.2.14)

- **Sync** command requests (section 2.2.2.19)

The definition of the **Options** element differs according to the context in which it is used. For more details, see section 2.2.3.115.1, section 2.2.3.115.2, section 2.2.3.115.3, section 2.2.3.115.4, and section 2.2.3.115.5.

2.2.3.115.1 Options (GetItemEstimate)

The **airsync:Options** element<55> is an optional child element of the **Collection** element in **GetItemEstimate** command requests that contains elements that filter the results of the **GetItemEstimate** command.

Elements referenced in this section are defined in the **GetItemEstimate** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
GetItemEstimate command request (section 2.2.2.7)	**Collection** (section 2.2.3.29.1)	**airsync:Class** (section 2.2.3.27.1) **airsync:FilterType** (section 2.2.3.64.1) **airsync:MaxItems** (section 2.2.3.93.1)	**container** ([MS-ASDTYPE] section 2.2)	0...2 (optional)

This element is optional; however, when it is present, it MUST include at least one child element. The **airsync:Options** element appears only in requests to the server from the client. If the **airsync:Options** element is not included in a request, then the **GetItemEstimate** command (section 2.2.2.7) will enumerate all of the items within the collection, without any filter (up to a maximum of 512 items).

2.2.3.115.2   Options (ItemOperations)

The **Options** element is an optional child element of the **EmptyFolderContents** element, the **Fetch** element, and the **Move** element in **ItemOperations** command requests. It contains the options for the **EmptyFolderContents** operation, the fetch operation, or the move operation.

Elements referenced in this section are defined in the **ItemOperations** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
ItemOperations command request (section 2.2.2.8) **EmptyFolderContents** operation	**EmptyFolderContents** (section 2.2.3.55)	**DeleteSubFolders** (section 2.2.3.44)	**container** ([MS-ASDTYPE] section 2.2)	0...1 (optional)
ItemOperations command request fetch operation	**Fetch** (section 2.2.3.63.1)	**Schema** (section 2.2.3.145) **Range** (section 2.2.3.130.1) **UserName** (section 2.2.3.177.1) **Password** (section 2.2.3.122.1) **airsync:MIMESupport** (section 2.2.3.100.1) **airsyncbase:BodyPreference** ([MS-ASAIRS] section 2.2.2.7) **airsyncbase:BodyPartPreference** ([MS-ASAIRS] section 2.2.2.6) **rm:RightsManagementSupport<56>** ([MS-ASRM] section 2.2.2.15)	**container**	0...1 (optional)
ItemOperations command request **Move** operation	**Move** (section 2.2.3.107.1)	**MoveAlways** (section 2.2.3.108)	**container**	0...1 (optional)

If the client specifies an option that is invalid for the parent element, the server returns a **Status** element (section 2.2.3.162.7) value of 2.

The following options are supported for the fetch operation:

- Schema

    - Per-class settings on format for fetch results.

    - This protocol supports schemas for **Personal Information Manager (PIM)** items only; it does not support schemas for document library items or attachments.

    - Supports all top-level property nodes.

    - For more information about the **Schema** element, section 2.2.3.145.

- Byte ranges

    - Facilitates a checkpoint to improve the reliability of large data downloads.

    - This protocol supports ranges for document library items and attachments; it does not support ranges for other item types—that is, PIM items, such as email, contact, calendar, or task items.

    - For attachments, the range applies to the file content.

    - For document library items, this applies to the file content.

    - For more information about the **Range** element, section 2.2.3.130.1.

- User name and password

    - Identifies the username and password required to fetch the desired item.

    - For more information about **UserName** and **Password** elements, see section 2.2.3.177.1 and section 2.2.3.122.1 respectively.

- MIME support

    - Indicates whether the server returns MIME content for S/MIME-only messages, all messages, or no messages

    - For more information about the **airsync:MIMESupport** element, section 2.2.3.100.1.

- Body format preference

    - Per-class settings on preferred body format.

    - This protocol supports body preferences for PIM items only; it does not support body format preferences for document library items or attachments.

    - For more information about the **airsyncbase:BodyPreference** element, see [MS-ASAIRS] section 2.2.2.7.

- Body part preference

    - Per email settings on preferred body part format.

- When the **airsyncbase:BodyPartPreference** element is specified, the server returns the unique parts of the email message body (2) with reference to the parent item across all collections.

- For more information about the **airsyncbase:BodyPartPreference** element, see [MS-ASAIRS] section 2.2.2.6.

- Rights management support

  - Indicates whether the device has information rights management enabled. If information rights management is enabled, the client requests that the information rights management content be returned in its schematized form in the response.

If the **airsyncbase:FileReference** element (section 2.2.3.8) is present, then the **Range** element is the only valid child element of the **Options** element.

2.2.3.115.3   Options (ResolveRecipients)

The **Options** element is an optional child element of the **ResolveRecipients** element in **ResolveRecipients** command requests that contains the options for resolving the list of recipients.

All elements referenced in this section are defined in the **ResolveRecipients** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ResolveRecipients command request (section 2.2.2.13)	**ResolveRecipients** (section 2.2.3.139)	**CertificateRetrieval** (section 2.2.3.22) **MaxCertificates** (section 2.2.3.91) **MaxAmbiguousRecipients** (section 2.2.3.90) **Availability** (section 2.2.3.16) **Picture** (section 2.2.3.125.1)<57>	container ([MS-ASDTYPE] section 2.2)	0...1 (optional)

2.2.3.115.4   Options (Search)

The **Options** element is an optional child element of the **Store** element in **Search** command requests that contains the search options.

Elements referenced in this section are defined in the **Search** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command request (section 2.2.2.14)	**Store** (section 2.2.3.163.2)	**airsync:MIMESupport** (section 2.2.3.100.2) **airsyncbase:BodyPreference** ([MS-ASAIRS] section 2.2.2.7) **airsyncbase:BodyPartPreference** ([MS-ASAIRS] section 2.2.2.6) **rm:RightsManagementSupport**<58>	container ([MS-ASDTYPE] section 2.2)	0...1 (optional)

Command request/response	Parent elements	Child elements	Data type	Number allowed
		([MS-ASRM] section 2.2.2.15) **Range** (section 2.2.3.130.2) **UserName** (section 2.2.3.177.2) **Password** (section 2.2.3.122.2) **DeepTraversal** (section 2.2.3.41) **RebuildResults** (section 2.2.3.131) **Picture** (section 2.2.3.125.2)		

The **UserName** element and **Password** element can only be sent in the request after receiving a **Status** element (section 2.2.3.162.12) value of 14 (see section 2.2.3.162.12 for more details). The server requires these credentials to access the requested resources. The client MUST only send these over a secure or trusted connection, and only in response to a **Status** value of 14. The **UserName** and **Password** element values are defined as strings consisting of at most 100 characters.

The supported options vary according to the **message store** that is being searched. The following table lists the valid options for each store.

Options	Store
Range **UserName** **Password** **Picture**	GAL
Range **DeepTraversal** **RebuildResults** **airsyncbase:BodyPreference** **airsyncbase:BodyPartPreference** **rm:RightsManagementSupport**<59>	Mailbox
Range **UserName** **Password**	Document Library

The **airsyncbase:BodyPartPreference** element is only valid in **Search** command requests that include a **ConversationId** element, as specified in section 2.2.3.35.2.

2.2.3.115.5 Options (Sync)

The **Options** element is an optional child element of the **Collection** element in **Sync** command requests that contains elements that control certain aspects of how the synchronization is performed.

Elements referenced in this section are defined in the **AirSync** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
Sync command request (section 2.2.2.19)	**Collection** (section 2.2.3.29.2)	**FilterType** (section 2.2.3.64.2) **Conflict** (section 2.2.3.34) **MIMETruncation** (section 2.2.3.101) **MIMESupport** (section 2.2.3.100.3) **Class** (section 2.2.3.27.5) **MaxItems** (section 2.2.3.93.2) **airsyncbase:BodyPreference** ([MS-ASAIRS] section 2.2.2.7) **airsyncbase:BodyPartPreference** ([MS-ASAIRS] section 2.2.2.6) **rm:RightsManagementSupport**<60> ([MS-ASRM] section 2.2.2.15)	**container** ([MS-ASDTYPE] section 2.2)	0...2 (optional)

This element is optional, but if it is present, it SHOULD include at least one child element. The **Options** element appears only in requests to the server from the client.

Synchronization options enable the client to specify truncation and content settings. These settings are encapsulated within a **airsyncbase:BodyPreference** child element, as specified in [MS-ASAIRS] section 2.2.2.7.

Because synchronization options are specified on a collection, the client can specify a unique **airsyncbase:BodyPreference** element value for each collection that it is being synchronized. For more details about the **airsyncbase:BodyPreference** element, see [MS-ASAIRS] section 2.2.2.7.

The server preserves the **Options** block across requests, using a concept referred to as "sticky options". If the **Options** block is not included in a request, the previous **Options** block is used. Whenever the client specifies new options by including an **Options** block in the request, the server MUST replace the original **Options** block with the new **Options** block.

If two **Options** elements are included in a single **Sync** command request, one of the **Options** elements MUST specify the synchronization options for the **SMS** class, while the other **Options** element specifies the options for the default class of the given folder.

2.2.3.116 Or

The **Or** element is an optional child element of the **Query** element in **Search** command requests. It contains elements that specify items on which to perform an **OR** operation.

If the **Or** element is included in a **Search** command request, the server responds with a **Status** element (section 2.2.3.162.12) value of 8 (SearchTooComplex).

2.2.3.117 OS

The **OS** element is an optional child element of the **Set** element in **Settings** command requests that specifies the operating system of the client device.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command **DeviceInformation** request (section 2.2.2.16)	**Set** (section 2.2.3.152)	None	**string** ([MS-ASDTYPE] section 2.6)	0...1 (optional)

Some information about the operating system of the device can be collected from the user agent string that is associated with requests from that client. The mapping from user agent to operating system is not one to one, however, and therefore does not provide sufficient information to troubleshoot and establish an inventory.

The **OS** element is a **string** value (up to 1024 characters in length) that enables the client to precisely specify the operating system of the device. The server does not perform any validation of this value, but clients SHOULD<61> use the following convention:

<Operating System Product Name> <Operating System Major Version> <Operating System Minor Version>

2.2.3.118 OSLanguage

The **OSLanguage** element is an optional child element of the **Set** element in **Settings** command requests that specifies the language that is used by the operating system of the client device.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command **DeviceInformation** request (section 2.2.2.16)	**Set** (section 2.2.3.152)	None	**string** ([MS-ASDTYPE] section 2.6)	0...1 (optional)

Knowledge of the user's language facilitates localization if the server is required to send localizable content to the client device. The **OS** element value can be up to 1024 characters in length. The server does not validate the value of the **OSLanguage** element.

2.2.3.119 ParentId

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

- **FolderCreate** command requests (section 2.2.2.2)

- **FolderSync** command responses (section 2.2.2.4)

- **FolderUpdate** command requests (section 2.2.2.5)

The definition of the **ParentId** element differs according to the context in which it is used. For more details, see section 2.2.3.119.1, section 2.2.3.119.2, and section 2.2.3.119.3.

2.2.3.119.1 ParentId (FolderCreate)

The **ParentId** element is a required child element of the **FolderCreate** element in **FolderCreate** command requests that specifies the server ID of the parent folder. The server ID of the parent folder is obtained from the **ServerId** element (section 2.2.3.151.3) of a previous **FolderSync**

command (section 2.2.2.4). A parent ID of 0 (zero) signifies the mailbox Root folder. The **ParentId** element value can be up to 64 characters in length.

All elements referenced in this section are defined in the **FolderHierarchy** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
FolderCreate command request (section 2.2.2.2)	**FolderCreate** (section 2.2.3.67)	None	**string** ([MS-ASDTYPE] section 2.6)	1…1 (required)

2.2.3.119.2 ParentId (FolderSync)

The **ParentId** element is a required child element of the **Update** element and the **Add** element in **FolderSync** command responses that specifies the server ID of the parent folder of the folder on the server that has been updated or added.

All elements referenced in this section are defined in the **FolderHierarchy** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
FolderSync command response (section 2.2.2.4)	**Update** (section 2.2.3.171)	None	**string** ([MS-ASDTYPE] section 2.6)	1…1 (required)
FolderSync command response	**Add** (section 2.2.3.7.1)	None	**string**	1…1 (required)

The client MUST store the parent ID as an opaque string of up to 64 characters.

Each **Update** element and each **Add** element included in a **FolderSync** response MUST contain one **ParentId** element.

2.2.3.119.3 ParentId (FolderUpdate)

The **ParentId** element is a required child element of the **FolderUpdate** element in **FolderUpdate** command requests that specifies the server ID of the parent folder of the folder to be renamed or the destination folder of the folder to be moved.

All elements referenced in this section are defined in the **FolderHierarchy** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
FolderUpdate command request (section 2.2.2.5)	**FolderUpdate** (section 2.2.3.72)	None	**string** ([MS-ASDTYPE] section 2.6)	1…1 (required)

The **ParentId** element value is obtained from the **ServerId** element (section 2.2.3.151.3) of a previous **FolderSync** command response (section 2.2.2.4) or **FolderCreate** command response (section 2.2.2.2). The client MUST store the **ParentId** element value as an opaque string of up to 64 characters.

A **ParentId** element value of 0 (zero) signifies the mailbox Root folder.

2.2.3.120 Part

The **Part** element is an optional child element of the **Properties** element or the **airsyncbase:Body** element in **ItemOperations** command responses that specifies an integer index into the metadata of the multipart response.

All elements referenced in this section are defined in the **ItemOperations** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ItemOperations command response (section 2.2.2.8) fetch operation	**Properties** (section 2.2.3.128.1) **airsyncbase:Body** ([MS-ASAIRS] section 2.2.2.4)	None	**integer** ([MS-ASDTYPE] section 2.5)	0...1 (optional)

The **Part** element is present only in a multipart **ItemOperations** response. It is a child of the **Properties** element if the item being fetched is located in a document library, and is a child of the **airsyncbase:Body** element if the item being fetched is located in a mailbox.

The **Part** element can be used to locate the [start, end] tuple that specifies the starting byte and ending byte for this item's binary content in the command response.

2.2.3.121 Partial

The **Partial** element is an optional child element of the **Sync** element in **Sync** command requests that indicates to the server that the client sent a partial list of collections, in which case the server obtains the rest of the collections from its cache.

All elements referenced in this section are defined in the **AirSync** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Sync command request (section 2.2.2.19)	**Sync** (section 2.2.3.165)	None	None	0...1 (optional)

The **Partial** 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 <Partial/> tag.

The client MUST NOT send a **Partial** element without any other elements in the **Sync** request. A **Sync** command request is valid with only a **Partial** element and either a **Wait** element (section 2.2.3.182) or **HeartbeatInterval** element (section 2.2.3.79.2), a **WindowSize** element (section 2.2.3.183), a **Collections** element (section 2.2.3.31.2), or any combination of the three. A **Sync** command request requires, at least, either a **Partial** element or a **Collections** element.

When a **Sync** command request includes a **Partial** element but does not specify some collections, the settings and synchronization key for each of those unspecified collections specified in the previous **Sync** request remain the same as specified in the previous request. Such a request is equivalent to a request that specifies each of these collections with the same settings and synchronization key as in the previous request. This enables the client to modify some aspect of the previous request (one of the collections, the wait time, the global window size, and so on) without sending up every unchanged collection.

2.2.3.122 Password

The **Password** element is used in the following command requests:

- **ItemOperations** command requests (section 2.2.2.8)

- **Search** command requests (section 2.2.2.14)

- **Settings** command requests (section 2.2.2.16)

The definition of the **Password** element differs according to the context in which it is used. For more details, see section 2.2.3.122.1, section 2.2.3.122.2, and section 2.2.3.122.3.

2.2.3.122.1 Password (ItemOperations)

The **Password** element is an optional child element of the **Options** element in **ItemOperations** command requests that specifies the password for the given **UserName** (section 2.2.3.177.1).

All elements referenced in this section are defined in the **ItemOperations** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ItemOperations command request (section 2.2.2.8) fetch operation	**Options** (section 2.2.3.115.2)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

The server accepts password values up to 256 characters in length. However, logon dialog boxes can limit password lengths to a smaller value.

2.2.3.122.2 Password (Search)

The **Password** element is an optional child element of the **Options** element in **Search** command requests that specifies the password for the given **UserName** (section 2.2.3.177.2). The value of the **Password** element has a maximum length of 100 characters.

All elements referenced in this section are defined in the **Search** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command request (section 2.2.2.14)	**Options** (section 2.2.3.115.2)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

2.2.3.122.3 Password (Settings)

The **Password** element is a required child element of the **Set** element in **Settings** command **DevicePassword Set** operation requests that specifies the recovery password of the client device, which is stored by the server.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command **DevicePassword** request (section 2.2.2.16)	**Set** (section 2.2.3.152)	None	**string** ([MS-ASDTYPE] section 2.6)	1…1 (required)

The value of the **Password** element has a maximum length of 255 characters.

To clear an existing recovery password, the client MUST send a **Settings** command **DevicePassword Set** request with an empty **Password** element.

2.2.3.123 Phone

The **Phone** element in the **GAL** namespace is an optional child element of the **search:Properties** element in **Search** command responses. The **Phone** element is only present in response to **Search** command requests that contain a **search:Name** element with a value of "GAL", as specified in section 2.2.3.110.2. It contains the phone number of a recipient in the GAL that matched the search criteria from the corresponding **Search** command request.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command response (section 2.2.2.14)	**Properties** (section 2.2.3.128.2)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

2.2.3.124 PhoneNumber

The **PhoneNumber** element is an optional child element of the **Set** element in **Settings** command requests that specifies a unique number that identifies the client device.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command **DeviceInformation** request (section 2.2.2.16)	**Set** (section 2.2.3.152)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

The telephone number facilitates troubleshooting and device management by providing a well-known and unique identifier for the client device. The **PhoneNumber** element value can be up to 1024 characters in length. The server does not validate the value of the **PhoneNumber** element.<62>

2.2.3.125 Picture

The **Picture** element<63> is used in the following command requests and responses:

▪ **ResolveRecipients** command requests and responses (section 2.2.2.13)

▪ **Search** command requests and responses (section 2.2.2.14)

The definition of the **Picture** element differs according to the context in which it is used. For more details, see section 2.2.3.125.1 and section 2.2.3.125.2.

2.2.3.125.1 Picture (ResolveRecipients)

The **Picture** element<64> is an optional child element of the **Options** element in **ResolveRecipients** command requests that indicates that the client is requesting that contact photos be returned in the server response. The **Picture** element is an optional child element of the **Recipient** element in **ResolveRecipients** command responses that contains the data related to the contact photos.

All elements referenced in this section are defined in the **ResolveRecipients** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ResolveRecipients command request (section 2.2.2.13)	**Options** (section 2.2.3.115.3)	**MaxSize** (section 2.2.3.95.1) **MaxPictures** (section 2.2.3.94.1)	**container** ([MS-ASDTYPE] section 2.2) or None	0…1 (optional)
ResolveRecipients command response	**Recipient** (section 2.2.3.132)	**Status** (section 2.2.3.162.11) **Data** (section 2.2.3.39.2)	**container**	0…1 (optional)

2.2.3.125.2 Picture (Search)

The **search:Picture** element<65> is an optional child element of the **Options** element in **Search** command requests that indicates that the client is requesting that contact photos be returned in the server response. The **gal:Picture** element<66> is an optional child element of the **Properties** element in **Search** command responses that contains the data related to the contact photos.

Elements referenced in this section are defined in the **Search** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command request (section 2.2.2.14)	**Options** (section 2.2.3.115.4)	**MaxSize** (section 2.2.3.95.2) **MaxPictures** (section 2.2.3.94.2)	**container** ([MS-ASDTYPE] section 2.2) or None	0…1 (optional)
Search command response	**Properties** (section 2.2.3.128.2)	**Status** (section 2.2.3.162.12) **gal:Data** (section 2.2.3.39.3)	**container**	0…1 (optional)

2.2.3.126 Ping

The **Ping** element is a required element in **Ping** command requests and responses that identifies the body of the HTTP **POST** as containing a **Ping** command (section 2.2.2.11). It is the top-level element in the XML stream.

All elements referenced in this section are defined in the **Ping** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Ping command request	None	**HeartbeatInterval** (section 2.2.3.79.1) **Folders** (section 2.2.3.70)	**container** ([MS-ASDTYPE] section 2.2)	1…1 (required)
Ping command response	None	**HeartbeatInterval** **Folders** **MaxFolders** (section 2.2.3.92) **Status** (section 2.2.3.162.10)	**container**	1…1 (required)

2.2.3.127 PrimarySmtpAddress

The **PrimarySmtpAddress** element<67> is an optional child element of the **EmailAddresses** element in **Settings** command responses that specifies the primary SMTP address for the given account.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command **UserInformation** response (section 2.2.2.16)	**EmailAddresses** (section 2.2.3.54)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

The value of the **PrimarySmtpAddress** element can also be returned as a value for the **SMTPAddress** element (section 2.2.3.156).

2.2.3.128 Properties

The **Properties** element is used in the following command responses:

- **ItemOperations** command responses (section 2.2.2.8)

- **Search** command responses (section 2.2.2.14)

The definition of the **Properties** element differs according to the context in which it is used. For more details, see section 2.2.3.128.1 and section 2.2.3.128.2.

2.2.3.128.1 Properties (ItemOperations)

The **Properties** element is an optional child element of the **Fetch** element in **ItemOperations** command responses that contains the properties that are returned for item(s) in the response.

Elements referenced in this section are defined in the **ItemOperations** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
ItemOperations command response (section 2.2.2.8) fetch operation	**Fetch** (section 2.2.3.63.1)	**Range** (section 2.2.3.130.1) **Data** (section 2.2.3.39.1) **Part** (section 2.2.3.120) **Version** (section 2.2.3.181) **Total** (section 2.2.3.169.1) **airsyncbase:Body** ([MS-ASAIRS] section 2.2.2.4) **airsyncbase:BodyPart** ([MS-ASAIRS] section 2.2.2.5) Data elements are from the content classes. For details about the content classes, see [MS-ASCAL], [MS-ASCNTC], [MS-ASDOC], [MS-ASEMAIL], and [MS-ASTASK]. **rm:RightsManagementLicense** ([MS-ASRM] section 2.2.2.14)	**container** ([MS-ASDTYPE] section 2.2)	0...1 (optional)

2.2.3.128.2   Properties (Search)

The **Properties** element is a required child element of the **Result** element in **Search** command responses that contains the properties that are returned for item(s) in the response.

Elements referenced in this section are defined in the **Search** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command response (section 2.2.2.14)	**Result** (section 2.2.3.142.2)	**airsyncbase:Body** ([MS-ASAIRS] section 2.2.2.4) **airsyncbase:BodyPart** ([MS-ASAIRS] section 2.2.2.5) **gal:Picture** (section 2.2.3.125.2) Data elements from the content classes. For more details about the content classes, see [MS-ASCAL], [MS-ASCNTC], [MS-ASDOC], [MS-ASEMAIL], [MS-ASNOTE], and [MS-ASTASK]. **rm:RightsManagementLicense** ([MS-ASRM] section 2.2.2.14)	**Container** ([MS-ASDTYPE] section 2.2)	1...1 (required)

The **Search** command response **Properties** element is a container for properties that apply to an individual entry that matches the **Query** element (section 2.2.3.129) search string. For example, the **Properties** element contains an element for each nonempty, text-valued GAL property that is attached to the matching GAL entry. Only those properties that are attached to the specific GAL entry are returned; therefore different sets of properties can be returned in the response XML for different matching GAL entries.

Each element in the **Properties** container is scoped to the appropriate namespace that is specified in the top-level **Search** element (section 2.2.3.146).

2.2.3.129 Query

The **Query** element is a required child element of the **Store** element in **Search** command requests that specifies the keywords to use for matching the entries in the store that is being searched.

All elements referenced in this section are defined in the **Search** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command request (section 2.2.2.14)	**Store** (section 2.2.3.163.2)	**And** (section 2.2.3.10) **EqualTo** (section 2.2.3.59)	**container** ([MS-ASDTYPE] section 2.2)	1...1 (required)

The value of the **Query** element is used as a prefix-string matching pattern, and returns entries that match the beginning of the string. For example, searching for "John" would match "John Frum" or "Barry Johnson", but would not match "James Littlejohn".

All nonempty text properties in the GAL that are indexed by using ANR are compared with the **Query** element value. **Search** comparisons are performed by using case-insensitive matching.

For a Windows SharePoint Services document library search, this protocol supports queries of the following form: LinkId == value, where value specifies the URL of the item or folder and LinkId indicates that the value is to be compared to the link ID property.

For mailbox search, the query syntax is as follows:

- Folders can be specified in the following ways:

  - Specified ID

  - Specified folder and subfolders

  - All email folders, including Draft, Inbox and subfolders, Outbox, and Sent Items

- The basic keyword query can be composed of the following:

  - The basic operator: **And** (section 2.2.3.10)

  - A **dateTime** filter specified by using the **GreaterThan** (section 2.2.3.78) and **LessThan** elements (section 2.2.3.87)

  - **FreeText** elements (section 2.2.3.73) that contain keywords

The basic keyword query is executed against all indexed properties.

2.2.3.130 Range

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

- **ItemOperations** command requests and responses (section 2.2.2.8)

- **Search** command requests and responses (section 2.2.2.14)

The definition of the **Range** element differs according to the context in which it is used. For more details, see section 2.2.3.130.1 and section 2.2.3.130.2.

2.2.3.130.1   Range (ItemOperations)

The **Range** element is an optional child element of the **Options** element in **ItemOperations** command requests that specifies the range of bytes that the client can receive in response to the fetch operation for a document library item. The **Range** element is an optional child element of the **Properties** element in **ItemOperations** command responses that specifies the actual range of bytes for an item that is contained in a given fetch operation.

Elements referenced in this section are defined in the **ItemOperations** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
ItemOperations command request (section 2.2.2.8) fetch operation	**Options** (section 2.2.3.115.2)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)
ItemOperations command response fetch operation	**Properties** (section 2.2.3.128.1)	None	**string**	0…1 (optional)

The **Range** element contains a **string** value in the format m-n, where m<n, and m is the minimum value and n is the maximum value. The byte range is zero-indexed; the first byte is indicated by a 0 (zero).

The server provides a best effort at fulfilling the request. Therefore, the client cannot assume that the byte-range that is specified in the request exactly matches the byte-range that is returned in the response. The byte-range that is specified by the server in the response is the authoritative value.

If the **Range** element is omitted in the request, the whole item is fetched.

If the **airsyncbase:FileReference** element (section 2.2.3.8) is present in the request, then the **Range** element is the only valid child element of the **Options** element.

2.2.3.130.2   Range (Search)

The **Range** element is an optional child element of the **Options** element in **Search** command requests that specifies the maximum number of matching entries to return. The **Range** element is an optional child element of the **Store** element in **Search** command responses that specifies the number of matching entries that are being returned.

All elements referenced in this section are defined in the **Search** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command request (section 2.2.2.14)	**Options** (section 2.2.3.115.4)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)
Search command response	**Store** (section 2.2.3.163.2)	None	**string**	0…1 (optional)

The result of including more than one **Range** element in a **Search** command request is undefined. The server MAY return a protocol status error in response to such a command request.

The **Range** element value specifies a number of entries, but indicates different things depending on whether the element is in the request or the response XML.

The format of the **Range** element value is in the form of a zero-based index specifier, formed with a zero, a hyphen, and another numeric value: "m-n." The m indicates the lowest index of a zero-based array that would hold the items. The n indicates the highest index of a zero-based array that would hold the items. For example, a **Range** element value of 0–9 indicates 10 items, and 0–10 indicates 11 items. A **Range** element value of 0–0 indicates 1 item.

If the request does not include a **Range** element, the default Range value for each **Store** type is used. The following table identifies the default Range values and maximum results returned for each **Store** type:

Store value	Default range value	Maximum results returned
Mailbox	0-99	100
DocumentLibrary	0-999	1000
GAL	0-99	100

If the **Range** element value specified in the request exceeds the default range value, a **Status** element (section 2.2.3.162.12) value of 12 is returned to indicate that the maximum range has been exceeded, as specified in section 2.2.3.162.12.

In the **Search** command response, the **Total** element (section 2.2.3.169.2) indicates an estimate of the total number of entries that matched the **Query** element (section 2.2.3.129) value.

Search results are stored in a **search folder (2)** on the server. This way, when a client comes back with the same query but a new row range, rows are pulled from the result set that is currently stored in the search folder. The entire result set does not have to be rebuilt.

2.2.3.131   RebuildResults

The **RebuildResults** element is an optional child element of the **Options** element in **Search** command requests that forces the server to rebuild the search folder (2) that corresponds to a given query.

All elements referenced in this section are defined in the **Search** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command request (section 2.2.2.14)	**Options** (section 2.2.3.115.4)	None	None	0...1 (optional)

The **RebuildResults** 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 <RebuildResults/> tag.

The result of including more than one **RebuildResults** element in a **Search** command request is undefined. The server MAY return a protocol status error in response to such a command request.

The search results (that is, the result set) are stored in a search folder on the server. This way, when a client comes back with the same query but a new row range, rows are pulled from the result set that is currently stored in the search folder. The entire result set does not have to be rebuilt.

The search folder remains unchanged until the client does one of the following to update the result set:

▪ Sends a **Search** request, specifying a new query. In this case, the search folder is automatically rebuilt. The **RebuildResults** node does not have to be included.

▪ Sends a **Search** request that includes the **RebuildResults** node. In this case, the server is forced to rebuild the search folder.

If a new item is added, the item does not appear in the result set until the result set is updated. If an item is deleted, the server will filter the deleted item out of the result set.

The client SHOULD send a new **Search** request with the given query and include the **RebuildResults** option every few days to ensure accurate results for that query.

2.2.3.132 Recipient

The **Recipient** element is an optional child element of the **Response** element in **ResolveRecipients** command responses that represents a single recipient that has been resolved.

All elements referenced in this section are defined in the **ResolveRecipients** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ResolveRecipients command response (section 2.2.2.13)	**Response** (section 2.2.3.140.5)	**Type** (section 2.2.3.170.4) **DisplayName** (section 2.2.3.47.4) **EmailAddress** (section 2.2.3.53.1) **Availability** (section 2.2.3.16) **Certificates** (section 2.2.3.23.1) **Picture** (section 2.2.3.125.1)	container ([MS-ASDTYPE] section 2.2)	0...N (optional)

One or more **Recipient** elements are returned to the client in a **Response** element by the server if the **To** element specified in the request was either resolved to a distribution list or found to be ambiguous. The status code returned in the **Response** element can be used to determine if the recipient was found to be ambiguous. The recipient would be a suggested match if the recipient specified in the request was found to be ambiguous.

A **Certificates** element is returned as a child element of the **Recipient** element if the client requested certificates to be returned in the response.

2.2.3.133 RecipientCount

The **RecipientCount** element is a required child element of the **Response** element and the **Certificates** element in **ResolveRecipients** command responses (section 2.2.2.13).

All elements referenced in this section are defined in the **ResolveRecipients** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ResolveRecipients command response	**Response** (section 2.2.3.140.5) **Certificates** (section 2.2.3.23.1)	None	**integer** ([MS-ASDTYPE] section 2.5)	1…1 (required)

As a child element of the **Response** element, the **RecipientCount** element specifies the number of recipients that are returned in the **ResolveRecipients** command response.

As a child element of the **Certificates** element, the **RecipientCount** element specifies the number of members belonging to a distribution list. It can be used to determine whether all recipients belonging to a distribution list have valid certificates by comparing values of the **CertificateCount** (section 2.2.3.21) and **RecipientCount** elements.

2.2.3.134   Redirect

The **Redirect** element is an optional child element of the **Action** element in **Autodiscover** command responses that specifies the SMTP address of the requested user.

All elements referenced in this section are defined in the http://schemas.microsoft.com/exchange/autodiscover/mobilesync/requestschema/2006 namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Autodiscover command response (section 2.2.2.1)	**Action** (section 2.2.3.6)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

The client device uses the **domain** part of the address to send a new **Autodiscover** command request.

2.2.3.135   ReplaceMime

The **ReplaceMime** element is an optional child element of the **SmartForward** element in **SmartForward** command requests and an optional child element of the **SmartReply** element in **SmartReply** command requests. In both contexts, it specifies whether the client is sending the entire message. When the **ReplaceMime** element is present, the server MUST not include the body or attachments of the original message being forwarded. When not included, the client MUST append the body of the original message as attachments to the outgoing message.

The client can use this tag to indicate whether the message was edited inline, or whether the message had reply/forward text prepended to the source message. If the **ReplaceMime** element is present, the message was edited.

All elements referenced in this section are defined in the **ComposeMail** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
SmartForward command request (section 2.2.2.17)	**SmartForward** (section 2.2.3.154)	None	None	0…1 (optional)

Command request/response	Parent elements	Child elements	Data type	Number allowed
SmartReply command request (section 2.2.2.18)	**SmartReply** (section 2.2.3.155)	None	None	0…1 (optional)

The **ReplaceMime** 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 <ReplaceMime/> tag.

2.2.3.136   ReplyMessage

The **ReplyMessage** element is an optional child element of the **OofMessage** element in **Settings** command requests and responses that specifies the message to be shown to a particular audience when the user is OOF.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command **Oof** request and response (section 2.2.2.16)	**OofMessage** (section 2.2.3.113)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

The **ReplyMessage** element can be used in a **Settings** command **Oof Get** (section 2.2.3.75) response to convey the requested OOF message, or in a **Settings** command **Oof Set** request to set the message that the client wants to send to a particular audience. If a **ReplyMessage** element is included in an **Oof Set** request, a **BodyType** element (section 2.2.3.17) MUST also be included.

The **Oof** property (section 2.2.3.112) supports the following three audiences for an OOF message:

- Internal—A user who is in the same organization as the sending user.

- Known external—A user who is outside the sending user's organization, but is represented in the sending user's contacts.

- Unknown external—A user who is outside the sending user's organization and is not represented in the sending user's contacts.

The presence of one of the following elements, which are mutually exclusive, indicates the audience to which an OOF message pertains:

- **AppliesToInternal**—The OOF message is relevant to an internal audience.

- **AppliesToExternalKnown**—The OOF message is relevant to a known external audience.

- **AppliesToExternalUnknown**—The OOF message is relevant to an unknown external audience.

2.2.3.137   Request

The **Request** element is used in the following command requests:

- **Autodiscover** command requests (section 2.2.2.1)

- **MeetingResponse** command requests (section 2.2.2.9)

The definition of the **Request** element differs according to the context in which it is used. For more details, see section 2.2.3.137.1 and section 2.2.3.137.2.

2.2.3.137.1 Request (Autodiscover)

The **Request** element is a required child element of the **Autodiscover** element in **Autodiscover** command requests that contains the **Autodiscover** command request parameters.

All elements referenced in this section are defined in the http://schemas.microsoft.com/exchange/autodiscover/mobilesync/requestschema/2006 namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Autodiscover command request (section 2.2.2.1)	**Autodiscover** (section 2.2.3.15)	**EMailAddress** (section 2.2.3.52) **AcceptableResponseSchema** (section 2.2.3.1)	container ([MS-ASDTYPE] section 2.2)	1…1 (required)

When more than one **Request** elements are present in an **Autodiscover** command request, the server returns an **ErrorCode** (section 2.2.3.61) value of 600.

2.2.3.137.2 Request (MeetingResponse)

The **Request** element is a required child element of the **MeetingResponse** element in **MeetingResponse** command requests. Its child elements specify the meeting request that is being responded to, the response to that meeting request, and the folder on the server that the meeting request is located in.

Elements referenced in this section are defined in the **MeetingResponse** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
MeetingResponse command request (section 2.2.2.9)	**MeetingResponse** (section 2.2.3.96)	**UserResponse** (section 2.2.3.178) **CollectionId** (section 2.2.3.30.3) **RequestId** (section 2.2.3.138) **search:LongId** (section 2.2.3.89.2) **InstanceId** (section 2.2.3.83.1)	container ([MS-ASDTYPE] section 2.2)	1…N (required)

2.2.3.138 RequestId

The **RequestId** element is a child element of the **Request** element in **MeetingResponse** command requests and a child element of the **Result** element in **MeetingResponse** command responses. In both contexts, it specifies the server ID of the meeting request message item.

Elements referenced in this section are defined in the **MeetingResponse** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
MeetingResponse command request (section 2.2.2.9)	**Request** (section 2.2.3.137.2)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (Required, or optional if **search:LongId** (section 2.2.3.89.2) is included)
MeetingResponse command response	**Result** (section 2.2.3.142.1)	None	**string**	0…1 (optional)

When the client sends a **MeetingResponse** command request, the client includes a **RequestId** element to identify which meeting request is being responded to. The **RequestId** element is also returned in the response to the client along with the status of the user's response to the meeting request. The **RequestId** element value can be up to 64 characters in length.

The **RequestId** element is present in **MeetingResponse** command responses only if it was present in the corresponding **MeetingResponse** command request.

2.2.3.139 ResolveRecipients

The **ResolveRecipients** element is a required element in **ResolveRecipients** command requests and responses that identifies the body of the HTTP **POST** as containing a **ResolveRecipients** command (section 2.2.2.13). It is the top-level element in the XML stream.

All elements referenced in this section are defined in the **ResolveRecipients** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ResolveRecipients command request	None	**To** (section 2.2.3.168) **Options** (section 2.2.3.115.3)	**container** ([MS-ASDTYPE] section 2.2)	1…1 (required)
ResolveRecipients command response	None	**Status** (section 2.2.3.162.11) **Response** (section 2.2.3.140.5)	**container**	1…1 (required)

2.2.3.140 Response

The **Response** element is used in the following command responses:

- **Autodiscover** command responses (section 2.2.2.1)

- **GetItemEstimate** command responses (section 2.2.2.7)

- **ItemOperations** command responses (section 2.2.2.8)

- **MoveItems** command responses (section 2.2.2.10)

- **ResolveRecipients** command responses (section 2.2.2.13)

▪ **Search** command responses (section 2.2.2.14)

The definition of the **Response** element differs according to the context in which it is used. For more details, see section 2.2.3.140.1, section 2.2.3.140.2, section 2.2.3.140.3, section 2.2.3.140.4, section 2.2.3.140.5, and section 2.2.3.140.6.

2.2.3.140.1   Response (Autodiscover)

The **Response** element is a required child element of the **Autodiscover** element in **Autodiscover** command responses. It contains the **Autodiscover** command response parameters.

All elements referenced in this section are defined in the http://schemas.microsoft.com/exchange/autodiscover/mobilesync/requestschema/2006 namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Autodiscover command response (section 2.2.2.1)	**Autodiscover** (section 2.2.3.15)	**Culture** (section 2.2.3.38) **User** (section 2.2.3.173) **Action** (section 2.2.3.6) **Error** (section 2.2.3.60)	**container** ([MS-ASDTYPE] section 2.2)	1…1 (required)

If an error occurs in the **Autodiscover** command framework that hosts the Autodiscovery implementation, then the **Response** element MUST have an **Error** child element.

2.2.3.140.2   Response (GetItemEstimate)

The **Response** element is a required child element of the **GetItemEstimate** element in **GetItemEstimate** command responses that contains elements that describe estimated changes. Its child elements specify the status of the **GetItemEstimate** command request and information about the collection on which the estimate was made.

All elements referenced in this section are defined in the **GetItemEstimate** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
GetItemEstimate command response (section 2.2.2.7)	**GetItemEstimate** (section 2.2.3.77)	**Status** (section 2.2.3.162.6) **Collection** (section 2.2.3.29.1)	**container** ([MS-ASDTYPE] section 2.2)	1…N (required)

2.2.3.140.3   Response (ItemOperations)

The **Response** element is an optional child element of the **ItemOperations** element in **ItemOperations** command responses. It contains the operation responses.

All elements referenced in this section are defined in the **ItemOperations** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ItemOperations command response (section 2.2.2.8)	**ItemOperations**	**EmptyFolderContents** (section 2.2.3.55) **Fetch** (section 2.2.3.63.1) **Move** (section 2.2.3.107.1)	**container** ([MS-ASDTYPE] section 2.2)	0…1 (optional)

2.2.3.140.4   Response (MoveItems)

The **Response** element is a required child element of the **MoveItems** element in **MoveItems** command responses that serves as a container for elements that describe the moved items.

All elements referenced in this section are defined in the **Move** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
MoveItems command response (section 2.2.2.10)	**MoveItems** (section 2.2.3.107.2)	**SrcMsgId** (section 2.2.3.160) **Status** (section 2.2.3.162.9) **DstMsgId** (section 2.2.3.50)	**container** ([MS-ASDTYPE] section 2.2)	1…N (required)

2.2.3.140.5   Response (ResolveRecipients)

The **Response** element is an optional child element of the **ResolveRecipients** element in **ResolveRecipients** command responses that contains information as to whether the recipient was resolved. If the recipient was resolved, the element also contains the type of recipient, the email address that the recipient resolved to, and, optionally, the S/MIME certificate for the recipient.

All elements referenced in this section are defined in the **ResolveRecipients** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ResolveRecipients command response (section 2.2.2.13)	**ResolveRecipients** (section 2.2.3.139)	**To** (section 2.2.3.168) **Status** (section 2.2.3.162.11) **RecipientCount** (section 2.2.3.133) **Recipient** (section 2.2.3.132)	**container** ([MS-ASDTYPE] section 2.2)	0…1 (optional)

2.2.3.140.6   Response (Search)

The **Response** element is an optional child element of the **Search** element in **Search** command responses that contains the search results that are returned from the server.

All elements referenced in this section are defined in the **Search** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command response (section 2.2.2.14)	**Search** (section 2.2.3.146)	**Store** (section 2.2.3.163.2)	container ([MS-ASDTYPE] section 2.2)	0...1 (optional)

2.2.3.141 Responses

The **Responses** element is an optional child element of the **Collection** element in **Sync** command responses that contains responses to operations that are processed by the server. Each response is wrapped in an element with the same name as the operation, such as the **Add** element and the **Change** element. The response contains a status code and other information, depending on the operation.

All elements referenced in this section are defined in the **AirSync** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Sync command response (section 2.2.2.19)	**Collection** (section 2.2.3.29.2)	**Add** (section 2.2.3.7.2), **Fetch** (section 2.2.3.63.2) (If the operation succeeded.) **Change** (section 2.2.3.24) (If the operation failed.)	container ([MS-ASDTYPE] section 2.2)	0...1 (optional)

The **Responses** element appears only in responses that are sent from the server to the client. It is present only if the server has processed operation from the client. It is omitted otherwise (for example, if the client requested server changes but had no changes to send to the server). If present, it MUST include at least one child element.

2.2.3.142 Result

The **Result** element is used in the following command responses:

- **MeetingResponse** command responses (section 2.2.2.9)

- **Search** command responses (section 2.2.2.14)

The definition of the **Result** element differs according to the context in which it is used. For more details, see section 2.2.3.142.1 and section 2.2.3.142.2.

2.2.3.142.1 Result (MeetingResponse)

The **Result** element is a required child element of the **MeetingResponse** element in **MeetingResponse** command responses that serves as a container for elements that are sent to the client in the response.

All elements referenced in this section are defined in the **MeetingResponse** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
MeetingResponse command response (section 2.2.2.9)	**MeetingResponse** (section 2.2.3.96)	**RequestId** (section 2.2.3.138) **Status** (section 2.2.3.162.8) **CalendarId** (section 2.2.3.18)	container ([MS-ASDTYPE] section 2.2)	1...N (required)

The **Result** element's child elements identify the meeting request message item on the server and the status of the response to the meeting request. If the meeting request is accepted, the server ID of the calendar item is also returned.

2.2.3.142.2   Result (Search)

The **Result** element is a required child element of the **Store** element in **Search** command responses that serves a container for an individual matching mailbox item.

Elements referenced in this section are defined in the **Search** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command response (section 2.2.2.14)	**Store** (section 2.2.3.163.2)	**airsync:Class** (section 2.2.3.27.4) **LongId** (section 2.2.3.89.3) **airsync:CollectionId** (section 2.2.3.30.4) **Properties** (section 2.2.3.128.2)	container ([MS-ASDTYPE] section 2.2)	1...N (required)

One **Result** element is present for each match that is found. If no matches are found, an empty **Result** element is present in the **Store** container element of the response XML.

Inside the **Result** element, the **Properties** element contains a list of nonempty text properties on the entry.

When the store that is being searched is the mailbox:

- There is one **Result** element for each match that is found in the mailbox. If no matches are found, an empty **Result** element is present in the **Store** container element of the response XML.

- Inside the **Result** element, the **Properties** element contains a list of requested properties for the mailbox item.

When the store that is being searched is the document library:

- The first result that is returned in the **Search** response is the metadata for the Root folder or item to which the **documentlibrary:LinkId** element (section 2.2.3.48) value is pointing. The client can choose to ignore this entry if it does not require it.

- If the **documentlibrary:LinkId** element value in the request points to a folder, the metadata properties of the folder are returned as the first item, and the contents of the folder are returned as subsequent results. The **Range** element (section 2.2.3.130.2) applies to these results with no difference; for example, the index 0 would always be for the root item to which the link is pointing.

- If the **documentlibrary:LinkId** element value in the request points to an item, only one result is returned: the metadata for the item.

- Inside the **Result** element, the **Properties** element contains a list of requested properties for the mailbox item.

2.2.3.143 RightsManagementInformation

The **RightsManagementInformation** element<68> is an optional child element of the **Settings** element in **Settings** command requests and responses. In **Settings** command requests, the **RightsManagementInformation** element is a container node that is used to request rights management information settings. In **Settings** command responses, the **RightsManagementInformation** element is a container node that contains rights management information settings retrieved from the server.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command request (section 2.2.2.16)	**Settings** (section 2.2.3.153.2)	**Get** (section 2.2.3.75)	container ([MS-ASDTYPE] section 2.2)	0…1 (optional)
Settings command response	**Settings**	**Get** **Status** (section 2.2.3.162.14)	container	0…1 (optional)

If the **RightsManagementInformation** element is specified in the request, it MUST include the **Get** child element.

2.2.3.144 SaveInSentItems

The **SaveInSentItems** element is an optional child element of the **SendMail** element in **SendMail** command requests, an optional child element of the **SmartForward** element in **SmartForward** command requests, and an optional child element of the **SmartReply** element in **SmartReply** command requests. In all contexts, the **SaveInSentItems** element specifies whether a copy of the message will be stored in the Sent Items folder. If the **SaveInSentItems** element is present, the message is stored -- if not present, the message is not stored.

All elements referenced in this section are defined in the **ComposeMail** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
SendMail command request (section 2.2.2.15)	**SendMail** (section 2.2.3.148)	None	None	0…1 (optional)
SmartForward command request (section 2.2.2.17)	**SmartForward** (section 2.2.3.154)	None	None	0…1 (optional)

Command request/response	Parent elements	Child elements	Data type	Number allowed
SmartReply command request (section 2.2.2.18)	**SmartReply** (section 2.2.3.155)	None	None	0...1 (optional)

The **SaveInSentItems** 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 <SaveInSentItems/> tag.

2.2.3.145 Schema

The **Schema** element is an optional child element of the **Options** element in **ItemOperations** command requests that specifies the schema of the item to be fetched.

All elements referenced in this section are defined in the **ItemOperations** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ItemOperations command request (section 2.2.2.8) fetch operation	**Options** (section 2.2.3.115.2)	Data elements from the content classes. For details about which of the elements from the content classes can be included, see [MS-ASCAL] section 3.1.5.1, [MS-ASCNTC] section 3.1.5.1, [MS-ASDOC] section 3.1.5.1, [MS-ASEMAIL] section 3.1.5.1, [MS-ASNOTE] section 3.1.5.1 and [MS-ASTASK] section 3.1.5.1.	**container** ([MS-ASDTYPE] section 2.2)	0...N (optional)

The **Schema** element is supported within options when fetching items from a mailbox. It is not supported when the client is retrieving items from a document library or retrieving an attachment.

If the **Schema** element is not specified, the server allows all properties to be retrieved.

2.2.3.146 Search

The **Search** element is a required element in **Search** command requests and responses that identifies the body of the HTTP **POST** as containing a **Search** command (section 2.2.2.14). It is the top-level element in the XML stream.

All elements referenced in this section are defined in the **Search** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command request (section 2.2.2.14)	None	**Store** (section 2.2.3.163.2)	**container** ([MS-ASDTYPE] section 2.2)	1...1 (required)
Search command response	None	**Status** (section 2.2.3.162.12) **Response** (section 2.2.3.140.6)	**container**	1...1 (required)

2.2.3.147 SendDisabled

The **SendDisabled** element<69> is an optional child element of the **Account** element in **Settings** command responses (section 2.2.2.16). The value of this element specifies whether the client can send messages using the given account. The server MAY return this element in **Settings** command responses.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command **UserInformation** response	**Account** (section 2.2.3.2)	None	**boolean** ([MS-ASDTYPE] section 2.1)	0…1 (optional)

The value of the **SendDisabled** element is 1 (**TRUE**) if the client cannot send using the given account; otherwise, 0 (**FALSE**).

2.2.3.148 SendMail

The **SendMail** element is a required element in **SendMail** command requests and responses that identifies the body of the HTTP **POST** as containing a **SendMail** command (section 2.2.2.15). It is the top-level element in the XML stream.

Elements referenced in this section are defined in the **ComposeMail** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
SendMail command request	None	**ClientId** (section 2.2.3.28.1) **AccountId** (section 2.2.3.3.1) **SaveInSentItems** (section 2.2.3.144) **Mime** (section 2.2.3.99) **rm:TemplateID** ([MS-ASRM] section 2.2.2.19.3)	**container** ([MS-ASDTYPE] section 2.2)	1…1 (required)
SendMail command response	None	**Status** (section 2.2.3.162.13)	**container**	1…1 (required)

2.2.3.149 Server

The **Server** element is a required child element of the **Settings** element in **Autodiscover** command responses that encapsulates settings that apply to a particular server in the **Autodiscover** command response.

All elements referenced in this section are defined in the http://schemas.microsoft.com/exchange/autodiscover/mobilesync/requestschema/2006 namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Autodiscover command response (section 2.2.2.1)	**Settings** (section 2.2.3.153.1)	**Type** (section 2.2.3.170.1) **Url** (section 2.2.3.172) **Name** (section 2.2.3.110.1) **ServerData** (section 2.2.3.150)	**container** ([MS-ASDTYPE] section 2.2)	1...N (required)

2.2.3.150   ServerData

The **ServerData** element is an optional child element of the **Server** element in **Autodiscover** command responses that specifies the template name for the client certificate.

All elements referenced in this section are defined in the http://schemas.microsoft.com/exchange/autodiscover/mobilesync/requestschema/2006 namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Autodiscover command response (section 2.2.2.1)	**Server** (section 2.2.3.149)	None	**string** ([MS-ASDTYPE] section 2.6)	0...1 (optional)

The **ServerData** element is a **string** value that is present only when the **Type** element (section 2.2.3.170.1) value is set to "CertEnroll".

2.2.3.151   ServerId

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

- **FolderCreate** command responses (section 2.2.2.2)

- **FolderDelete** command requests (section 2.2.2.3)

- **FolderSync** command responses (section 2.2.2.4)

- **FolderUpdate** command requests (section 2.2.2.5)

- **ItemOperations** command requests and responses (section 2.2.2.8)

- **Sync** command requests and responses (section 2.2.2.19)

The definition of the **ServerId** element differs according to the context in which it is used. For more details, see section 2.2.3.151.1, section 2.2.3.151.2, section 2.2.3.151.3, section 2.2.3.151.4, section 2.2.3.151.5, and section 2.2.3.151.6.

2.2.3.151.1   ServerId (FolderCreate)

The **ServerId** element is an optional child element of the **FolderCreate** element in **FolderCreate** command responses that uniquely identifies a new folder on a server. The **ServerId** of the new folder is returned to the client after a successful **FolderCreate** command request (section 2.2.2.2).

The **ServerId** can also be used in the **ServerId** element of future **FolderDelete** command requests (section 2.2.2.3) and **FolderUpdate** command requests (section 2.2.2.5). The client MUST store the **ServerId** for each folder and MUST be able to locate a folder on the client given a **ServerId**.

All elements referenced in this section are defined in the **FolderHierarchy** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
FolderCreate command response	**FolderCreate** (section 2.2.3.67)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

The **ServerId** element MUST be returned if the **FolderCreate** command request was successful and the element MUST NOT be returned if the **FolderCreate** command request fails. The **ServerId** element value can be up to 64 characters in length.

2.2.3.151.2   ServerId (FolderDelete)

The **ServerId** element is a required child element of the **FolderDelete** element in **FolderDelete** command requests that specifies the folder on the server to be deleted, and it is a unique identifier assigned by the server to each folder that can be synchronized.

All elements referenced in this section are defined in the **FolderHierarchy** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
FolderDelete command request (section 2.2.2.3)	**FolderDelete** (section 2.2.3.68)	None	**string** ([MS-ASDTYPE] section 2.6)	1…1 (required)

The server ID of the folder to be deleted is returned to the client in the **ServerId** element of a previous **FolderSync** command (section 2.2.2.4) or **FolderCreate** command (section 2.2.2.2). The client MUST store the server ID for each folder and MUST be able to locate a folder given a server ID.

The client MUST store the synchronization key as an opaque string of up to 64 characters.

2.2.3.151.3   ServerId (FolderSync)

The **ServerId** element is a required child element of the **Update** element, the **Delete** element, and the **Add** element in **FolderSync** command responses. In all contexts, it specifies the server-unique identifier for a folder on the server.

All elements referenced in this section are defined in the **FolderHierarchy** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
FolderSync command response (section 2.2.2.4)	**Update** (section 2.2.3.171)	None	**string** ([MS-ASDTYPE] section 2.6)	1…1 (required)
FolderSync command response	**Delete** (section 2.2.3.42.1)	None	**string**	1…1 (required)

Command request/response	Parent elements	Child elements	Data type	Number allowed
FolderSync command response	**Add** (section 2.2.3.7.1)	None	**string**	1…1 (required)

The **ServerId** element is used to identify folders that have been updated, deleted, or added on the server.

The client MUST store the server ID as an opaque string of up to 64 characters.

Each **Update** element, each **Delete** element, and each **Add** element included in a **FolderSync** response MUST contain one **ServerId** element.

2.2.3.151.4 ServerId (FolderUpdate)

The **ServerId** element is a required child element of the **FolderUpdate** element in **FolderUpdate** command requests that identifies the folder on the server to be renamed or moved.

All elements referenced in this section are defined in the **FolderHierarchy** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
FolderUpdate command request (section 2.2.2.5)	**FolderUpdate** (section 2.2.3.72)	None	**string** ([MS-ASDTYPE] section 2.6)	1…1 (required)

The server ID is obtained from the **ServerId** element of a previous **FolderSync** command (section 2.2.2.4) or **FolderCreate** command (section 2.2.2.2). The **ServerId** specifies a unique identifier assigned by the server to each object that can be synchronized. The client MUST store the **ServerId** for each folder and MUST be able to locate a folder given a **ServerId**.

The client MUST store the **ServerId** as an opaque string of up to 64 characters.

2.2.3.151.5 ServerId (ItemOperations)

The **airsync:ServerId** element is an optional child element of the **Fetch** element in **ItemOperations** command requests and responses. It specifies a unique identifier that is assigned by the server to each object that can be synchronized or have an item operation applied to it. This element is defined in the **AirSync** namespace.

Elements referenced in this section are defined in the **ItemOperations** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
ItemOperations command request (section 2.2.2.8) fetch operation	**Fetch** (section 2.2.3.63.1)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)
ItemOperations command response fetch operation	**Fetch**	None	**string**	0…1 (optional)

The client MUST store the value of the **airsync:ServerId** element for any item that is retrieved by means of the **Sync** command (section 2.2.2.19) or **Search** command (section 2.2.2.14). In an **ItemOperations** request, the **airsync:ServerId** element can be used by the **Fetch** element to refer to the location of the item in question.

2.2.3.151.6 ServerId (Sync)

The **ServerId** element is a required child element of the **Change** element, the **Delete** element, and the **Fetch** element in **Sync** command requests. It is an optional child element of the **Add** element, the **Change** element, and the **Fetch** element under the **Responses** element (section 2.2.3.141) in **Sync** command responses. It is a required child element of the **Add** element, the **Change** element, the **Delete** element, the **Fetch** element, and the **SoftDelete** element under the **Commands** element (section 2.2.3.32) in **Sync** command responses. It represents a unique identifier that is assigned by the server to each object that can be synchronized. The client MUST store the server ID for each object as an opaque string of up to 64 characters and MUST be able to locate an object given a server ID.

All elements referenced in this section are defined in the **AirSync** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Sync command request (section 2.2.2.19)	**Change** (section 2.2.3.24) **Delete** (section 2.2.3.42.2) **Fetch** (section 2.2.3.63.2)	None	**string** ([MS-ASDTYPE] section 2.6)	1…1 (required)
Sync command response	**Add** (as a child of **Commands**) (section 2.2.3.7) **Change** (as a child of **Commands**) **Fetch** (as a child of **Commands**) **Delete** (as a child of **Commands**)(**SoftDelete** (as a child of **Commands**) (section 2.2.3.157)	None	**string**	1…1 (required)
Sync command response	**Add** (as a child of **Responses**) **Change** (as a child of **Responses**) **Fetch** (as a child of **Responses**)	None	**string**	0…1 (optional)

2.2.3.152 Set

The **Set** element is an optional child element of the **Oof** element, the **DevicePassword** element, and the **DeviceInformation** element in **Settings** command requests. It sets OOF information, device password settings, or device information settings on the server.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command **Oof** request (section 2.2.2.16)	**Oof** (section 2.2.3.112)	**OofState** (section 2.2.3.114) **StartTime** (section 2.2.3.161.2) **EndTime** (section 2.2.3.58.2) **OofMessage** (section 2.2.3.113)	**container** ([MS-ASDTYPE] section 2.2)	0...1 (optional)
Settings command **DevicePassword** request	**DevicePassword** (section 2.2.3.46)	**Password** (section 2.2.3.122.3)	**container**	0...1 (optional)
Settings command **DeviceInformation** request	**DeviceInformation** (section 2.2.3.45)	**Model** (section 2.2.3.105) **IMEI** (section 2.2.3.82) **FriendlyName** (section 2.2.3.74) **OS** (section 2.2.3.117) **OSLanguage** (section 2.2.3.118) **PhoneNumber** (section 2.2.3.124) **UserAgent** (section 2.2.3.174) **EnableOutboundSMS** (section 2.2.3.57) **MobileOperator** (section 2.2.3.103)	**container**	0...1 (optional)

Using the **Settings** command **Oof** request, the client can set the following:

▪ OOF state

▪ Start time and end time, if the user wants to schedule an OOF message

▪ OOF message or messages for one or more of the supported audiences

Using the **Settings** command **DevicePassword** request, the client can set or clear the recovery password of the device.

Using the **Settings** command **DeviceInformation** request, clients SHOULD<70> send **DeviceInformation** parameters to the server as soon as possible after the client has been provisioned, and before the **FolderSync** command (section 2.2.2.4), so that the server can use this information to determine what the device has access to. **Set** enables the client to specify values for any of the **DeviceInformation** parameters. The following statements apply to the **Set** element request implementation:

▪ The client SHOULD specify all supported **DeviceInformation** parameters in the **Set** request. An error is not returned if all **DeviceInformation** parameters are not set if at least one **DeviceInformation** parameter is set.

- The client or server makes no assumptions about ordering. The **DeviceInformation** parameters can be specified in any order.

- To delete a given **DeviceInformation** value, the client MUST send the **Set** element as an empty element.

2.2.3.153 Settings

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

- **Autodiscover** command responses (section 2.2.2.1)

- **Settings** command requests and responses (section 2.2.2.16)

The definition of the **Settings** element differs according to the context in which it is used. For more details, see section 2.2.3.153.1 and section 2.2.3.153.2.

2.2.3.153.1 Settings (Autodiscover)

The **Settings** element is an optional child element of the **Action** element in **Autodiscover** command responses that contains the settings for the specified user or schema.

All elements referenced in this section are defined in the http://schemas.microsoft.com/exchange/autodiscover/mobilesync/requestschema/2006 namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Autodiscover command response (section 2.2.2.1)	**Action** (section 2.2.3.6)	**Server** (section 2.2.3.149)	container ([MS-ASDTYPE] section 2.2)	0...1 (optional)

2.2.3.153.2 Settings (Settings)

The **Settings** element is a required element in **Settings** command requests and responses that identifies the body of the HTTP **POST** as containing a **Settings** command (section 2.2.2.16). It is the top-level element in the XML stream.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command request	None	**RightsManagementInformation** (section 2.2.3.143) **Oof** (section 2.2.3.112) **DeviceInformation** (section 2.2.3.45) **DevicePassword** (section 2.2.3.46) **UserInformation** (section 2.2.3.176)	container ([MS-ASDTYPE] section 2.2)	1...1 (required)
Settings command response	None	**RightsManagementInformation** (section 2.2.3.143) **Oof**	container	1...1 (required)

Command request/response	Parent elements	Child elements	Data type	Number allowed
		DeviceInformation **DevicePassword** **UserInformation** **Status** (section 2.2.3.162.14)		

The **Settings** element encapsulates one or more named property nodes that contain actions and arguments that apply to those properties.

2.2.3.154 SmartForward

The **SmartForward** element is a required element in **SmartForward** command requests and responses that identifies the body of the HTTP **POST** as containing a **SmartForward** command (section 2.2.2.17). It is the top-level element in the XML stream.

Elements referenced in this section are defined in the **ComposeMail** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
SmartForward command request	None	**ClientId** (section 2.2.3.28.1) **Source** (section 2.2.3.158) **AccountId** (section 2.2.3.3.1) **SaveInSentItems** (section 2.2.3.144) **ReplaceMime** (section 2.2.3.135) **Mime** (section 2.2.3.99) **rm:TemplateID** ([MS-ASRM] section 2.2.2.19.3)	container ([MS-ASDTYPE] section 2.2)	1…1 (required)
SmartForward command response	None	**Status** (section 2.2.3.162.15)	container	1…1 (required)

2.2.3.155 SmartReply

The **SmartReply** element is a required element in **SmartReply** command requests and responses that identifies the body of the HTTP **POST** as containing a **SmartReply** command (section 2.2.2.18). It is the top-level element in the XML stream.

Elements referenced in this section are defined in the **ComposeMail** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
SmartReply command request	None	**ClientId** (section 2.2.3.28.1)	container ([MS-ASDTYPE] section	1…1 (required)

Command request/response	Parent elements	Child elements	Data type	Number allowed
		Source (section 2.2.3.158)	2.2)	
		AccountId (section 2.2.3.3.1)		
		SaveInSentItems (section 2.2.3.144)		
		ReplaceMime (section 2.2.3.135)		
		Mime (section 2.2.3.99)		
		rm:TemplateID ([MS-ASRM] section 2.2.2.19.3)		
SmartReply command reply	None	**Status** (section 2.2.3.162.15)	**container**	1...1 (required)

2.2.3.156 SMTPAddress

The **SMTPAddress** element is a required child element of the **EmailAddresses** element in **Settings** command responses that specifies one of the user's email addresses.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command **UserInformation** response (section 2.2.2.16)	**EmailAddresses** (section 2.2.3.54)	None	**string** ([MS-ASDTYPE] section 2.6)	1...N (required)

2.2.3.157 SoftDelete

The **SoftDelete** element is an optional child element of the **Commands** element in **Sync** command responses that deletes an object from the client when it falls outside the **FilterType** (section 2.2.3.64.2) results or it is no longer included as part of the **SyncOptions** (section 2.2.3.115.5) instructions. The object that is soft deleted is identified by its **ServerId** element.

All elements referenced in this section are defined in the **AirSync** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Sync command response (section 2.2.2.19)	**Commands** (section 2.2.3.32)	**ServerId** (section 2.2.3.151.6)	**container** ([MS-ASDTYPE] section 2.2)	0...N (optional)

The **SoftDelete** element contains any items that are filtered out of the **Sync** query due to being outside the **FilterType** date range, or no longer specified as part of the **SyncOptions** instructions. For example, if the **SyncOptions** element no longer specifies SMS class items as being synchronized, the current SMS items on the device will be soft deleted.

2.2.3.158 Source

The **Source** element is a required child element of the **SmartForward** element in **SmartForward** command requests and a required child element of the **SmartReply** element in **SmartReply** command requests. In both contexts, it contains information about the source message.

All elements referenced in this section are defined in the **ComposeMail** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
SmartForward command request (section 2.2.2.17)	**SmartForward** (section 2.2.3.154)	**FolderId** (section 2.2.3.69) **ItemId** (section 2.2.3.84) **LongId** (section 2.2.3.89.4) **InstanceId** (section 2.2.3.83.2)	**container** ([MS-ASDTYPE] section 2.2)	1…1 (required)
SmartReply command request (section 2.2.2.18)	**SmartReply** (section 2.2.3.155)	**FolderId** **ItemId** **LongId** **InstanceId**	**container**	1…1 (required)

2.2.3.159 SrcFldId

The **SrcFldId** element is a required child element of the **Move** element in **MoveItems** command requests that specifies the server ID of the source folder (that is, the folder that contains the items to be moved).

Elements referenced in this section are defined in the **Move** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
MoveItems command request (section 2.2.2.10)	**Move** (section 2.2.3.107.2)	None	**string** ([MS-ASDTYPE] section 2.6)	1…1 (required)

The server ID of the source folder is obtained from the **folderhierarchy:ServerId** element (section 2.2.3.151) of a previous **FolderSync** command (section 2.2.2.4) or **FolderCreate** command (section 2.2.2.2). The **SrcFldId** element value can be up to 64 characters in length.

2.2.3.160 SrcMsgId

The **SrcMsgId** element is a required child element of the **Move** element in **MoveItems** command requests that specifies the server ID of the item to be moved. The **SrcMsgId** element is a required child element of the **Response** element in **MoveItems** command responses that specifies the server ID of the item that was moved. The **SrcMsgId** element value can be up to 64 characters in length.

All elements referenced in this section are defined in the **Move** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
MoveItems command request (section 2.2.2.10)	**Move** (section 2.2.3.107.2)	None	**string** ([MS-ASDTYPE] section 2.6)	1…1 (required)
MoveItems command response	**Response** (section 2.2.3.140.4)	None	**string**	1…1 (required)

2.2.3.161 StartTime

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

▪ **ResolveRecipients** command requests (section 2.2.2.13)

▪ **Settings** command requests and responses (section 2.2.2.16)

The definition of the **StartTime** element differs according to the context in which it is used. For more details, see section 2.2.3.161.1 and section 2.2.3.161.2.

2.2.3.161.1 StartTime (ResolveRecipients)

The **StartTime** element is a required child element of the **Availability** element in **ResolveRecipients** command requests that identifies the start of the span of free/busy time requested by the client.<71>

All elements referenced in this section are defined in the **ResolveRecipients** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ResolveRecipients command request (section 2.2.2.13)	**Availability** (section 2.2.3.16)	None	**datetime** ([MS-ASDTYPE] section 2.3)	1…1 (required)

If the **Availability** element is included in the request, the request MUST also include the **StartTime** and **EndTime** (section 2.2.3.58.1) elements.

If the client sends an invalid **StartTime** element value, then the server returns a **Status** element (section 2.2.3.162.11) value of 5 in the **ResolveRecipients** command response.

2.2.3.161.2 StartTime (Settings)

The **StartTime** element is an optional child element of the **Set** element in **Settings** command responses and an optional child element of the **Get** element in **Settings** command responses. It used with the **EndTime** element (section 2.2.3.58.2) to specify the range of time during which the user is out of office.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command **Oof** request (section 2.2.2.16)	**Set** (section 2.2.3.152)	None	**datetime** ([MS-ASDTYPE] section 2.3)	0...1 (optional)
Settings command **Oof** response	**Get** (section 2.2.3.75)	None	**datetime**	0...1 (optional)

The **StartTime** element can be present within the **Get** element of the **Settings** response for the **Oof** property (section 2.2.3.112), or within the **Set** element of the **Settings** request for the **Oof** property.

In a **Set Oof** node, both **StartTime** and **EndTime** elements MUST be included in the **Settings** request, or neither the **StartTime** element nor the **EndTime** element MUST be included in the **Settings** request. If either the **StartTime** element or the **EndTime** element is included in the request without the other, a **Status** element (section 2.2.3.162.14) value of 2 is returned as a child element of the **Oof** element.

2.2.3.162   Status

The **Status** element is used in the following command responses:

- **Autodiscover** command responses (section 2.2.2.1)

- **FolderCreate** command responses (section 2.2.2.2)

- **FolderDelete** command responses (section 2.2.2.3)

- **FolderSync** command responses (section 2.2.2.4)

- **FolderUpdate** command responses (section 2.2.2.5)

- **GetItemEstimate** command responses (section 2.2.2.7)

- **ItemOperations** command responses (section 2.2.2.8)

- **MeetingResponse** command responses (section 2.2.2.9)

- **MoveItems** command responses (section 2.2.2.10)

- **Ping** command responses (section 2.2.2.11)

- **ResolveRecipients** command responses (section 2.2.2.13)

- **Search** command responses (section 2.2.2.14)

- **SendMail** command responses (section 2.2.2.15)

- **Settings** command responses (section 2.2.2.16)

- **SmartForward** command responses (section 2.2.2.17)

- **SmartReply** command responses (section 2.2.2.18)

- **Sync** command responses (section 2.2.2.19)

- **ValidateCert** command responses (section 2.2.2.20)

The definition of the **Status** element differs according to the context in which it is used. For more details, see section 2.2.3.162.1, section 2.2.3.162.2, section 2.2.3.162.3, section 2.2.3.162.4, section 2.2.3.162.5, section 2.2.3.162.6, section 2.2.3.162.7, section 2.2.3.162.8, section 2.2.3.162.9, section 2.2.3.162.10, section 2.2.3.162.11, section 2.2.3.162.12, section 2.2.3.162.13, section 2.2.3.162.14, section 2.2.3.162.15, section 2.2.3.162.16, and section 2.2.3.162.17.

2.2.3.162.1 Status (Autodiscover)

The **Status** element is an optional child element of the **Error** element in **Autodiscover** command responses that specifies a status code that corresponds to the error.

All elements referenced in this section are defined in the http://schemas.microsoft.com/exchange/autodiscover/mobilesync/requestschema/2006 namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Autodiscover command response (section 2.2.2.1)	**Error** (section 2.2.3.60)	None	integer ([MS-ASDTYPE] section 2.5)	0...1 (optional)

The following table lists the status codes for the **Autodiscover** command. For status values common to all ActiveSync commands, see section 2.2.4.

Value	Meaning
1	Success. Because the **Status** element is only returned when the command encounters an error, the success status code is never included in a response message.
2	Protocol error

The client device can implement custom recovery logic pertaining to the status code. If an unknown status code is returned to the client, the client SHOULD have logic in place to handle the error by sending an error message to the user, resending the command with new settings, or custom logic.

2.2.3.162.2 Status (FolderCreate)

The **Status** element is a required child element of the **FolderCreate** element in **FolderCreate** command responses that indicates the success or failure of a **FolderCreate** command request (section 2.2.2.2). If the command failed, the **Status** element contains a code indicating the type of failure.

All elements referenced in this section are defined in the **FolderHierarchy** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
FolderCreate command response	**FolderCreate** (section 2.2.3.67)	None	unsignedByte ([MS-ASDTYPE] section 2.7)	1...1 (required)

The following table lists the status codes for the **FolderCreate** command (section 2.2.2.2). For information about the scope of the status value and for status values common to all ActiveSync commands, see section 2.2.4.

Value	Meaning	Cause	Scope	Resolution
1	Success.	Server successfully completed command.	Global	None.
2	A folder that has this name already exists.	The parent folder already contains a folder that has this name.	Item	Prompt user to supply a unique name.
3	The specified parent folder is a special system folder.	The specified parent folder is the Recipient information folder.	Item	Create the folder under a different parent.
5	The specified parent folder was not found.	The parent folder does not exist on the server, possibly because it has been deleted or renamed.	Item	Issue a **FolderSync** command (section 2.2.2.4) for the new hierarchy and prompt the user for a new parent folder.
6	An error occurred on the server.	Server misconfiguration, temporary system issue, or bad item. This is frequently a transient condition.	Global	Retry the **FolderSync** command. If continued attempts to synchronization fail, consider returning to synchronization key zero (0).
9	Synchronization key mismatch or invalid synchronization key.	The client sent a malformed or mismatched synchronization key, or the synchronization state is corrupted on the server.	Global	Delete folders added since last synchronization and return to synchronization key to zero (0).
10	Malformed request.	The client sent a **FolderCreate** command request (section 2.2.2.2) that contains a semantic error, or the client attempted to create a default folder, such as the Inbox folder, **Outbox folder**, or **Contacts folder**.	Global	Double-check the request for accuracy.
11	An unknown error occurred.	Unknown.	Global.	None.
12	Code unknown.	Unusual back-end issue.	Global	None.

2.2.3.162.3   Status (FolderDelete)

The **Status** element is a required child element of the **FolderDelete** element in **FolderDelete** command responses that indicates the success or failure of the **FolderDelete** command request (section 2.2.2.3). If the command failed, the **Status** element in the server response contains a code indicating the type of failure.

All elements referenced in this section are defined in the **FolderHierarchy** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
FolderDelete command response	FolderDelete (section 2.2.3.68)	None	unsignedByte ([MS-ASDTYPE] section 2.7)	1…1 (required)

The following table lists the status codes for the **FolderDelete** command (section 2.2.2.3). For information about the scope of the status value and for status values common to all ActiveSync commands, see section 2.2.4.

Value	Meaning	Cause	Scope	Resolution
1	Success.	Server successfully completed command.	Global	None.
3	The specified folder is a special system folder, such as the Inbox folder, Outbox folder, Contacts folder, Recipient information, or **Drafts folder**, and cannot be deleted by the client.	The client specified a special folder in a **FolderDelete** command request (section 2.2.2.3). special folders cannot be deleted.	Item	None.
4	The specified folder does not exist.	The client specified a nonexistent folder in a **FolderDelete** command request.	Item	Issue a **FolderSync** command (section 2.2.2.4) for the new hierarchy.
6	An error occurred on the server.	Server misconfiguration, temporary system issue, or bad item. This is frequently a transient condition.	Global	Retry the **FolderDelete** command. If continued attempts to synchronization fail, consider returning to synchronization key zero (0).
9	Synchronization key mismatch or invalid synchronization key.	The client sent a malformed or mismatched synchronization key, or the synchronization state is corrupted on the server.	Global	Issue a **FolderSync** command request with a synchronization key of zero (0).
10	Incorrectly formatted request.	The client sent a **FolderDelete** command request (section 2.2.2.3) that contains a semantic or syntactic error.	Global	Double-check the request for accuracy.
11	An unknown error occurred.	Unusual back-end issue.	Global	None.

2.2.3.162.4   Status (FolderSync)

The **Status** element is a required child element of the **FolderSync** element in **FolderSync** command responses that indicates the success or failure of a **FolderSync** command request (section 2.2.2.4).

All elements referenced in this section are defined in the **FolderHierarchy** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
FolderSync command response	**FolderSync** (section 2.2.3.71)	None	**unsignedByte** ([MS-ASDTYPE] section 2.7)	1…1 (required)

If the command fails, the **Status** element contains a code that indicates the type of failure. The **Status** element is global for all collections. If one collection fails, a failure status MUST be returned for all collections.

The following table lists the status codes for the **FolderSync** command (section 2.2.2.4). For information about the scope of the status value and for status values common to all ActiveSync commands, see section 2.2.4.

Value	Meaning	Cause	Scope	Resolution
1	Success	Server successfully completed command.	Global	None.
6	An error occurred on the server.	Server misconfiguration, temporary system issue, or bad item. This is frequently a transient condition.	Global	Retry the **FolderSync** command. If continued attempts to synchronization fail, consider returning to synchronization key zero (0).
9	Synchronization key mismatch or invalid synchronization key.	The client sent a malformed or mismatched synchronization key, or the synchronization state is corrupted on the server.	Global	Delete items added since last synchronization and return to synchronization key zero (0).
10	Incorrectly formatted request.	The client sent a **FolderSync** command request that contains a semantic or syntactic error.	Global	Double-check the request for accuracy.
11	An unknown error occurred.	Server misconfiguration, temporary system issue, or bad item. This is frequently a transient condition.	Global	Retry the **FolderSync** command. If continued attempts to synchronization fail, consider returning to synchronization key zero (0).
12	Code unknown.	Unusual back-end issue.	Global	None.

2.2.3.162.5 Status (FolderUpdate)

The **Status** element is a required child element of the **FolderUpdate** element in **FolderUpdate** command responses that indicates the success or failure of a **FolderUpdate** command request (section 2.2.2.5).

All elements referenced in this section are defined in the **FolderHierarchy** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
FolderUpdate command response	**FolderUpdate** (section 2.2.3.72)	None	**unsignedByte** ([MS-ASDTYPE] section 2.7)	1…1 (required)

Release: February 10, 2014

If the command fails, the **Status** element contains a code that indicates the type of failure.

The following table lists the status codes for the **FolderUpdate** command (section 2.2.2.5). For information about the scope of the status value and for status values common to all ActiveSync commands, see section 2.2.4.

Code	Meaning	Cause	Scope	Resolution
1	Success.	Server successfully completed command.	Global	None.
2	A folder with that name already exists or the specified folder is a special folder.	A folder with that name already exists or the specified folder is a special folder, such as the Inbox, Outbox, Contacts, or Drafts folders. Special folders cannot be updated.	Item	None.
3	The specified folder is the Recipient information folder, which cannot be updated by the client.	The client specified the Recipient information folder, which is a special folder. Special folders cannot be updated.	Item	None.
4	The specified folder does not exist.	Client specified a nonexistent folder in a **FolderUpdate** command request.	Item	Issue a **FolderSync** command (section 2.2.2.4) for the new hierarchy.
5	The specified parent folder was not found.	Client specified a nonexistent folder in a **FolderUpdate** command request.	Item	Issue a **FolderSync** command for the new hierarchy.
6	An error occurred on the server.	Server misconfiguration, temporary system issue, or bad item. This is frequently a transient condition.	Global	Retry the **FolderUpdate** command. If continued attempts to synchronization fail, consider returning to synchronization key 0.
9	Synchronization key mismatch or invalid synchronization key.	The client sent a malformed or mismatched synchronization key, or the synchronization state is corrupted on the server.	Global	Issue a **FolderSync** command request with a synchronization key of 0.
10	Incorrectly formatted request.	The client sent a **FolderUpdate** command request that contains a semantic error.	Global	Double-check the request for accuracy.
11	An unknown error occurred.	Unusual back-end issue.	Global	None.

2.2.3.162.6 Status (GetItemEstimate)

The **Status** element is a required child element of the **Response** element in **GetItemEstimate** command responses that indicates the success or failure of part or all of a **GetItemEstimate** command request (section 2.2.2.7).

Elements referenced in this section are defined in the **GetItemEstimate** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
GetItemEstimate command response	**GetItemEstimate** (section 2.2.3.77)	None	**unsignedByte** ([MS-ASDTYPE] section 2.7)	0…1 (optional)
GetItemEstimate command response	**Response** (section 2.2.3.140.2)	None	**unsignedByte**	1…1 (required)

If the entire request command fails, the **Status** element is present as a child of the **GetItemEstimate** element and contains a value that indicates the type of global failure. However, if the failure occurs at the **Collection** (section 2.2.3.29.1) level, a **Status** value is returned per **Collection** as a child of the **Response** element.

The following table lists the status codes for the **GetItemEstimate** command (section 2.2.2.7). For information about the scope of the status value and for status values common to all ActiveSync commands, see section 2.2.4.

Value	Meaning	Cause	Scope	Resolution
1	Success.	Server successfully completed command.	Global	None.
2	A collection was invalid or one of the specified collection IDs was invalid.	One or more of the specified folders does not exist or an incorrect folder was requested.	Item	Issue a **FolderSync** command (section 2.2.2.4) to get the new hierarchy. Then retry with a valid collection or collection ID.
3	The synchronization state has not been primed.	The client has issued a **GetItemEstimate** command without first issuing a **Sync** command request (section 2.2.2.19) with an **airsync:SyncKey** element (section 2.2.3.166.4) value of zero (0).	Item	Issue a **Sync** command with synchronization key of zero (0) before issuing the **GetItemEstimate** command again.
4	The specified synchronization key was invalid.	Malformed or mismatched synchronization key. —or— The synchronization state is corrupted on the server.	Global	Issue a successful **Sync** command prior to issuing the **GetItemEstimate** command again. If the error is repeated, issue a **Sync** command with an **airsync:SyncKey** element value of zero (0).

2.2.3.162.7   Status (ItemOperations)

The **Status** element is a required child element of the **ItemOperations** element, the **Move** element, the **EmptyFolderContents** element, and the **Fetch** element in **ItemOperations** command responses. It contains a code that indicates the success or failure of the **ItemOperations** command request (section 2.2.2.8) and the operations within the **ItemOperations** command request.

All elements referenced in this section are defined in the **ItemOperations** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ItemOperations command response	**ItemOperations** (section 2.2.3.85)	None	**integer** ([MS-ASDTYPE] section 2.5)	1...1 (required)
ItemOperations command response **Move** operation	**Move** (section 2.2.3.107.1)	None	**integer**	1...1 (required)
ItemOperations command response **EmptyFolderContents** operation	**EmptyFolderContents** (section 2.2.3.55)	None	**integer**	1...1 (required)
ItemOperations command response fetch operation	**Fetch** (section 2.2.3.63.1)	None	**integer**	1...1 (required)

The following table lists the status codes for the **ItemOperations** command (section 2.2.2.8). For information about status values common to all ActiveSync commands, see section 2.2.4.

Status code	Meaning
1	Success.
2	Protocol error - protocol violation/XML validation error.
3	Server error.
4	Document library access - The specified URI is bad.
5	Document library - Access denied.
6	Document library - The object was not found or access denied.
7	Document library - Failed to connect to the server.
8	The byte-range is invalid or too large.
9	The store is unknown or unsupported.
10	The file is empty.
11	The requested data size is too large.
12	Failed to download file because of input/output (I/O) failure.
14	Mailbox fetch provider - The item failed conversion.
15	Attachment fetch provider - Attachment or attachment ID is invalid.
16	Access to the resource is denied.
17	Partial success; the command completed partially.
18	Credentials required.

Status code	Meaning
155<72>	Protocol error. The **Options** element (section 2.2.3.115.2) and the **MoveAlways** element (section 2.2.3.108) are missing from the **ItemOperations** request.
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.

2.2.3.162.8   Status (MeetingResponse)

The **Status** element is a required child element of the **Result** element in **MeetingResponse** command responses that indicates the success or failure of the **MeetingResponse** command request (section 2.2.2.9).

All elements referenced in this section are defined in the **MeetingResponse** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
MeetingResponse command response (section 2.2.2.9)	**Result** (section 2.2.3.142.1)	None	**integer** ([MS-ASDTYPE] section 2.5)	1...1 (required)

The following table lists the status codes for the **MeetingResponse** command (section 2.2.2.9). For information about the scope of the status value and for status values common to all ActiveSync commands, see section 2.2.4.

Value	Meaning	Cause	Scope	Resolution
1	Success.	Server successfully completed command.	Global	None.
2	Invalid meeting request.	The client has sent a malformed or invalid item. The request is referencing an item other than a meeting request, email, or calendar item. The request points to an appointment in which the user is the organizer. The **InstanceId** element (section 2.2.3.83.1) specifies an email meeting request item. The **InstanceId** element specifies a nonexistent instance or is null.	Item	Stop sending the item. This is not a transient condition.
3	An error occurred on the server mailbox.	Server misconfiguration, temporary system issue, or bad item. This is frequently a transient condition.	Global	Retry the **MeetingResponse** command. If continued attempts fail, synchronize the folder again, and then attempt the **MeetingResponse** command again. If it still continues to fail, make no changes.

Value	Meaning	Cause	Scope	Resolution
4	An error occurred on the server.	Server misconfiguration, temporary system issue, or bad item. This is frequently a transient condition.	Global	Retry the **MeetingResponse** command. If continued attempts fail, synchronize the folder again, and then attempt the **MeetingResponse** command again. If it still continues to fail, make no changes.

2.2.3.162.9 Status (MoveItems)

The **Status** element is a required child element of the **MoveItems** element in **MoveItems** command responses that indicates the success or failure the **MoveItems** command request (section 2.2.2.10). If the command failed, **Status** contains a code indicating the type of failure.

All elements referenced in this section are defined in the **Move** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
MoveItems command response	**Response** (section 2.2.3.140.4)	None	**unsignedByte** ([MS-ASDTYPE] section 2.7)	1…1 (required)

The following table lists the status codes for the **MoveItems** command (section 2.2.2.10). For information about the scope of the status value and for status values common to all ActiveSync commands, see section 2.2.4.

Value	Meaning	Cause	Scope	Resolution
1	Invalid source collection ID or invalid source Item ID.	The source folder collection ID (**CollectionId** element (section 2.2.3.30.5) value) is not recognized by the server, possibly because the source folder has been deleted. Or, the item with the Item ID (**SrcMsgId** element (section 2.2.3.160)) has been previously moved out of the folder with the Folder ID (**SrcFldId** element (section 2.2.3.159)).	Item	Issue a **FolderSync** command (section 2.2.2.4) to get the new hierarchy. Then, issue a **Sync** command (section 2.2.2.19) for the **SrcFldId** and reissue the **MoveItems** command request if the items are still present in this source collection.
2	Invalid destination collection ID.	The destination folder collection ID (**CollectionId** element value) is not recognized by the server, possibly because the source folder has been deleted.	Item	Issue a **FolderSync** command to get the new hierarchy. Then, use a valid collection ID.
3	Success.	Server successfully completed command.	Global	None.
4	Source and destination collection IDs are the same.	The client supplied a destination folder that is the same as the source.	Item	Send only requests where the **CollectionId** element values for the source and destination differ.
5	One of the following failures occurred: the item	More than one **DstFldId** element (section 2.2.3.49.2) was included in the request or an item with	Global	Retry the **MoveItems** command request with only one **DstFldId**

Value	Meaning	Cause	Scope	Resolution
	cannot be moved to more than one item at a time, or the source or destination item was locked.	that name already exists.		element or move the item to another location.
7	Source or destination item was locked.	Transient server condition.	Item	Retry the **MoveItems** command request.

2.2.3.162.10 Status (Ping)

The **Status** element is a required child element of the **Ping** element in **Ping** command responses that indicates the success or failure of the **Ping** command request (section 2.2.2.11). If the command failed, the **Status** element contains a code that indicates the type of failure. Certain status codes have additional information that is included in the response.

All elements referenced in this section are defined in the **Ping** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Ping command response	**Ping** (section 2.2.3.126)	None	**unsignedByte** ([MS-ASDTYPE] section 2.7)	1…1 (required)

The following table lists the status codes for the **Ping** command (section 2.2.2.11). For information about the scope of the status value and for status values common to all ActiveSync commands, see section 2.2.4.

Value	Meaning	Cause	Scope	Resolution
1	The heartbeat interval expired before any changes occurred in the folders being monitored.		Global	Reissue the **Ping** command request.
2	Changes occurred in at least one of the monitored folders. The response specifies the changed folders.		Global	Issue a **Sync** command request (section 2.2.2.19) for each folder that was specified in the **Ping** command response to retrieve the server changes. Reissue the **Ping** command when the **Sync** command completes to stay up to date.
3	The **Ping** command request omitted required parameters.	The **Ping** command request did not specify all the necessary parameters. The client MUST issue a **Ping** request that includes both the heartbeat interval and the folder list at	Global	Reissue the **Ping** command request with the entire XML body.

Value	Meaning	Cause	Scope	Resolution
		least once. The server saves the heartbeat interval value (section 2.2.3.79.1), so only the folder list is required on subsequent requests.		
4	Syntax error in **Ping** command request.	Frequently caused by poorly formatted WBXML.	Global	Double-check the request for accuracy.
5	The specified heartbeat interval is outside the allowed range. For intervals that were too short, the response contains the shortest allowed interval. For intervals that were too long, the response contains the longest allowed interval.	The client sent a **Ping** command request with a heartbeat interval that was either too long or too short.	Global	Reissue the **Ping** command by using a heartbeat interval inside the allowed range. Setting the interval to the value returned in the **Ping** response will most closely accommodate the original value specified.
6	The **Ping** command request specified more than the allowed number of folders to monitor. The response indicates the allowed number in the **MaxFolders** element (section 2.2.3.92).	The client sent a **Ping** command request that specified more folders than the server is configured to monitor.	Global	Direct the user to select fewer folders to monitor. Resend the **Ping** command request with the new, shorter list.
7	Folder hierarchy sync required.	The folder hierarchy is out of date; a folder hierarchy sync is required.	Global	Issue a **FolderSync** command (section 2.2.2.4) to get the new hierarchy and prompt the user, if it is necessary, for new folders to monitor. Reissue the **Ping** command.
8	An error occurred on the server.	Server misconfiguration, temporary system issue, or bad item. This is frequently a transient condition.	Global	Retry the **Ping** command.

2.2.3.162.11   Status (ResolveRecipients)

The **Status** element is a required child element of the **ResolveRecipients** element, the **Response** element, the **Availability** element, the **Certificates** element, and the **Picture** element in **ResolveRecipients** command responses.

All elements referenced in this section are defined in the **ResolveRecipients** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ResolveRecipients command response (section 2.2.2.13)	**ResolveRecipients** (section 2.2.3.139) **Response** (section 2.2.3.140.5) **Availability** (section 2.2.3.16) **Certificates** (section 2.2.3.23.1) **Picture** (section 2.2.3.125.1)	None	**unsignedByte** ([MS-ASDTYPE] section 2.7)	1…1 (required)

As a child element of the **ResolveRecipients** element, the **Status** element indicates the success or failure of the **ResolveRecipients** command request. The following table shows valid values for the **Status** element when it is returned as a child of the **ResolveRecipients** element.

Value	Meaning
1	Success.
5	Protocol error. Either an invalid parameter was specified or the range exceeded limits.
6	An error occurred on the server. The client SHOULD retry the request.

As a child element of the **Response** element, the **Status** element provides the status of the **ResolveRecipients** command response **Response** element. The following table shows valid values for the **Status** element when it is returned as a child element of the **Response** element.

Value	Meaning
1	The recipient was resolved successfully. For more details about the **Recipient** element, see section 2.2.3.132.
2	The recipient was found to be ambiguous. The returned list of recipients are suggestions. No certificate nodes were returned. Prompt the user to select the intended recipient from the list returned.
3	The recipient was found to be ambiguous. The returned list is a partial list of suggestions. The total count of recipients can be obtained from the **RecipientCount** element (section 2.2.3.133). No certificate nodes were returned. Prompt the user to select the intended recipient from the list returned or to get more recipients.
4	The recipient did not resolve to any contact or GAL entry. No certificates were returned. Inform the user of the error and direct the user to check the spelling.

As a child element of the **Availability** element, the **Status** element provides the status of the **ResolveRecipients** command response **Availability** element. The following table shows valid values for the **Status** element when it is returned as a child element of the **Availability** element.<73>

Value	Meaning
1	Free/busy data was successfully retrieved for a given recipient. This value does not indicate that the response is complete.

Value	Meaning
160	There were more than 100 recipients identified by the **To** elements (section 2.2.3.168) in the **ResolveRecipient** request.
161	The distribution group identified by the **To** element of the **ResolveRecipient** request included more than 20 recipients.
162	The free/busy data could not be retrieved by the server due to a temporary failure. The client SHOULD reissue the request. This error is caused by a timeout value being reached while requesting free/busy data for some users, but not others.
163	Free/busy data could not be retrieved from the server for a given recipient. Clients SHOULD NOT reissue the request as it is caused by a lack of permission to retrieve the data.

As a child element of the **Certificates** element, the **Status** element provides the status of the **ResolveRecipients** command response **Certificates** element. The following table shows valid values for the **Status** element when it is returned as a child element of the **Certificates** element.

Value	Meaning
1	One or more certificates were successfully returned.
7	The recipient does not have a valid S/MIME certificate. No certificates were returned.
8	The global certificate limit was reached and the recipient's certificate could not be returned. The count certificates not returned can be obtained from the **CertificateCount** element. Retry with fewer recipients if possible, otherwise prompt the user.

As a child element of the **Picture** element, the **Status** element provides the status of the **ResolveRecipients** command response **Picture** element. The following table shows valid values for the **Status** element when it is returned as a child element of the **Picture** element.<74>

Value	Meaning
1	The contact photo was retrieved successfully.
173	The user does not have a contact photo.
174	The contact photo exceeded the size limit set by the **MaxSize** element (section 2.2.3.95.1).
175	The number of contact photos returned exceeded the size limit set by the **MaxPictures** element (section 2.2.3.94.1).

For information about **Status** values common to all ActiveSync commands, see section 2.2.4.

2.2.3.162.12   Status (Search)

The **Status** element is a required child element of the **Search** element, the **Store** element, and the **gal:Picture** element in **Search** command responses.

Elements referenced in this section are defined in the **Search** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command response (section 2.2.2.14)	**Search** (section 2.2.3.146) **Store** (section 2.2.3.163.2) **gal:Picture** (section 2.2.3.125.2)	None	**integer** ([MS-ASDTYPE] section 2.5)	1…1 (required)

As a child element of the **Search** element, the **Status** element indicates whether the server encountered an error while it was processing the search query. The **Status** element value indicates only that the overall **Search** command request (section 2.2.2.14) was processed correctly. It does not indicate whether any matches were found. The **Total** (section 2.2.3.169.2) and **Range** (section 2.2.3.130.2) elements in the **Search** command response indicate how many matches were found and returned, respectively. The following table specifies valid values for the **Status** element when it is returned as a child element of the **Search** element.

Value	Meaning
1	Success
3	Server error

As a child element of the **Store** element, the **Status** element indicates whether the server encountered an error while it was processing the **Search** query for the specified store. The following table specifies valid values for the **Status** element as a child of the **Store** element in the **Search** response.

Value	Meaning	Cause	Scope	Resolution
1	Success.	Server successfully completed command.	Global	None.
2	The request was invalid.	One or more of the client's search parameters was invalid.	Item	If the user formatted the request, prompt the user to retry with different options.
3	An error occurred on the server.	Server misconfiguration, temporary system issue, or bad item. This is frequently a transient condition.	Global	Retry the **Search** command.
4	Bad link.	A bad link was supplied.	Global	Prompt user to reformat link.
5	Access denied.	Access was denied to the resource	Global	Prompt the user.
6	Not found.	Resource was not found.	Global	Prompt the user.
7	Connection failed.	Failed to connect to the resource.	Global	Prompt the user. Sometimes these are transient, so retry. If it continues to fail, point user to administrator.
8	Too complex.	The query was too complex.	Global	Reduce the complexity of the

Release: February 10, 2014

Value	Meaning	Cause	Scope	Resolution
				query. Prompt user if necessary.
10	Timed out.	The search timed out.	Global	The search timed out. Retry with or without rebuilding results. If it continues, contact the Administrator.
11	FolderSync required.	The folder hierarchy is out of date.	Global	Issue a **FolderSync** command (section 2.2.2.4) and then try the **Search** command again.
12	End of retrievable range warning.	The requested range has gone past the end of the range of retrievable results.	Local	Prompt the user that there are no more results that can be fetched, and the user might consider restricting their search query.
13	Access blocked.	Access is blocked to the specified resource.	Global	Prompt the user.
14	Credentials required.	To complete this request, basic credentials are required.	Global	If over a trusted connection, supply the basic credentials from the user (prompt if necessary). Otherwise, fail as if the access denied status code was provided.

As a child element of the **gal:Picture** element, the **Status** element indicates whether the server encountered an error while it was processing the **Search** query for the contact photos. The following table shows valid values for the **Status** element when it is returned as a child of the **gal:Picture** element.<75>

Value	Meaning
1	The contact photo was retrieved successfully.
173	The user does not have a contact photo.
174	The contact photo exceeded the size limit set by the **MaxSize** element (section 2.2.3.95.2).
175	The number of contact photos returned exceeded the size limit set by the **MaxPictures** element (section 2.2.3.94.2).

For information about **Status** values common to all ActiveSync commands, see section 2.2.4.

2.2.3.162.13   Status (SendMail)

The **Status** element is an optional child element of the **SendMail** element in **SendMail** command responses that indicates the reason for the failure of a **SendMail** command request (section 2.2.2.15).

All elements referenced in this section are defined in the **ComposeMail** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
SendMail command response	**SendMail** (section 2.2.3.148)	None	**integer** ([MS-ASDTYPE] section	0...1 (optional)

Release: February 10, 2014

Command request/response	Parent elements	Child elements	Data type	Number allowed
			2.5)	

If the command succeeds, no XML body is returned in the response. If the command fails, the **Status** element contains a code that indicates the type of failure. Valid **Status** values are listed in section 2.2.4.

2.2.3.162.14   Status (Settings)

The **Status** element is a required child element of the **Settings** element, the **RightsManagementInformation** element, the **Oof** element, the **DevicePassword** element, and the **UserInformation** element in **Settings** command responses. It contains a code that indicates either the overall success or failure of the **Settings** command request (section 2.2.2.16) or the success or failure of a **Get** (section 2.2.3.75) or **Set** (section 2.2.3.152) operation.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command response	**Settings** (section 2.2.3.153.2) **RightsManagementInformation** (section 2.2.3.143) **Oof** (section 2.2.3.112) **DevicePassword** (section 2.2.3.46) **DeviceInformation** (section 2.2.3.45) **UserInformation** (section 2.2.3.176)	None	**integer** ([MS-ASDTYPE] section 2.5)	1 (required)

As a child element of the **Settings** element in the **Settings** command response, the **Status** element indicates the overall success or failure of the **Settings** command request. The following table lists the valid values for the **Status** element as the child element of the **Settings** element in the **Settings** command response.

Value	Meaning
1	Success.
2	Protocol error.
3	Access denied.
4	Server unavailable.
5	Invalid arguments.
6	Conflicting arguments
7	Denied by policy

Error code values 100 to 255 are reserved for property-specific error codes and vary from property to property. Any status value that is not 1 is a failure.

The following table lists the valid values for **Status** in a **Settings** command **RightsManagementInformation Get** operation, **Oof Get** operation, **Oof Set** operation response, **DeviceInformation Set** operation or **UserInformation Get** operation.

Value	Meaning
1	Success.
2	Protocol error. The XML code is formatted incorrectly.
5	Invalid arguments.
6	Conflicting arguments.

For a **Settings** command **Oof Set** operation, if **Oof** nodes **AppliesToExternalKnown** (section 2.2.3.12) or **AppliesToExternalUnknown** (section 2.2.3.13) are not allowed and are disabled by the administrator but are sent by the client in the **Settings** command **Set** operation request, the **Settings** command **Set** operation response returns a successful **Status** element value of 1 even though the user does not have access to these settings.

The following table lists the values for **Status** in a **Settings** command **DevicePassword Set** response.

Value	Meaning
1	Success.
2	Protocol error. The XML code is formatted incorrectly.
5	Invalid arguments. The specified password is too long.
7	Denied by policy. The administrator has disabled password recovery in this deployment.

For **Status** values common to all ActiveSync commands, see section 2.2.4.

2.2.3.162.15   Status (SmartForward and SmartReply)

The **Status** element is an optional child element of the **SmartForward** element in **SmartForward** command responses and an optional child element of the **SmartReply** element in **SmartReply** command responses. It indicates the reason for the failure of a **SmartForward** command request (section 2.2.2.17) or **SmartReply** command request (section 2.2.2.18).

All elements referenced in this section are defined in the **ComposeMail** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
SmartForward command response	**SmartForward** (section 2.2.3.154)	None	**integer** ([MS-ASDTYPE] section 2.5)	0…1 (optional)
SmartReply command response	**SmartReply** (section 2.2.3.155)	None	**integer**	0…1 (optional)

If the **SmartForward** command request or **SmartReply** command request succeeds, no XML body is returned in the response. If the **SmartForward** command request or **SmartReply** command

request fails, the **Status** element contains a code that indicates the type of failure. Valid **Status** element values are listed in section 2.2.4. In particular, a **Status** element value of 117 indicates that the server does not allow a reply to the message.

2.2.3.162.16 Status (Sync)

The **Status** element is a required child element of the **Collection** element, the **Change** element, the **Add** element, and the **Fetch** element in **Sync** command responses. It is an optional child element of the **Sync** element in **Sync** command responses.

Elements referenced in this section are defined in the **AirSync** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
Sync command response (section 2.2.2.19)	**Collection** (section 2.2.3.29.2) **Change** (section 2.2.3.24) **Add** (section 2.2.3.7.2) **Fetch** (section 2.2.3.63.2)	None	**integer** ([MS-ASDTYPE] section 2.5)	1…1 (required)
Sync command response	**Sync** (section 2.2.3.165)	None	**integer**	0…1 (optional)

As a child element of the **Sync** element, the **Status** element indicates the success or failure of the **Sync** command request. If the request failed, the **Status** element contains a code that indicates the type of failure.

As a child element of the **Collection** element, the **Status** element indicates the success or failure of the **Sync** command operation on the collection. If the operation failed, the **Status** element contains a code that indicates the type of failure.

As a child element of the **Change** element, the **Add** element, or the **Fetch** element, the **Status** element indicates the success or failure of the change operation, the add operation, or the fetch operation, respectively. If the operation failed, the **Status** element contains a code that indicates the type of failure.

The following table lists the status codes for the **Sync** command (section 2.2.2.19). For information about the scope of the status value and for status values common to all ActiveSync commands, see section 2.2.4.

Value	Meaning	Cause	Scope	Resolution
1	Success.	Server successfully completed command.	Global	None.
3	Invalid synchronization key.	Invalid or mismatched synchronization key. —or— Synchronization state	Global	MUST return to **SyncKey** element value (section 2.2.3.166.4) of 0 for the collection. The client SHOULD either delete any

Value	Meaning	Cause	Scope	Resolution
		corrupted on server.		items that were added since the last successful Sync or the client MUST add those items back to the server after completing the full resynchronization.
4	Protocol error.	There was a semantic error in the synchronization request. The client is issuing a request that does not comply with the specification requirements.	Item or Global	Double-check the request for accuracy and retry the **Sync** command.
5	Server error.	Server misconfiguration, temporary system issue, or bad item. This is frequently a transient condition.	Global	Retry the synchronization. If continued attempts to synchronization fail, consider returning to synchronization key 0.
6	Error in client/server conversion.	The client has sent a malformed or invalid item.	Item	Stop sending the item. This is not a transient condition.
7	Conflict matching the client and server object.	The client has changed an item for which the conflict policy indicates that the server's changes take precedence.	Item	If it is necessary, inform the user that the change they made to the item has been overwritten by a server change.
8	Object not found.	The client issued a fetch or change operation that has an **ItemID** (section 2.2.3.84) value that is no longer valid on the server (for example, the item was deleted).	Item	Issue a synchronization request and prompt the user if necessary.
9	The **Sync** command cannot be completed.	User account could be out of disk space.	Item	Free space in the user's mailbox and retry the **Sync** command.
12	The folder hierarchy has changed.	Mailbox folders are not synchronized.	Global	Perform a **FolderSync** command (section 2.2.2.4) and then retry the **Sync** command.
13	The **Sync** command request is not complete.	An empty or partial **Sync** command request is received and the cached set of notify-able collections is missing.	Item	Resend a full **Sync** command request.
14	Invalid **Wait** or **HeartbeatInterval** value.	The **Sync** request was processed successfully but the wait interval (**Wait** element value (section 2.2.3.182)) or heartbeat interval (**HeartbeatInterval** element value (section 2.2.3.79.2)) that is specified by the client is outside the range set by the server administrator.	Item	Update the **Wait** element value according to the **Limit** element and then resend the **Sync** command request.

Value	Meaning	Cause	Scope	Resolution
		If the **HeartbeatInterval** element value or **Wait** element value included in the **Sync** request is larger than the maximum allowable value, the response contains a **Limit** element that specifies the maximum allowed value. If the **HeartbeatInterval** element value or **Wait** value included in the **Sync** request is smaller than the minimum allowable value, the response contains a **Limit** element that specifies the minimum allowed value.		
15	Invalid **Sync** command request.	Too many collections are included in the **Sync** request.	Item	Notify the user and synchronize fewer folders within one request.
16	Retry	Something on the server caused a retriable error.	Global	Resend the request.

2.2.3.162.17 Status (ValidateCert)

The **Status** element is a required child element of the **ValidateCert** element and the **Certficate** element in **ValidateCert** command responses that indicates whether one or more certificates were successfully validated.

All elements referenced in this section are defined in the **ValidateCert** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ValidateCert command response (section 2.2.2.20)	**ValidateCert** (section 2.2.3.179) **Certificate** (section 2.2.3.19)	None	integer ([MS-ASDTYPE] section 2.5)	1…N (required)

As a child of the **ValidateCert** element, the **Status** element indicates the success or failure of the **ValidateCert** command. A value of 1 indicates success, and a value of 17 indicates failure.

As a child of the **Certificate** element, the **Status** element contains a status code which indicates the results of the validation of the specific certificate. The following table lists the status codes that apply to certificate validation for the **ValidateCert** command (section 2.2.2.20).

Value	Meaning	Cause	Scope	Resolution
1	Success.	Server successfully completed command.	Global	None.
2	Protocol error.	Supplied protocol parameters are out of range or invalid.	Global	Fix client code.

Value	Meaning	Cause	Scope	Resolution
3	The signature in the digital ID cannot be validated.	The signature in the certificate is invalid.	Item	Verify that the certificate has a valid signature.
4	The digital ID was issued by an untrusted source.	The certificate source is not trusted by the server.	Item	Contact the administrator to add the certificate to the trusted sources list if it is required.
5	The certificate chain that contains the digital ID was not created correctly.	Invalid, incorrectly formatted certificate.	Item	Verify that the certificate chain is formatted correctly.
6	The digital ID is not valid for signing email messages.	The supplied certificate is not meant to be used for signing email.	Item	Prompt the user.
7	The digital ID used to sign the message has expired or is not yet valid.	The certificate has expired.	Item	Obtain a new certificate.
8	The time periods during which the digital IDs in the certificate chain are valid are not consistent.	One or more certificates in the chain could be out of date.	Item	Get the most recent certificate chain for the certificate.
9	A digital ID in the certificate chain is used incorrectly.	The supplied certificate is not valid for what it is being used for.	Item	Obtain a new certificate.
10	Information associated with the digital ID is missing or incorrect.	The certificate format is incorrect.	Item	Obtain a new certificate.
11	A digital ID in the certificate chain is used incorrectly.	A certificate that can only be used as an end-entity is being used as a certification authority (CA) (1), or a CA that can only be used as an end-entity is being used as a certificate.	Item	Obtain the correct certificate chain.
12	The digital ID does not match the recipient's email address.	Incorrect certificate was supplied, could be malicious.	Item	Obtain the correct certificate for the user.
13	The digital ID used to sign this message has been revoked. This can indicate that the issuer of the digital ID no longer trusts the sender, the digital ID was reported stolen, or the digital ID was compromised.	The certificate has been revoked by the certification authority (CA) (1) that issued it.	Item	Obtain a new certificate.
14	The validity of the digital ID cannot be determined because the server that provides this information cannot be	The certificate revocation server is offline.	Item	Retry request after some time.

Value	Meaning	Cause	Scope	Resolution
	contacted.			
15	A digital ID in the chain has been revoked by the authority that issued it.	A certificate in the chain has been revoked.	Item	Obtain a new certificate.
16	The digital ID cannot be validated because its revocation status cannot be determined.	The signature in the certificate is invalid.	Item	Verify that the certificate has a valid signature.
17	An unknown server error has occurred.	The certificate source is not trusted by the server.	Item	Contact the administrator to add the certificate to the trusted sources list if it is necessary.

2.2.3.163   Store

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

▪ **ItemOperations** command requests (section 2.2.2.8)

▪ **Search** command requests and responses (section 2.2.2.14)

The definition of the **Store** element differs according to the context in which it is used. For more details, see section 2.2.3.163.1 and section 2.2.3.163.2.

2.2.3.163.1   Store (ItemOperations)

The **Store** element is a required child element of the **Fetch** element in **ItemOperations** command requests that specifies the name of the store to which the parent operation applies.

All elements referenced in this section are defined in the **ItemOperations** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ItemOperations command request (section 2.2.2.8) fetch operation	**Fetch** (section 2.2.3.63.1)	None	**string** ([MS-ASDTYPE] section 2.6)	1...1 (required)

The **Store** element value can be up to 256 characters in length. The following values are valid for the **Store** element:

▪ Document Library (Microsoft SharePoint Server and UNC links)

▪ Mailbox (items and attachments)

2.2.3.163.2   Store (Search)

The **Store** element is a required child element of the **Search** element in **Search** command requests that contains elements that specify the location, string, and options for the search. The **Store** element is a required child element of the **Response** element in **Search** command responses that contains the **Status**, **Result**, **Range**, and **Total** elements for the returned mailbox entries.

All elements referenced in this section are defined in the **Search** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command request (section 2.2.2.14)	**Search** (section 2.2.3.146)	**Name** (section 2.2.3.110.2) **Query** (section 2.2.3.129) **Options** (section 2.2.3.115.4)	**container** ([MS-ASDTYPE] section 2.2)	1...1 (required)
Search command response	**Response** (section 2.2.3.140.6)	**Status** (section 2.2.3.162.12) **Result** (section 2.2.3.142.2) **Range** (section 2.2.3.130.2) **Total** (section 2.2.3.169.2)	**container**	1...1 (required)

The result of including more than one **Store** element in a **Search** command request is undefined. The server MAY return a protocol status error in response to such a command request.

2.2.3.164 Supported

The **Supported** element is an optional child element of the **Collection** element, as specified in section 2.2.3.29.2, in **Sync** command requests (section 2.2.2.19) that specifies which contact and calendar elements in a **Sync** request are managed by the client and therefore not ghosted. Elements that are not named by the **Supported** element are said to be ghosted.

Elements referenced in this section are defined in the **AirSync** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
Sync command request	**Collection** (section 2.2.3.29.2)	See the description following this table.	**container** ([MS-ASDTYPE] section 2.2)	0...1 (optional)

The status of properties that can be ghosted is determined by the client's usage of the **Supported** element in the initial **Sync** command request for the containing folder, according to the following rules:

1. If the client does not include a **Supported** element in the initial **Sync** command request for a folder, then all of the elements that can be ghosted are considered not ghosted.

2. If the client includes a **Supported** element that contains child elements in the initial **Sync** command request for a folder, then each child element of that **Supported** element is considered not ghosted. All elements that can be ghosted that are not included as child elements of the **Supported** element are considered ghosted.

3. If the client includes an empty **Supported** element in the initial **Sync** command request for a folder, then all elements that can be ghosted are considered ghosted.

When an existing item is modified via the **Change** element (section 2.2.3.24) in a **Sync** command request, the result of omitting an element that can be ghosted changes depending on the status of the element. If the element is ghosted, any existing value for that element is preserved. If the element is not ghosted, any existing value for that element is deleted.

The initial **Sync** request for a folder MUST include a **CollectionId** element (section 2.2.3.30.5), which MUST always precede the optional **Supported** element. For details about the order of elements within the **Collection** node, see section 2.2.3.29.2. This order is strictly enforced. A **Status** element (section 2.2.3.162.16) value of 4 is returned in the **Sync** response if the **CollectionId** element is not included in the **Sync** request. A **Status** element value of 4 is also returned if the order of elements is incorrect within the **Collection** node.

If the client uses the **Supported** element, it MUST be sent in a **Sync** command request when the **SyncKey** element value (section 2.2.3.166.4) is set to 0 (zero). The server caches the list of child elements of the **Supported** element for subsequent synchronizations. If the **Supported** element is included when the **SyncKey** element value is not set to 0 (zero), no error is returned, but the server ignores the request.

All of the **Contact** class properties specified in [MS-ASCNTC] can be included as child elements of the **Supported** element.

To support elements of the **Calendar** class [MS-ASCAL], the following required elements MUST be included as child elements of the **Supported** element:

- **calendar:DtStamp** ([MS-ASCAL] section 2.2.2.16)

- **calendar:Categories** ([MS-ASCAL] section 2.2.2.10)

- **calendar:Sensitivity** ([MS-ASCAL] section 2.2.2.39)

- **calendar:BusyStatus** ([MS-ASCAL] section 2.2.2.8)

- **calendar:UID** ([MS-ASCAL] section 2.2.2.44)

- **calendar:Timezone** ([MS-ASCAL] section 2.2.2.42)

- **calendar:StartTime** ([MS-ASCAL] section 2.2.2.40)

- **calendar:Subject** ([MS-ASCAL] section 2.2.2.41)

- **calendar:Location** ([MS-ASCAL] section 2.2.2.25)

- **calendar:EndTime** ([MS-ASCAL] section 2.2.2.18)

- **calendar:Recurrence** ([MS-ASCAL] section 2.2.2.35)

- **calendar:AllDayEvent** ([MS-ASCAL] section 2.2.2.1)

- **calendar:Reminder** ([MS-ASCAL] section 2.2.2.36)

- **calendar:Exceptions** ([MS-ASCAL] section 2.2.2.20)

The following elements are optional child elements of the **Supported** element when synchronizing the **Calendar** class.

- **calendar:Attendees** ([MS-ASCAL] section 2.2.2.4)

- **calendar:OrganizerName** ([MS-ASCAL] section 2.2.2.34)

- **calendar:OrganizerEmail** ([MS-ASCAL] section 2.2.2.33)

- **calendar:MeetingStatus** ([MS-ASCAL] section 2.2.2.26)

- **calendar:ResponseRequested** ([MS-ASCAL] section 2.2.2.37)<76>

- **calendar:DisallowNewTimeProposal** ([MS-ASCAL] section 2.2.2.15)<77>

For more information on which properties are ghosted by default, consult the ActiveSync class protocol specifications, [MS-ASCAL] and [MS-ASCNTC].

2.2.3.165   Sync

The **Sync** element is a required element in **Sync** command requests and **r**esponses that identifies the body of the HTTP **POST** as containing a **Sync** command (section 2.2.2.19). It is the top-level element in the XML stream.

All elements referenced in this section are defined in the **AirSync** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Sync command request	None	**Collections** (section 2.2.3.31.2) **Partial** (section 2.2.3.121) **Wait** (section 2.2.3.182) **HeartbeatInterval** (section 2.2.3.79.2) **WindowSize** (section 2.2.3.183)	**container** ([MS-ASDTYPE] section 2.2)	1...1 (required)
Sync command response	None	**Collections** (section 2.2.3.31.2) **Limit** (section 2.2.3.88) **Status** (section 2.2.3.162.16)	**container**	1...1 (required)

The **Limit** element and **Collections** element are mutually exclusive in a **Sync** command response. That is, a **Sync** command response can include either a **Limit** element or a **Collections** element, but not both.

2.2.3.166   SyncKey

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

- **FolderCreate** command requests and responses (section 2.2.2.2)

- **FolderDelete** command requests and responses (section 2.2.2.3)

- **FolderUpdate** command requests and responses (section 2.2.2.5)

- **FolderSync** command requests and responses (section 2.2.2.4)

- **GetItemEstimate** command requests (section 2.2.2.7)

- **Sync** command requests and responses (section 2.2.2.19)

The definition of the **SyncKey** element differs according to the context in which it is used. For more details, see section 2.2.3.166.1, section 2.2.3.166.2, section 2.2.3.166.3, section 2.2.3.166.4.

2.2.3.166.1 SyncKey (FolderCreate, FolderDelete, and FolderUpdate)

The **SyncKey** element is a used in **FolderCreate** command requests and responses, **FolderDelete** command and responses, and **FolderUpdate** command and responses to represent the synchronization state of a collection.

All elements referenced in this section are defined in the **FolderHierarchy** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
FolderCreate command request (section 2.2.2.2)	**FolderCreate** (section 2.2.3.67)	None	**string** ([MS-ASDTYPE] section 2.6)	1…1 (required)
FolderCreate command response	**FolderCreate**	None	**string**	0…1 (optional)
FolderDelete command request (section 2.2.2.3)	**FolderDelete** (section 2.2.3.68)	None	**string**	1…1 (required)
FolderDelete command response	**FolderDelete**	None	**string**	0…1 (optional)
FolderUpdate command request (section 2.2.2.5)	**FolderUpdate** (section 2.2.3.72)	None	**string**	1…1 (required)
FolderUpdate command response	**FolderUpdate**	None	**string**	0…1 (optional)

The server returns a **Status** element (section 2.2.3.162.5) value of 10 if the **SyncKey** element is not included in a **FolderUpdate** command request.

After a successful **FolderCreate** command (section 2.2.2.2), **FolderDelete** command (section 2.2.2.3), or **FolderUpdate** command (section 2.2.2.5), the server MUST send a synchronization key to the client in a response. If the **FolderCreate** command, **FolderDelete** command, or **FolderUpdate** command is not successful, the server MUST NOT return a **SyncKey** element.

The client MUST store this key and send it back to the server the next time the folder hierarchy is synchronized or updated. The server MUST check the value of the key to make sure the value of the **SyncKey** element provided in the request matches a synchronization key value on the server. The server MUST return a **Status** element (section 2.2.3.162.4) value of 9 if the value of the **SyncKey** element does not match the value of the synchronization key on the server.

The client MUST store the synchronization key as an opaque string of up to 64 characters.

2.2.3.166.2 SyncKey (FolderSync)

The **SyncKey** element is a required child element of the **FolderSync** element in **FolderSync** command requests and an optional child element of the **FolderSync** element in **FolderSync** command responses. In both contexts, it is used by the server to track the current state of the client.

All elements referenced in this section are defined in the **FolderHierarchy** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
FolderSync command request (section 2.2.2.4)	**FolderSync** (section 2.2.3.71)	None	**string** ([MS-ASDTYPE] section 2.6)	1…1 (required)
FolderSync command response	**FolderSync**	None	**string**	0…1 (optional)

After successful folder synchronization, the server MUST send a synchronization key to the client. The client MUST store this key and send the key back to the server the next time the folder hierarchy is synchronized or updated. The server MUST check the value of the key to make sure the value of the **SyncKey** element provided in the request matches a synchronization key value on the server. The server MUST return a **Status** element (section 2.2.3.162.4) value of 9 if the value of the **SyncKey** element does not match the value of the synchronization key on the server.

The client MUST store the synchronization key as an opaque string of up to 64 characters.

2.2.3.166.3 SyncKey (GetItemEstimate)

The **airsync:SyncKey** element is a required child element of the **Collection** element in **GetItemEstimate** command requests that represents the current state of a collection.

Elements referenced in this section are defined in the **GetItemEstimate** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
GetItemEstimate command request (section 2.2.2.7)	**Collection** (section 2.2.3.29.1)	None	**string** ([MS-ASDTYPE] section 2.6)	1…1 (required)

The value of the element is examined by the server to determine the state of the synchronization process. The **airsync:SyncKey** element is the first child element of the **Collection** element (section 2.2.3.29.1) in a **GetItemEstimate** command request.<78>

The **airsync:SyncKey** element value used within the **GetItemEstimate** requests is the same as the one returned within the **Sync** responses. The server does not update the **airsync:SyncKey** value on **GetItemEstimate** requests. For more details about the **airsync:SyncKey** element, see section 2.2.3.166.4. The server checks the value of the key to verify that the value of the **airsync:SyncKey** element provided in the request matches a **airsync:SyncKey** value on the server. The server MUST provide a **Status** element (section 2.2.3.162.6) value of 4 if the **airsync:SyncKey** value provided in the **GetItemEstimate** request does not match those expected within the next **Sync** command request (section 2.2.2.19).

The **airsync:SyncKey** element value is a **string** value up to 64 characters in length.

2.2.3.166.4 SyncKey (Sync)

The **SyncKey** element is a required child element of the **Collection** element in **Sync** command requests and responses that contains a value that is used by the server to mark the synchronization state of a collection.

All elements referenced in this section are defined in the **AirSync** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Sync command request (section 2.2.2.19)	**Collection** (section 2.2.3.29.2)	None	**string** ([MS-ASDTYPE] section 2.6)	1…1 (required)
Sync command response	**Collection** (section 2.2.3.29.2)	None	**string**	1…1 (required)

A synchronization key of value 0 (zero) initializes the synchronization state on the server and causes a full synchronization of the collection. The server sends a response that includes a new synchronization key value. The client MUST store this synchronization key value until the client requires the key value for the next synchronization request for that collection. When the client uses this synchronization key value to do the next synchronization of the collection, the client sends this synchronization key value to the server in a **Sync** request. If the synchronization is successful, the server responds by sending all objects in the collection. The response includes a new synchronization key value that the client MUST use on the next synchronization of the collection.

The client MUST store the synchronization key as an opaque string of up to 64 characters.

The client MUST send a synchronization key value of 0 in an initial **Sync** request and the server sends a new synchronization key value in its response to the client. The client MUST NOT ignore the synchronization key value that is included in the initial response from the server.

2.2.3.167 Title

The **Title** element in the **GAL** namespace is an optional child element of the **search:Properties** element in **Search** command responses. The **Title** element is only present in response to **Search** command requests that contain a **search:Name** element with a value of "GAL", as specified in section 2.2.3.110.2. It contains the title of a recipient in the GAL that matched the search criteria from the corresponding **Search** command request.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command response (section 2.2.2.14)	**Properties** (section 2.2.3.128.2)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

2.2.3.168 To

The **To** element is a required child element of the **ResolveRecipients** element in **ResolveRecipients** command requests that specifies one or more recipients to be resolved. The **To** element is a required child element of the **Response** element in **ResolveRecipients** command responses that specifies a recipient to be resolved. The **To** element is an ANR search field. Its value can be up to 256 characters in length.

All elements referenced in this section are defined in the **ResolveRecipients** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ResolveRecipients command request (section 2.2.2.13)	ResolveRecipients (section 2.2.3.139)	None	string ([MS-ASDTYPE] section 2.6)	1…1000 (required)
ResolveRecipients command response	Response (section 2.2.3.140.5)	None	string	1…1 (required)

The result of including more than 1000 **To** elements in the request is undefined. The server MAY return a protocol status error in response to such a command request.

If the **To** element specifies an ambiguous name and the **Availability** element (section 2.2.3.16) is included in the request, the response will not include free/busy data for that user. The **Availability** element is only included when the **To** element includes a valid SMTP address or name that resolves to a unique individual on the server.

The server attempts to match the **To** element value specified in the request to common directory service user attributes, and then return the matches. The **To** element(s) that are returned in the response corresponds directly to the **To** element(s) that are specified in the request.<79>

If the **ResolveRecipients** command request includes the **Availability** element, a maximum of 100 **To** elements containing SMTP addresses can be included in the request. If more than 100 SMTP addresses are included in the request, **Status** element (section 2.2.3.162.11) value 160 is returned in the response.

If the **ResolveRecipients** command request includes the **Availability** element and includes a **To** element for an ambiguous user, the response does not include a **MergedFreeBusy** element (section 2.2.3.97) for that user. Only users or distribution lists specified with valid SMTP addresses or a uniquely identifiable string in the request message **To** element have **MergedFreeBusy** elements included in the response.

If the **ResolveRecipients** command request includes the **Availability** element and the **To** element specifies a distribution group, then the availability data is returned as a single string that merges the data for the individual members of the distribution group. If the distribution group contains more than 20 members, a **Status** element value of 161 is returned in the response indicating that the merged free busy information of such a large distribution group is not useful. For more information about the **Status** element, see section 2.2.3.162.11.

2.2.3.169 Total

The **Total** element is used in the following command responses:

- **ItemOperations** command responses (section 2.2.2.8)

- **Search** command responses (section 2.2.2.14)

The definition of the **Total** element differs according to the context in which it is used. For more details, see section 2.2.3.169.1 and section 2.2.3.169.2.

2.2.3.169.1 Total (ItemOperations)

The **Total** element is an optional child element of the **Properties** element in **ItemOperations** command responses that indicates the total size of an item on the server, in bytes.

All elements referenced in this section are defined in the **ItemOperations** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ItemOperations command response (section 2.2.2.8) fetch operation	**Properties** (section 2.2.3.128.1)	None	**integer** ([MS-ASDTYPE] section 2.5)	0…1 (optional)

2.2.3.169.2 Total (Search)

The **Total** element is a required child element of the **Store** element in **Search** command responses that provides an estimate of the total number of mailbox entries that matched the search **Query** element (section 2.2.3.129) value.

All elements referenced in this section are defined in the **Search** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command response (section 2.2.2.14)	**Store** (section 2.2.3.163.2)	None	**integer** ([MS-ASDTYPE] section 2.5)	1…1 (required)

The value of the **Total** element does not always equal the number of entries that are returned. To determine the number of entries that are returned by the **Search** command, use the **Range** element (section 2.2.3.130.2) value.

The **Total** element indicates the number of entries that are available. In cases where all the results are returned in the response XML, the value of the **Total** element is one more than the end-index value that is provided in the **Range** element. For example, if the **Search** command returns 15 entries, the value of the **Range** element is 0–14, while the value of the **Total** element is 15.

The **Total** element is used by clients to determine whether more matching entries were found in the mailbox than have been returned by the **Search** command. For example, a client might perform an initial search and specify a requested **Range** of 0–4 (return 5 entries maximum). If the **Total** element indicates that there are actually 25 matching items, the client can then enable the user to retrieve the full results.

2.2.3.170 Type

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

- **Autodiscover** command responses (section 2.2.2.1)

- **FolderCreate** command requests (section 2.2.2.2)

- **FolderSync** command requests (section 2.2.2.4)

- **ResolveRecipients** command responses (section 2.2.2.13)

The definition of the **Type** element differs according to the context in which it is used. For more details, see section 2.2.3.170.1, section 2.2.3.170.2, section 2.2.3.170.3, and section 2.2.3.170.4.

2.2.3.170.1   Type (Autodiscover)

The **Type** element is an optional child element of the **Server** element in **Autodiscover** command responses that specifies the server type.

All elements referenced in this section are defined in the http://schemas.microsoft.com/exchange/autodiscover/mobilesync/requestschema/2006 namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Autodiscover command response (section 2.2.2.1)	**Server** (section 2.2.3.149)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

The following are the valid values for the **Type** element:

- MobileSync—Indicates that the URL that is returned by the **URL** element (section 2.2.3.172) can be accessed by clients.

- CertEnroll—Indicates that the URL that is returned by the **URL** element can be accessed by clients to obtain a client certificate for **Secure Sockets Layer (SSL)** negotiation.

If the server supports both "MobileSync" and "CertEnroll", the response buffer includes multiple **Server** elements (section 2.2.3.149) that contain a URL value for each **Type** element value. The presence of a **Server** element with a child **Type** element with a value of "CertEnroll" is optional.

2.2.3.170.2   Type (FolderCreate)

The **Type** element is a required child element of the **FolderCreate** element in **FolderCreate** command requests that specifies the type of the folder to be created.

All elements referenced in this section are defined in the **FolderHierarchy** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
FolderCreate command request (section 2.2.2.2)	**FolderCreate** (section 2.2.3.67)	None	**unsignedByte** ([MS-ASDTYPE] section 2.7)	1…1 (required)

The folder type values are listed in the following table. Folder types 2–11 and 19 are reserved for default folder types.

Value	Meaning
1	User-created folder (generic)
12	User-created mail folder
13	User-created Calendar folder
14	User-created Contacts folder
15	User-created Tasks folder
16	User-created Journal folder

Value	Meaning
17	User-created Notes folder

2.2.3.170.3 Type (FolderSync)

The **Type** element is a required child element of the **Update** element and the **Add** element in **FolderSync** command responses that specifies the type of the folder that was updated (renamed or moved) or added on the server.

All elements referenced in this section are defined in the **FolderHierarchy** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
FolderSync command response (section 2.2.2.4)	**Update** (section 2.2.3.171)	None	**integer** ([MS-ASDTYPE] section 2.5)	1…1 (required)
FolderSync command response	**Add** (section 2.2.3.7.1)	None	**integer**	1…1 (required)

Each **Update** element and each **Add** element included in a **FolderSync** response MUST contain one **Type** element.

The folder type values are listed in the following table.

Value	Meaning
1	User-created folder (generic)
2	Default Inbox folder
3	Default Drafts folder
4	Default Deleted Items folder
5	Default Sent Items folder
6	Default Outbox folder
7	Default Tasks folder
8	Default Calendar folder
9	Default Contacts folder
10	Default Notes folder
11	Default Journal folder
12	User-created Mail folder
13	User-created Calendar folder
14	User-created Contacts folder
15	User-created Tasks folder

Value	Meaning
16	User-created journal folder
17	User-created Notes folder
18	Unknown folder type
19<80>	Recipient information cache

2.2.3.170.4 Type (ResolveRecipients)

The **Type** element is a required child element of the **Recipient** element in **ResolveRecipients** command responses that indicates the type of recipient, either a contact entry (2) or a GAL entry (1).

All elements referenced in this section are defined in the **ResolveRecipients** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ResolveRecipients command response (section 2.2.2.13)	**Recipient** (section 2.2.3.132)	None	**unsignedByte** ([MS-ASDTYPE] section 2.7)	1...1 (required)

2.2.3.171 Update

The **Update** element is an optional child element of the **Changes** element in **FolderSync** command responses that identifies a folder on the server that has been updated (renamed or moved).

All elements referenced in this section are defined in the **FolderHierarchy** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
FolderSync command response (section 2.2.2.4)	**Changes** (section 2.2.3.25)	**ServerId** (section 2.2.3.151.3) **ParentId** (section 2.2.3.119.2) **DisplayName** (section 2.2.3.47.3) **Type** (section 2.2.3.170.3)	**container** ([MS-ASDTYPE] section 2.2)	0...N (optional)

The child elements of the **Update** element identify the server ID of the folder that was updated, the server ID of its parent folder, the new display name of the updated folder, and the folder type.

2.2.3.172 Url

The **Url** element is an optional child element of the **Server** element in **Autodiscover** command responses that specifies a URL string that conveys the protocol, port, resource location, and other information.

All elements referenced in this section are defined in the http://schemas.microsoft.com/exchange/autodiscover/mobilesync/requestschema/2006 namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Autodiscover command response (section 2.2.2.1)	**Server** (section 2.2.3.149)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

2.2.3.173 User

The **User** element is a required child element of the **Response** element in **Autodiscover** command responses that encapsulates information about the user to whom the **Response** element relates.

All elements referenced in this section are defined in the http://schemas.microsoft.com/exchange/autodiscover/mobilesync/requestschema/2006 namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Autodiscover command response (section 2.2.2.1)	**Response** (section 2.2.3.140.1)	**DisplayName** (section 2.2.3.47.1) **EMailAddress** (section 2.2.3.52)	**container** ([MS-ASDTYPE] section 2.2)	1…1 (required)

2.2.3.174 UserAgent

The **UserAgent** element is an optional child element of the **Set** element in **Settings** command requests that specifies the user agent. The **UserAgent** element value can be up to 1024 characters in length.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command **DeviceInformation** request (section 2.2.2.16)	**Set** (section 2.2.3.152)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

The **UserAgent** element SHOULD contain the information in the User-Agent header. The User-Agent header SHOULD be removed from the HTTP request. The server does not validate the value of the **UserAgent** element.

2.2.3.175 UserDisplayName

The **UserDisplayName** element<81> is an optional child element of the **Account** element in **Settings** command responses (section 2.2.2.16). The value of this element specifies the display name of the user associated with the given account. The **UserDisplayName** element value can be up to 512 characters in length. The server MAY return this element in **Settings** command responses.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command **UserInformation** response	**Account** (section 2.2.3.2)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

2.2.3.176 UserInformation

The **UserInformation** element is an optional child element of the **Settings** element in **Settings** command requests that serves as a container node that is used to request a list of a user's email addresses from the server. The **UserInformation** element is an optional child element of the **Settings** element in **Settings** command responses that serves as a container node that is used to return the status of the request and a list of a user's email addresses.

All elements referenced in this section are defined in the **Settings** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Settings command request (section 2.2.2.16)	**Settings** (section 2.2.3.153.2)	**Get** (section 2.2.3.75)	**container** ([MS-ASDTYPE] section 2.2)	0…1 (optional)
Settings command response	**Settings**	**Get** **Status** (section 2.2.3.162.14)	container	0…1 (optional)

The list of a user's email addresses can be useful, for example, for ensuring that the user is not included when performing a Reply to All operation to an email message.

In a request, the **UserInformation** element contains the **Get** element to indicate that the server is to return all available email addresses for the user.

The **Settings** command (section 2.2.2.16) supports read-only access to the list of a user's various email addresses via the **Get** element. The client is unable to write this information.

2.2.3.177 UserName

The **UserName** element is used in the following command requests:

- **ItemOperations** command requests (section 2.2.2.8)

- **Search** command requests (section 2.2.2.14)

The definition of the **UserName** element differs according to the context in which it is used. For more details, see section 2.2.3.177.1 and section 2.2.3.177.2.

2.2.3.177.1 UserName (ItemOperations)

The **UserName** element is an optional child element of the **Options** element in **ItemOperations** command requests that specifies the username of the account leveraged to fetch the desired item. The **Password** element (section 2.2.3.122.1) contains the corresponding account password. The **UserName** element value can be up to 100 characters in length.

All elements referenced in this section are defined in the **ItemOperations** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ItemOperations command request (section 2.2.2.8) fetch operation	**Options** (section 2.2.3.115.2)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

2.2.3.177.2   UserName (Search)

The **UserName** element is an optional child element of the **Options** element in **Search** command requests that specifies the user account used to search the document from the document library. The **UserName** element value can be up to 100 characters in length.

All elements referenced in this section are defined in the **Search** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command request (section 2.2.2.14)	**Options** (section 2.2.3.115.4)	None	**string** ([MS-ASDTYPE] section 2.6)	0…1 (optional)

2.2.3.178   UserResponse

The **UserResponse** element is a required child element of the **Request** element in **MeetingResponse** command requests that indicates whether the meeting is being accepted, tentatively accepted, or declined.

All elements referenced in this section are defined in the **MeetingResponse** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
MeetingResponse command request (section 2.2.2.9)	**Request** (section 2.2.3.137.2)	None	**integer** ([MS-ASDTYPE] section 2.5)	1…1 (required)

The following table shows valid values for the element.

Value	Meaning
1	Accepted
2	Tentatively accepted
3	Declined

2.2.3.179   ValidateCert

The **ValidateCert** element is a required element in **ValidateCert** command requests and responses that identifies the body of the HTTP **POST** as containing a **ValidateCert** command (section 2.2.2.20). It is the top-level element in the XML stream.

All elements referenced in this section are defined in the **ValidateCert** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ValidateCert command request	None	**CertificateChain** (section 2.2.3.20) **Certificates** (section 2.2.3.23.2) **CheckCRL** (section 2.2.3.26)	**container** ([MS-ASDTYPE] section 2.2)	1…1 (required)
ValidateCert command response	None	**Status** (section 2.2.3.162.17) **Certificate** (section 2.2.3.19.2)	**container**	1…1 (required)

2.2.3.180 Value

The **Value** element is a required child element of the **GreaterThan** element, the **LessThan** element, and the **EqualTo** element in **Search** command requests. In each context, it specifies the value that is to be used in a comparison. The **Value** element value can be up to 1024 bytes in length.

All elements referenced in this section are defined in the **Search** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Search command request (section 2.2.2.14)	**GreaterThan** (section 2.2.3.78) **LessThan** (section 2.2.3.87) **EqualTo** (section 2.2.3.59)	None	**string** ([MS-ASDTYPE] section 2.6)	1…1 (required)

The **Value** element is used in the query together with an element that specifies the name of a property. The value that is specified by the **Value** element is compared with the value of the specified property.

2.2.3.181 Version

The **Version** element is an optional child element of the **Properties** element in **ItemOperations** command responses that indicates the time at which a document item was last modified.

All elements referenced in this section are defined in the **ItemOperations** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
ItemOperations command response (section 2.2.2.8) fetch operation	**Properties** (section 2.2.3.128.1)	None	**datetime** ([MS-ASDTYPE] section 2.3)	0…1 (optional)

The **Version** element is present only when the **ItemOperations** command is used to access a Windows SharePoint Services or UNC resource.

2.2.3.182   Wait

The **Wait** element is an optional child element of the **Sync** element in **Sync** command requests that specifies the number of minutes that the server SHOULD delay a response if no new items are added to the included folders, as specified in section 3.1.5.4.

All elements referenced in this section are defined in the **AirSync** namespace.

Command request/response	Parent elements	Child elements	Data type	Number allowed
Sync command request (section 2.2.2.19)	**Sync** (section 2.2.3.165)	None	**integer** ([MS-ASDTYPE] section 2.5)	0...1 (optional)

Valid values for the **Wait** element are 1 through 59. When the client requests a wait interval that is outside the acceptable range, the server will send a response that includes a **Status** element (section 2.2.3.162.16) value of 14 and a **Limit** element (section 2.2.3.88).

Either the **Wait** element or the **HeartbeatInterval** element (section 2.2.3.79.2) can be included in a **Sync** command request, but not both. If both elements are included, the server response will contain a **Status** element value of 4.

2.2.3.183   WindowSize

The **WindowSize** element is an optional child element of the **Collection** element and the **Sync** element in **Sync** command requests that specifies a maximum number of changed items in a collection or a request that SHOULD be included in the synchronization response.

Elements referenced in this section are defined in the **AirSync** namespace, except where otherwise indicated by the presence of a namespace prefix (as defined in section 2.2.1).

Command request/response	Parent elements	Child elements	Data type	Number allowed
Sync command request (section 2.2.2.19)	**Collection** (section 2.2.3.29.2) **Sync** (section 2.2.3.165)	None	**integer** ([MS-ASDTYPE] section 2.5)	0...1 (optional)

The maximum value for the **WindowSize** element is 512. However, if the **WindowSize** element is set to 512, the server can send **Sync** response messages that contain less than 512 updates. If the server does not send all the updates in a single message, the **Sync** response message contains the **MoreAvailable** element (section 2.2.3.106), which indicates that there are additional updates on the server to be downloaded to the client.

The **WindowSize** element appears only in requests that are sent to the server from the client. If the **WindowSize** element is omitted, the server behaves as if a **WindowSize** element with a value of 100 were submitted. The server interprets the value 0 (zero) and values above 512 as 512.

WindowSize element values less than 100 increase the load on the server, increase bandwidth, and decrease battery life because of the additional requests that are required to obtain all changes. **WindowSize** element values larger than 100 cause larger responses, which are more susceptible to communication errors. A **WindowSize** element value less than 100 can be useful if the client can display the initial set of objects while additional ones are still being retrieved from the server.

If the number of changes on the server is greater than the value of the **WindowSize** element, the server returns a **MoreAvailable** element in the response. If this occurs, the client MUST synchronize again to continue getting items from the server.

The **WindowSize** element has been repurposed to also impose a global limit on the number of changes that are returned by the server. The **WindowSize** element value can still be specified at the collection level and the server MUST honor both the global and collection level settings.

Collections are to be processed by the server in the order received, as follows:

- If the server has filled the **WindowSize** on a particular collection that has more changes, it will return the **MoreAvailable** element for that collection and continue to process the other collections until the global **WindowSize** has been filled.

- When the server has filled the global **WindowSize** and collections that have changes did not fit in the response, the server can return a **MoreAvailable** element.

- If a collection is not present in a **Sync** command response, the client can assume that no changes are currently available for that collection.

The actual number of changes that are included in a **Sync** command response for any particular collection depends on the value of the **WindowSize** element of the collection, the overall number of changes that are already included in the response, and the value of the global **WindowSize** element. The server will stop processing after the global **WindowSize** has been filled and simply not process the remaining collections. Any server-side changes that are pending in the unprocessed collections are picked up in the next synchronization.

2.2.4   Common Status Codes

The status codes returned for each command are specified in the **Status** element (section 2.2.3.162) section corresponding to each command. Links to the **Status** element sections are listed in the following table.

Command	Status element section
Autodiscover (section 2.2.2.1)	Section 2.2.3.162.1
FolderCreate (section 2.2.2.2)	Section 2.2.3.162.2
FolderDelete (section 2.2.2.3)	Section 2.2.3.162.3
FolderSync (section 2.2.2.4)	Section 2.2.3.162.4
FolderUpdate (section 2.2.2.5)	Section 2.2.3.162.5
GetItemEstimate (section 2.2.2.7)	Section 2.2.3.162.6
ItemOperations (section 2.2.2.8)	Section 2.2.3.162.7
MeetingResponse (section 2.2.2.9)	Section 2.2.3.162.8
MoveItems (section 2.2.2.10)	Section 2.2.3.162.9
Ping (section 2.2.2.11)	Section 2.2.3.162.10
Provision (section 2.2.2.12)	[MS-ASPROV] section 3.1.5.2
ResolveRecipients (section 2.2.2.13)	Section 2.2.3.162.11

Command	Status element section
Search (section 2.2.2.14)	Section 2.2.3.162.12
SendMail (section 2.2.2.15)	Section 2.2.3.162.13
Settings (section 2.2.2.16)	Section 2.2.3.162.14
SmartForward (section 2.2.2.17)	Section 2.2.3.162.15
SmartReply (section 2.2.2.18)	Section 2.2.3.162.15
Sync (section 2.2.2.19)	Section 2.2.3.162.16
ValidateCert (section 2.2.2.20)	Section 2.2.3.162.17

Many of the status codes listed in the **Status** element sections have a scope assigned to them. The following table defines the scope values.

Value	Meaning
Global	The status pertains to the overall client request.
Item	The status pertains to a particular item within the overall client request.
Policy	The status pertains to a particular policy within the **Provision** command.

In addition to the status values specified for individual commands in the **Status** element sections of this document, the following status values are common to all commands.

Value	Element name	Meaning
101	InvalidContent	The body of the HTTP request sent by the client is invalid.<82> Ensure the HTTP request is using the specified Content-Type and length, and that the request is not missing (when an empty body is not allowed). Examples: **Ping** command (section 2.2.2.11) with a text/plain body, or **SendMail** command (section 2.2.2.15) with version 12.1 and a WBXML body.
102	InvalidWBXML	The request contains WBXML but it could not be decoded into XML.
103	InvalidXML	The XML provided in the request does not follow the protocol requirements.
104	InvalidDateTime	The request contains a timestamp that could not be parsed into a valid date and time.
105	InvalidCombinationOfIDs	The request contains a combination of parameters that is invalid.
106	InvalidIDs	The request contains one or more IDs

Value	Element name	Meaning
		that could not be parsed into valid values. That is different from specifying an ID in the proper format that does not resolve to an existing item.<83>
107	InvalidMIME	The request contains MIME that could not be parsed.
108	DeviceIdMissingOrInvalid	The device ID is either missing or has an invalid format.
109	DeviceTypeMissingOrInvalid	The device type is either missing or has an invalid format.
110	ServerError	The server encountered an unknown error, the device SHOULD NOT retry later.<84>
111	ServerErrorRetryLater	The server encountered an unknown error, the device SHOULD retry later.<85>
112	ActiveDirectoryAccessDenied	The server does not have access to read/write to an object in the directory service.<86>
113	MailboxQuotaExceeded	The mailbox has reached its size quota.<87>
114	MailboxServerOffline	The mailbox server is offline.
115	SendQuotaExceeded	The request would exceed the send quota.
116	MessageRecipientUnresolved	One of the recipients could not be resolved to an email address.
117	MessageReplyNotAllowed	The mailbox server will not allow a reply of this message.
118	Message PreviouslySent	The message was already sent in a previous request. The server determined this by remembering the **ClientId** (section 2.2.3.28) values of the last few sent messages. This request contains a **ClientId** that was already used in a recent message.
119	MessageHasNoRecipient	The message being sent contains no recipient.
120	MailSubmissionFailed	The server failed to submit the message for delivery.
121	MessageReplyFailed	The server failed to create a reply message.
122	AttachmentIsTooLarge	The attachment is too large to be

Value	Element name	Meaning
		processed by this request.
123	UserHasNoMailbox	A mailbox could not be found for the user.
124	UserCannotBeAnonymous	The request was sent without credentials. Anonymous requests are not allowed.
125	UserPrincipalCouldNotBeFound	The user was not found in the directory service.
126	UserDisabledForSync	The user object in the directory service indicates that this user is not allowed to use ActiveSync.
127	UserOnNewMailboxCannotSync	The server is configured to prevent users from syncing.
128	UserOnLegacyMailboxCannotSync	The server is configured to prevent users on legacy servers from syncing.
129	DeviceIsBlockedForThisUser	The user is configured to allow only some devices to sync. This device is not the allowed device.
130	AccessDenied	The user is not allowed to perform that request.
131	AccountDisabled	The user's account is disabled.
132	SyncStateNotFound	The server's data file that contains the state of the client was unexpectedly missing. It might have disappeared while the request was in progress. The next request will likely answer a sync key error and the device will be forced to do full sync.<88>
133	SyncStateLocked	The server's data file that contains the state of the client is locked, possibly because the mailbox is being moved or was recently moved.
134	SyncStateCorrupt	The server's data file that contains the state of the client appears to be corrupt.
135	SyncStateAlreadyExists	The server's data file that contains the state of the client already exists. This can happen with two initial syncs are executed concurrently.
136	SyncStateVersionInvalid	The version of the server's data file that contains the state of the client is invalid.
137	CommandNotSupported	The command is not supported by this server.<89>

Value	Element name	Meaning
138	VersionNotSupported	The command is not supported in the protocol version specified.<90>
139	DeviceNotFullyProvisionable	The device uses a protocol version that cannot send all the policy settings the admin enabled. For more details, see [MS-ASPROV] section 3.1.5.2.
140	RemoteWipeRequested	A remote wipe was requested. The device SHOULD provision to get the request and then do another provision to acknowledge it.<91> For more details, see [MS-ASPROV] section 3.1.5.1.
141	LegacyDeviceOnStrictPolicy	A policy is in place but the device is not provisionable. For more details, see [MS-ASPROV] section 3.1.5.2.
142	DeviceNotProvisioned	There is a policy in place; the device needs to provision as specified in [MS-ASPROV] section 3.1.5.1.<92>
143	PolicyRefresh	The policy is configured to be refreshed every few hours. The device needs to re-provision as specified in [MS-ASPROV] section 3.1.5.1.
144	InvalidPolicyKey	The device's policy key is invalid. The policy has probably changed on the server. The device needs to re-provision as specified in [MS-ASPROV] section 3.1.5.1.
145	ExternallyManagedDevicesNotAllowed	The device claimed to be externally managed, but the server doesn't allow externally managed devices to sync. For more details, see [MS-ASPROV] section 3.1.5.2.
146	NoRecurrenceInCalendar	The request tried to forward an occurrence of a meeting that has no recurrence.
147	UnexpectedItemClass<93>	The request tried to operate on a type of items unknown to the server.
148	RemoteServerHasNoSSL	The request needs to be proxied to another server but that server doesn't have Secure Sockets Layer (SSL) enabled. This server is configured to only proxy requests to servers with SSL enabled.
149	InvalidStoredRequest	The server had stored the previous request from that device. When the device sent an empty request, the server tried to re-execute that previous request but it was found to be

Value	Element name	Meaning
		impossible. The device needs to send the full request again.
150	ItemNotFound	The **ItemId** (section 2.2.3.84) value specified in the **SmartReply** command (section 2.2.2.18) or **SmartForward** command (section 2.2.2.17) request could not be found in the mailbox.
151	TooManyFolders	The mailbox contains too many folders. By default, the mailbox cannot contain more than 1000 folders.
152	NoFoldersFound	The mailbox contains no folders.
153	ItemsLostAfterMove	After moving items to the destination folder, some of those items could not be found.
154	FailureInMoveOperation	The mailbox server returned an unknown error while moving items.
155	MoveCommandDisallowedForNonPersistentMoveAction	An **ItemOperations** command request (section 2.2.2.8) to move a conversation is missing the **MoveAlways** element (section 2.2.3.108).
156	MoveCommandInvalidDestinationFolder	The destination folder for the move is invalid.
160	AvailabilityTooManyRecipients	The command has reached the maximum number of recipients that it can request availability for.<94>
161	AvailabilityDLLimitReached	The size of the distribution list is larger than the availability service is configured to process.<95>
162	AvailabilityTransientFailure	Availability service request failed with a transient error.<96>
163	AvailabilityFailure	Availability service request failed with an error.<97>
164	BodyPartPreferenceTypeNotSupported	The **BodyPartPreference** node (as specified in [MS-ASAIRS] section 2.2.2.7) has an unsupported **Type** element (as specified in [MS-ASAIRS] section 2.2.2.22.4) value.<98>
165	DeviceInformationRequired	The required **DeviceInformation** element (as specified in [MS-ASPROV] section 2.2.2.52) is missing in the **Provision** request.<99>
166	InvalidAccountId	The **AccountId** (section 2.2.3.3) value is not valid.<100>

Value	Element name	Meaning
167	AccountSendDisabled	The **AccountId** value specified in the request does not support sending email.<101>
168	IRM_FeatureDisabled	The Information Rights Management feature is disabled.<102>
169	IRM_TransientError	Information Rights Management encountered an error.<103>
170	IRM_PermanentError	Information Rights Management encountered an error.<104>
171	IRM_InvalidTemplateID	The Template ID value is not valid.<105>
172	IRM_OperationNotPermitted	Information Rights Management does not support the specified operation.<106>
173	NoPicture	The user does not have a contact photo.<107>
174	PictureTooLarge	The contact photo exceeds the size limit set by the **MaxSize** element (section 2.2.3.95).<108>
175	PictureLimitReached	The number of contact photos returned exceeds the size limit set by the **MaxPictures** element (section 2.2.3.94).<109>
176	BodyPart_ConversationTooLarge	The conversation is too large to compute the body parts. Try requesting the body of the item again, without body parts.<110>
177	MaximumDevicesReached	The user's account has too many device partnerships. Delete partnerships on the server before proceeding.<111>

3   Protocol Details

3.1   Common Details

3.1.1   Abstract Data Model

None.

3.1.2   Timers

None.

3.1.3   Initialization

None.

3.1.4   Higher-Layer Triggered Events

None.

3.1.5   Message Processing Events and Sequencing Rules

The client creates request messages consisting of an HTTP header, as specified in [MS-ASHTTP], and the XML command to be performed on the server, as specified in [MS-ASHTTP] section 2.2.1. The request message is sent to the server by the client and a response message is received back from the server.

3.1.5.1   Downloading Policy Settings

This section specifies how the client downloads policy settings from the server by using the **Provision** command (section 2.2.2.12).
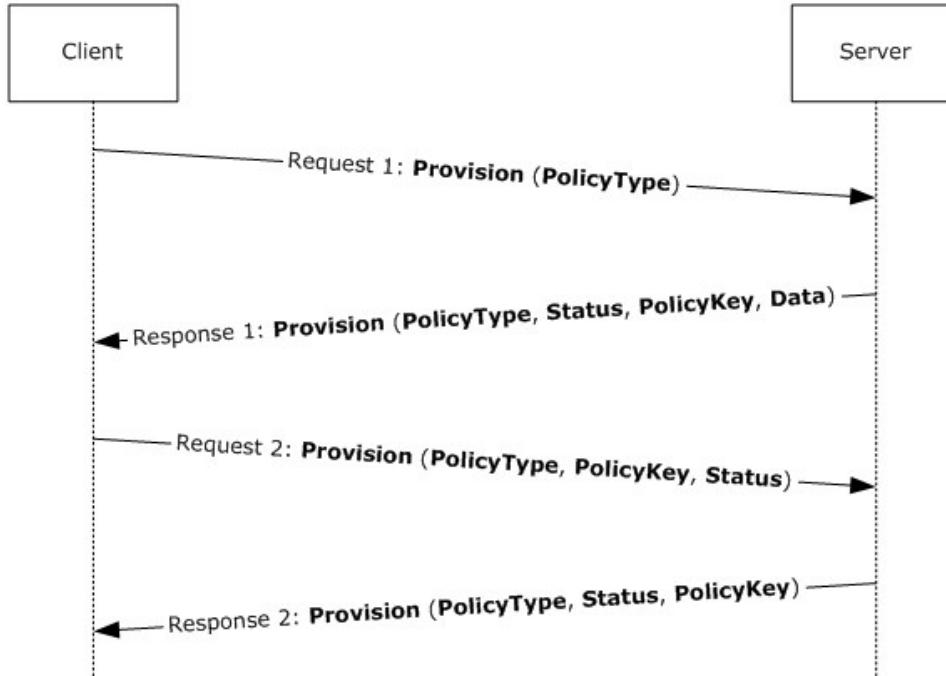
The first command the client issues to the server SHOULD<112> be a **Provision** command, as specified in [MS-ASPROV]. The client can send the HTTP **OPTIONS** command to the server before sending the **Provision** command, to retrieve server settings, but the HTTP **OPTIONS** command is optional. If the client sends other commands to the server before sending the **Provision** command, the server will return a **Status** element (as specified in [MS-ASPROV] section 2.2.2.53) value of 142 to the client.

The initial **Provision** command request MUST contain the **provision:PolicyType** element (as specified in [MS-ASPROV] section 2.2.2.42), which specifies the format in which the policy settings are provided. If the **provision:PolicyType** element is not included in the initial **Provision** command request, the server responds with a **provision:Status** element value of 2. The server then responds with the **provision:PolicyType**, **provision:PolicyKey** (as specified in [MS-ASPROV] section 2.2.2.41), and **provision:Data** ([MS-ASPROV] section 2.2.2.23) elements. The **provision:PolicyKey** element is used by the server to mark the state of policy settings on the client device. The policy settings, in the format specified in the **provision:PolicyType** element, are contained in the **provision:Data** element.

The client device then applies the policy settings that were received from the server and sends an acknowledgement back to the server in another **Provision** command request. The acknowledgement from the client device contains **provision:PolicyType**, **provision:PolicyKey**, and **provision:Status** elements. The **provision:Status** element indicates whether the policy

settings were successfully applied by the client. The response from the server contains **provision:PolicyType**, **provision:PolicyKey**, and **provision:Status** elements. The **provision:Status** element indicates whether the server successfully recorded the client's acknowledgement.

The following figure shows the process for downloading policy settings.

Figure 3: Downloading policy settings

The following table lists the command sequence for downloading policy settings.

Order	Client action	Server action
1	The client sends a **Provision** command request with the type of policy settings to be downloaded.	The server responds with the policy type, policy key, data, and status code.
2	The client acknowledges that it received and applied the policy settings by sending another **Provision** command request with the policy type, policy key, and status code.	The server responds with the policy type, policy key, and status code to indicate that the server recorded the client's acknowledgement.

3.1.5.2 Setting Device Information

This section specifies how the client uses the **Settings** command (section 2.2.2.16) to set device information on the server.

Clients SHOULD<113> send **settings:DeviceInformation** (section 2.2.3.45) parameters to the server as soon as possible after the client has been provisioned, and before the **FolderSync** command (section 2.2.2.4), so that the server can use this information to determine what the device has access to.<114>

The client sets device information by sending an initial **Settings** command request to the server with the **settings:Set** element (section 2.2.3.152) identifying **settings:DeviceInformation** parameters as specified in section 2.2.3.45.

3.1.5.3 Synchronizing a Folder Hierarchy

This section specifies how the client uses the **FolderSync** command (section 2.2.2.4) to replicate the folder hierarchy of the user's mailbox on the client.

The client initiates folder synchronization by sending an initial **FolderSync** command request to the server with a **folderhierarchy:SyncKey** element (section 2.2.3.166.2) value of zero (0). The server responds with a new **folderhierarchy:SyncKey** element value and provides a list of all the folders in the user's mailbox. The folders are identified by a **folderhierarchy:ServerId** element (section 2.2.3.151.3), which can then be used in a **Sync** command (section 2.2.2.19) to synchronize the items in those folders.

Additional folder synchronizations can be performed by using the **folderhierarchy:SyncKey** element value from the initial **FolderSync** command response to get folder additions, deletions, or updates from the server. At any point, the client can repeat the initial **FolderSync** command, sending a **SyncKey** element value of zero (0), and resynchronizing the entire hierarchy. Existing **folderhierarchy:ServerId** values do not change when the client resynchronizes.

The client can use the **GetItemEstimate** command (section 2.2.2.7) to obtain an estimate of the number of items that need to be synchronized in a collection, which is useful when the client UI displays a progress bar while it retrieves items from the server. The client can also limit the number of changed items returned in the **Sync** response by submitting the **airsync:WindowSize** element (section 2.2.3.183), which specifies the maximum number of items to synchronize at one time. If the number of items returned is larger than the value specified by the **airsync:WindowSize** element, the **airsync:MoreAvailable** element (section 2.2.3.106) is returned in the **Sync** command response. The client then continues to call the **Sync** command (section 2.2.2.19) until no more items are available.

The following table lists the command sequence for folder hierarchy synchronization.

The asterisk (*) in the Order column means that a step is run once and can be repeated multiple times.

Order	Client action	Server action
1	The client sends the **FolderSync** command with the **folderhierarchy:SyncKey** element set to zero (0) to get the folder hierarchy and the **folderhierarchy:ServerId** values of all the folders.	The server responds with the folder hierarchy and a new **folderhierarchy:SyncKey** value. The client stores the names and **folderhierarchy:ServerId** values of all folders that can be synchronized.
2*	The client sends the **FolderSync** command with the new **folderhierarchy:SyncKey** value to update the folder hierarchy.	If any changes have occurred on the server, the new, deleted, or changed folders are returned to the client.
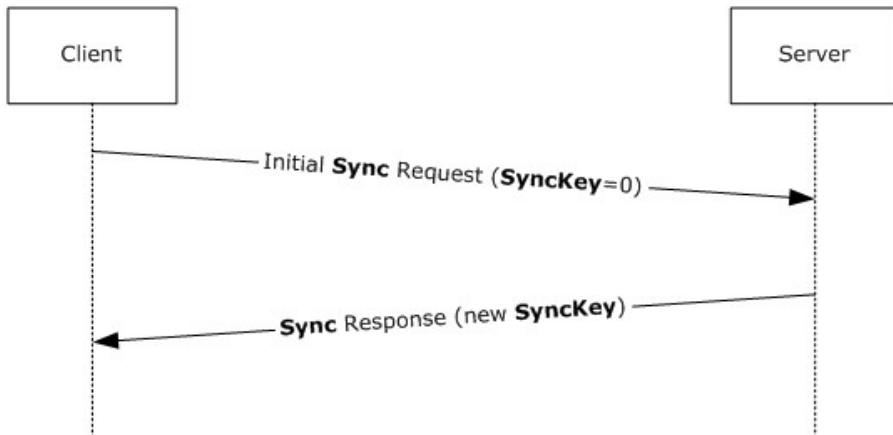
The folder hierarchy is now populated on the client and ready for the contents of the folders to be synchronized.

If the **FolderSync** response contains a **Status** element value of 9 (see section 2.2.3.162.4), the client's local copy of the folder hierarchy list can no longer be considered valid. The client SHOULD<115> restart the synchronization process with a synchronization key of 0. Any changes that were sent in the **FolderSync** request were not applied to the server.

3.1.5.4 Synchronizing Inbox, Calendar, Contacts, and Tasks Folders
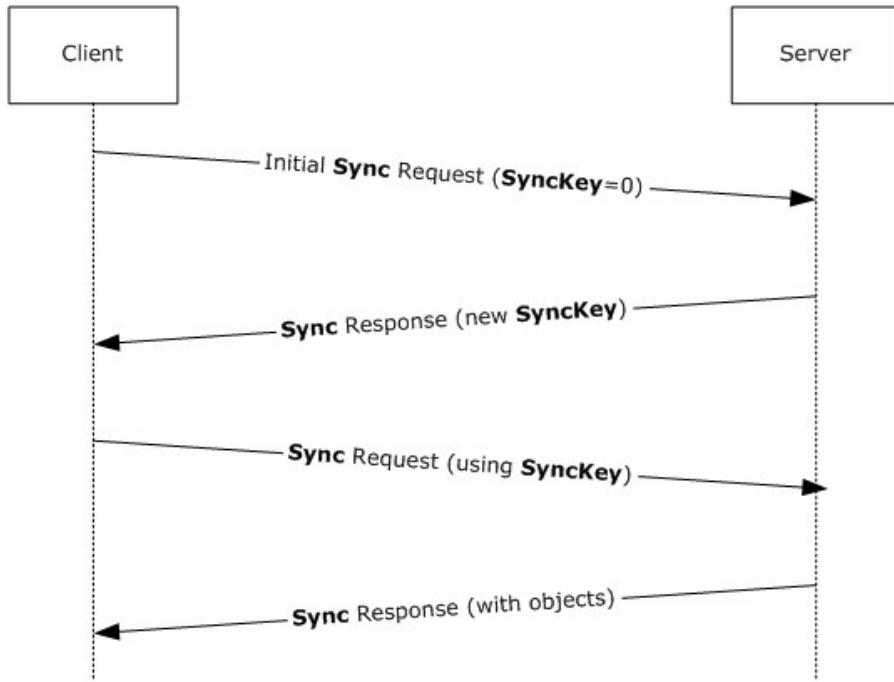
The client synchronizes the contents of individual folders by using the **Sync** command (section 2.2.2.19). The client can synchronize the Inbox folder, Calendar folder, or Contacts folder, or any folder within the mailbox after the folder hierarchy has been populated by the **FolderSync** command (section 2.2.2.4), as specified in section 3.1.5.3. Clients SHOULD NOT synchronize the Drafts folder.

In order to synchronize the content of each of the folders, an initial synchronization key for each folder MUST be obtained from the server. The client obtains the key by sending the server an initial **Sync** request with a **airsync:SyncKey** element (section 2.2.3.166.4) value of zero (0) and the **airsync:CollectionId** element (section 2.2.3.30.5) value that identifies the folder to be synchronized. The **Sync** command response includes a new **airsync:SyncKey** value, which is generated by the server for each transaction.

Figure 4: Retrieval of SyncKey value

The **airsync:SyncKey** issued in the initial **Sync** response MUST be stored by the client, and is sent in the second **Sync** request. The second **Sync** request includes the new **airsync:SyncKey** element as well as the **airsync:GetChanges** (section 2.2.3.76) element.

Figure 5: Retrieval of folder content

The server responds by adding all the items in the collection to the client and returning a new **airsync:SyncKey** element value, which can be used in successive synchronizations. The order of items in the response is implementation-specific. Clients SHOULD NOT rely on any specific ordering of items in the response. The client deletes its copy of all objects in the collection that are being synchronized before the client performs a full synchronization. The client can use the **GetItemEstimate** command (section 2.2.2.7) to obtain an estimate of the number of items that have to be synchronized before completely synchronizing a collection, which is useful when the client user interface displays a progress bar while getting items from the server. In some cases, the client could have to submit a **airsync:WindowSize** element (section 2.2.3.183) that specifies the number of items to be synchronized at a time.

If more items remain to be synchronized, the **airsync:MoreAvailable** element (section 2.2.3.106) is returned in the **Sync** command response. The client then continues to call the **Sync** command until no more items are available. For more details about the **airsync:WindowSize** element of the **Sync** command, see section 2.2.3.183.

After a full synchronization has been performed on a collection, successive synchronizations are used to obtain additions, deletions, or changes to the initial collection state. The client can use the **Sync** command request to add, delete, or change items on the server, and the server can use the **Sync** command response to add, delete, or change items on the client.

The following table lists the command sequence for folder synchronization.

The asterisk (*) in the Order column means that the step can be repeated multiple times. [n] means that a step is optional.

Order	Client action	Server action
1	The client sends the **Sync** command for the	The server responds with the synchronization key

Order	Client action	Server action
	Email, Calendar, Contacts, and/or Tasks collection with a synchronization key of zero (0). This establishes a partnership with the server, initializing server data for the device.	for the collection, to be used in successive synchronizations.
2*	The client sends the **Sync** command with a synchronization key of zero (0) for other collections to be synchronized.	The server responds with new synchronization keys for each collection.
[3]	The client sends the **GetItemEstimate** command for all collections to be synchronized. This step can be skipped if it is not required by the client UI.	The server responds to indicate how many items will be added, changed, or deleted, for each collection.
4*	The client sends the **Sync** command with the **airsync:GetChanges** element for a collection. The command SHOULD include the **airsync:WindowSize** element, the recommended value for which is 100. This step is repeated for each collection to be synchronized or all collections can be combined into one request.	The server responds with **airsync:Add** (section 2.2.3.7.2), **airsync:Change** (section 2.2.3.24), or **airsync:Delete** (section 2.2.3.42.2) elements for items in the collection. If the response contains the **airsync:MoreAvailable** element, this step is repeated.

The client can use the **airsync:WindowSize** element to break the server **airsync:Add** elements into multiple sets of items. The recommended window size is 100. For more details about the **airsync:WindowSize** element used by the **Sync** command, see section 2.2.3.183.

When the **Sync** command request contains a **HeartbeatInterval** element (section 2.2.3.79.2) or a **Wait** element (section 2.2.3.182), the server SHOULD delay sending a response until either of the following conditions are met:

- The interval specified by the **HeartbeatInterval** element or **Wait** element has elapsed.

- An item is added to one or more of the enumerated collections. This condition includes new mail being delivered and items being copied or moved into the collection.

In most cases, the server SHOULD NOT send the response before the interval specified by the **HeartbeatInterval** element or **Wait** element for any other changes to the collection, including items being modified, items being deleted, items being moved out of the collection, or items being soft deleted due to falling out of scope of a synchronization filter, as specified in section 2.2.3.64.2. However, if the **Sync** command request includes an **Add** or **Change** element, and the addition or change does not involve only flags, the server SHOULD return immediately, regardless of the current **HeartbeatInterval** or **Wait** value.

3.1.5.5 Receiving and Accepting Meeting Requests

This section specifies how the client retrieves items from the Inbox folder by using the **Sync** command (section 2.2.2.19), responds to a meeting request item by using the **MeetingResponse** command (section 2.2.2.9), and synchronizes the Calendar folder by using the **Sync** command so that the new **Calendar object** is added to the client's calendar.

A meeting request is returned by the server in response to a synchronization of the Inbox folder. A meeting request is an email message that has an embedded calendar item. The message contains an **email:MessageClass** element (as specified in [MS-ASEMAIL] section 2.2.2.41) that has a value of "IPM.Schedule.Meeting.Request", and its **airsync:ApplicationData** element (section 2.2.3.11)

contains an **email:MeetingRequest** element (as specified in [MS-ASEMAIL] section 2.2.2.40). If the meeting request is a delegated meeting request, the server SHOULD<116> return a substitute meeting invitation email message instead of a meeting request, as specified in section 3.1.5.5.1. Servers MAY<117> return the actual meeting request with the **email2:MeetingMessageType** element set to 6, as specified in [MS-ASEMAIL] section 2.2.2.39. In this case, clients MUST NOT attempt to respond to the meeting request with the **MeetingResponse** command.

When the client displays the meeting request message, if the value of the **email2:MeetingMessageType** element is 1 or 2, the client SHOULD offer the options of accepting, declining, or tentatively accepting the meeting. If one of these actions is selected, the client sends a **MeetingResponse** command to the server. For other values of the **email2:MeetingMessageType** element, the client SHOULD NOT send a **MeetingResponse** command to the server.

If the response to the meeting is accepted or is tentatively accepted, the server will add or update the corresponding calendar item and return its server ID in the **meetingresponse:CalendarId** element (section 2.2.3.18) of the response. If the response to the meeting is declined, the response will not contain a **meetingresponse:CalendarId** element because the server will delete the corresponding calendar item. If the client had created a tentative meeting calendar item, the client updates that item with the returned server ID (if accepted or tentative). The client MUST also change the busy status on the client calendar item from tentative to busy if the meeting request was accepted. Note that, if the client synchronizes the Calendar folder after responding to a meeting request, the calendar item in question will be in conflict if the client also sends the changed item change for it back to the server. This conflict is resolved according to the conflict resolution rules that are specified by the client in the **Sync** command request.

If the meeting request was accepted, the Calendar folder MUST be synchronized for the client to obtain the new calendar item. The new calendar item for the accepted meeting is added here and MUST be added to the client's calendar.

The following table lists the command sequence for receiving and accepting meeting requests. The asterisk (*) in the Order column means that a step can be repeated multiple times.

Order	Client action	Server action
1	The client sends the **Sync** command request for the Inbox collection with the value of the **airsync:SyncKey** element (section 2.2.3.166.4) set to zero (0).	The server responds with the **airsync:SyncKey** for the collection, to be used in successive synchronizations.
2*	The client sends a **Sync** command request, specifying the **airsync:GetChanges** element (section 2.2.3.76) and the **airsync:SyncKey** element for the Inbox folder. The command SHOULD include the **airsync:WindowSize** element (section 2.2.3.183), the recommended value for which is 100.	The server responds with **airsync:Add** elements (section 2.2.3.7.2) for items in the Inbox collection, including a meeting request item. If the response contains the **airsync:MoreAvailable** element (section 2.2.3.106), this step is repeated.
3	The user chooses to accept, decline, or tentatively accept a meeting request that is displayed in the client UI.	
4	The client sends a **MeetingResponse** command request (section 2.2.2.9) to the server, which specifies that the meeting was accepted, declined, or tentatively accepted, and provides the server IDs of the meeting request message and its parent folder.	The server sends a response that contains the **MeetingResponse** command request status along with the ID of the calendar item that corresponds to this meeting request if the meeting was not declined.

Order	Client action	Server action
5	If a response was requested by the organizer, the client SHOULD use a **SendMail** command (section 2.2.2.15) to send an appropriately formatted meeting response.	If the message was sent successfully, the server returns an empty response. Otherwise, the server responds with a **Status** element (section 2.2.3.162.8) that indicates the type of failure.
6	If the meeting was not declined, the client sends a **Sync** command for the calendar collection, specifying the **GetChanges** element.	The server responds with any changes to the Calendar folder caused by the last synchronization and the new calendar item for the accepted meeting.

3.1.5.5.1   Substitute Meeting Invitation Email

To send a substitute meeting invitation email message, the server creates a temporary item. The server MUST NOT modify or remove the original meeting request in order to create the temporary item.

The substitute meeting invitation email message elements are set according to the following table.

Substitute meeting invitation email message element	Value
email:To	Set to the email address of the delegate.
email:Cc	Blank.
email:From	Copy value from original meeting request.
email:Subject	Original subject prepended with explanatory text.
email:DateReceived	Copy value from original meeting request.
email:DisplayTo	Copy value from original meeting request.
email:ThreadTopic	Copy value from original meeting request.
email:Importance	Copy value from original meeting request.
email:Read	Copy value from original meeting request.
airsyncbase:Body	Summary of meeting details.
email:MessageClass	Set to "IPM.Note".
email:InternetCPID	Copy value from original meeting request.
email:Flag	Copy value from original meeting request.
email:ContentClass	Set to "urn:content-classes:message".
airsyncbase:NativeBodyType	Copy value from original meeting request.
email2:ConversationId	Copy value from original meeting request.
email2:ConversationIndex	Copy value from original meeting request.

Substitute meeting invitation email message element	Value
email:Categories	Copy value from original meeting request.

The exact format of the explanatory text for the **email:Subject** element is implementation-specific but SHOULD indicate that the email message represents a meeting request for the delegator.

The exact format of the **airsyncbase:Body** element is implementation-specific but SHOULD include summary details of the original meeting request, such as date, time, and location.

Actions taken by the client on the substitute meeting invitation email message will actually take effect on the original meeting request on the server (for example, flagging the substitute meeting invitation email message for follow-up will result in the original meeting request being flagged for follow-up on the server), with the following exceptions:

- Replying to the substitute meeting invitation email message with the **SendMail** command will reply to the substitute meeting invitation email message. The server will append the original meeting request to the reply.

- Forwarding the substitute meeting invitation email message with the **SendMail** command will forward the substitute meeting invitation email message. The server will append the original meeting request to the forwarded message.

- Any attachments on the original meeting request are not exposed via the substitute meeting invitation email message.

For an example of a substitute meeting invitation email message, see section 4.5.11.

3.1.5.6   Monitoring Folders for New Items

The client uses the **Ping** command (as specified in section 2.2.2.11) to monitor a folder or set of folders for changes. The client MUST include a **HeartbeatInterval** element (as specified in section 2.2.3.79.1) in the initial **Ping** command request to specify an amount of time that the server SHOULD wait before sending a response if there are no new items added to the included folders. The server MUST cache the value of the **HeartbeatInterval** element and use it in subsequent **Ping** requests that do not include a **HeartbeatInterval** element.

The client MUST include a **Folders** element (as specified in section 2.2.3.70) in the initial **Ping** command request to specify one or more folders that the server will monitor for additions. The server MUST cache the list of included folders and use it in subsequent **Ping** requests that do not include a **Folders** element.

Because the server caches the values of the **HeartbeatInterval** element and the **Folders** element, the client can omit either element, or both from subsequent **Ping** requests if those parameters have not changed from the previous **Ping** request. If neither the heartbeat interval nor the folder list has changed, the client can issue an empty **Ping** request – one with no XML body. The server will use the previously cached XML sent by the client if it receives an empty **Ping** request

The server uses the last **airsync:SyncKey** element (as specified in section 2.2.3.166.4) value returned to the client when determining what to report in the **Ping** response. Therefore, the client MUST have received the response to its last **Sync** request (as specified in section 2.2.2.19) and successfully applied the changes sent by the server, prior to issuing a **Ping** request.

The server SHOULD delay sending a response until either of the following conditions are met:

- The interval specified by the **HeartbeatInterval** element has elapsed.

- An item is added to one or more of enumerated collections. This condition includes new mail being delivered and items being copied or moved into the collection.

The server SHOULD NOT send the response before the interval specified by the **HeartbeatInterval** element for any other changes to a collection, including items being modified, items being deleted, or items being moved out of the collection.

In the case of no changes on the server, the client can then reissue a new **Ping** request. In the case of changes, the response indicates in which folders those changes occurred so that the client can resynchronize those folders.

Note that if no changes occur in any of the folders that are specified by the client for a significant length of time (longer than the value of the **HeartbeatInterval** element (section 2.2.3.79.1)), the client can run in a loop in which it issues a **Ping** request, receives a response indicating that there are no changes, and then reissues the **Ping** request. The length of time that the server waits before issuing a response is called the heartbeat interval.

3.1.5.7 Handling Status Errors

The client MUST handle errors that occur during synchronization sessions. Errors fall into two categories: HTTP errors and ActiveSync protocol errors. HTTP errors are standard error codes, such as "401 Logon failed", and they are returned from the server in response to an HTTP **POST**. ActiveSync protocol errors result from a problem on the server, in an attempt to perform the task requested by the command request message. ActiveSync protocol errors are indicated by codes that are returned in the **Status** element (section 2.2.3.162) of a command response. For more details about the status codes, see section 2.2.4.

The client MUST implement error handling and a user interface (UI). Some errors are handled by a recovery procedure. Other errors require that an error message be displayed, along with a prompt for the user to respond. The client determines whether to run a recovery procedure or prompt for user input.

In addition to the ActiveSync protocol errors that the server sends, incomplete communication between server and client can result in the failure of a synchronization session. The server has an error recovery feature that enables a client to respond to errors by repeating the most recent synchronization session. The client MUST handle synchronization failures by retrying the synchronization. The server tracks synchronization requests to be able to respond appropriately in both of the following cases:

- The client failed in communicating a full request to the server for synchronization.

  - In this case, the client sends a request but the server does not receive the request. The server does not act on the request, and no server-side changes occur. Therefore, no response is sent to the client. The client MUST resend a synchronization request if there is no immediate server response and neither the **airsync:Wait** element (section 2.2.3.182) nor the **airsync:HeartbeatInterval** element (section 2.2.3.79.2) was sent in the **Sync** request (section 2.2.2.19), or if the **airsync:Wait** element value or **airsync:HeartbeatInterval** element value was specified in the **Sync** request and the time has elapsed.

- The server failed in communicating a response to the client for updates.

  - In this case, the server response is not received by the client. The client knows it has not received a response if neither the **airsync:Wait** element nor **airsync:HeartbeatInterval** element was sent in the **Sync** request and the server response is not received immediately, or

if the **airsync:Wait** element value or the **airsync:HeartbeatInterval** element value was specified in the **Sync** request and that time has elapsed. The data on the server changed. The client MUST resend the request. The server recognizes the duplicate request. Because the server changes have already occurred, the server resends the response to the client to keep the server and client synchronized.

3.1.5.8 Limiting Size of Command Requests

Clients SHOULD limit the number of elements in command requests as specified in the following table. Servers SHOULD<118> impose the limits specified in the table and return the specified error if the limit is exceeded. Servers MAY<119> use different values for the limits specified in the table, so long as they conform to the specified minimum and maximum values.

Command request	Element to limit	Limit value	Error returned
GetItemEstimate (section 2.2.2.7)	**Collection** (section 2.2.3.29.1)	1000 (minimum 30, maximum 5000)	**Status** element (section 2.2.3.162.6) value of 103
MeetingResponse (section 2.2.2.9)	**Request** (section 2.2.3.137.2)	100 (minimum 1, maximum 2,147,483,647)	**Status** element (section 2.2.3.162.8) value of 103
MoveItems (section 2.2.2.10)	**Move** (section 2.2.3.107.2)	1000 (minimum 1, maximum 2,147,483,647)	**Status** element (section 2.2.3.162.9) value of 103
ResolveRecipients (section 2.2.2.13)	**To** (section 2.2.3.168)	100 (minimum 1, maximum 2,147,483,647)	**Status** element (section 2.2.3.162.11) value of 5
Sync (section 2.2.2.19)	The sum of the number of **Add** (section 2.2.3.7.2), **Change** (section 2.2.3.24), **Delete** (section 2.2.3.42.2), and **Fetch** (section 2.2.3.63.2) elements	200 (minimum 1, maximum 2,147,483,647)	**Status** element (section 2.2.3.162.16) value of 4
ValidateCert (section 2.2.2.20)	**Certificate** (section 2.2.3.19.2)	100 (minimum 1, maximum 2,147,483,647)	**Status** element (section 2.2.3.162.17) value of 2

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

4   Protocol Examples

4.1   Downloading the Current Server Security Policy by Using the Provision Command

For examples on downloading the current server security policy by using the **Provision** command (section 2.2.2.12), see [MS-ASPROV] section 4.

4.2   Discovering Account Settings by Using the Autodiscover Command

The **Autodiscover** command (section 2.2.2.1) enables clients to discover core account configuration information by using the user's SMTP address as the primary input by means of the following process:

1. The end-user enters his or her email address and domain credentials, for example, kim@contoso.com.

2. The client uses the domain information in the user's email address, that is, contoso.com, and tries to locate the Autodiscover service by sending an **Autodiscover** command request to the following predefined URLs:

    ▪ https://*<SMTP-address-domain>*/autodiscover/autodiscover.xml (In this example, this URL is: https://contoso/autodiscover/autodiscover.xml.)

    ▪ https://autodiscover.*<SMTP-address-domain>*/autodiscover/autodiscover.xml (In this example, this URL is: https://autodiscover.contoso/autodiscover/autodiscover.xml.)

3. If the **Domain Name System (DNS)** contains a host record that maps one of these URLs to a server where the Autodiscover service is hosted, then the Autodiscover service responds with the settings that are required for the device to begin synchronizing. This response includes values for the server type, the URL, and the **Name** element (section 2.2.3.110.1).

4. If redirection to another Autodiscover service is required, then the **Redirect** element (section 2.2.3.134) is present and contains a URL to the **Autodiscover server** to query for the desired information.

5. The device then re-creates a partnership with the new server, and send an **Autodiscover** command request to that server.

6. If the response included the settings that are required for the device to begin synchronization, then the device applies the settings to initiate synchronization.

7. If the **Autodiscover** command request in step 3 fails, then the device performs a DNS SRV lookup for _autodiscover._tcp.*<smtp-address-domain>*.com, which in this example maps to _autodiscover._tcp.contoso.com. If the DNS lookup is successful, then "mail.*<smtp-address-domain>*.com" is returned, which maps to "mail.contoso.com". The device then applies the settings to initiate synchronization. For more information about performing the DNS SRV lookup, see [MSFT-DNS-SRV].

The following sections show success and error response messages.

Account autodiscovery uses an email address to look up information that is required to configure software. Given an email name (such as chris@woodgrovebank.com), a list of possible Autodiscover servers is generated. The client contacts the name Autodiscover.*domainname* to provide the information. If that information is not found, the client tries to send the request to the domain

name. If the information still is not retrieved, the client can use a manual configuration. For example, the client tries these servers:

- autodiscover.woodgrovebank.com

- woodgrovebank.com

The client sends each server an HTTP **POST** command. The post data is an XML request for a certain type of information. Email account configuration is the first use. The XML contains information that helps execute the request. For mail, the information includes the email address, the protocols that the client software supports, the Web browser that is installed, the type of proxy that is being used, and the types of authentication that can be used.

The post is sent for *servername*/autodiscover/autodiscover.xml. The server name is defined according to the process described earlier in this topic.

4.2.1 Request

The following example shows the **Autodiscover** command request (section 2.2.2.1) sent from the client to the server.

```
<?xml version="1.0" encoding="utf-8"?>
<Autodiscover
xmlns="http://schemas.microsoft.com/exchange/autodiscover/mobilesync/requestschema/2006">
    <Request>
        <EMailAddress>chris@woodgrovebank.com</EMailAddress>
        <AcceptableResponseSchema>
        http://schemas.microsoft.com/exchange/autodiscover/mobilesync/
        responseschema/2006
        </AcceptableResponseSchema>
    </Request>
</Autodiscover>
```

4.2.2 Response - Case Error

The following example shows an **Autodiscover** command response (section 2.2.2.1) sent from the server to the client when an error occurred. In this example, the error is that the directory service could not be reached.

```
<?xml version="1.0" encoding="utf-8"?>
<Autodiscover
xmlns="http://schemas.microsoft.com/exchange/autodiscover/mobilesync/responseschema/2006">
    <Response>
        <Culture>en:us</Culture>
        <User>
            <EMailAddress>chris@woodgrovebank.com</EMailAddress>
        </User>
        <Action>
            <Error>
                <Status>2</Status>
                <Message>The directory service could not be reached</Message>
                <DebugData>MailUser</DebugData>
            </Error>
        </Action>
    </Response>
```

Release: February 10, 2014

```
        </Autodiscover>
```

4.2.3   Response - Case Redirect

The following example shows an **Autodiscover** command response (section 2.2.2.1) that indicates that a redirection to another Autodiscover service is required. In the following redirect example, assume that the initial **Autodiscover** command request was sent to autodiscover.woodgrovebank.com. The redirect node indicates to the client to try issuing an **Autodiscover** command request to autodiscover.loandept.woodgrovebank.com.

```
<?xml version="1.0" encoding="utf-8"?>
<Autodiscover
xmlns="http://schemas.microsoft.com/exchange/autodiscover/mobilesync/responseschema/2006">
    <Response>
        <Culture>en:us</Culture>
        <User>
           <DisplayName>Chris Gray</DisplayName>
           <EMailAddress>chris@woodgrovebank.com</EMailAddress>
        </User>
        <Action>
           <Redirect>chris@loandept.woodgrovebank.com </Redirect>
        </Action>
    </Response>
</Autodiscover>
```

4.2.4   Response - Case Server Settings

The following example shows an **Autodiscover** command response (section 2.2.2.1) that provides server URL information for two services: MobileSync and CertEnroll. The client can use the MobileSync URL to configure the synchronization settings. The client can also optionally use the CertEnroll information to obtain a client certificate for SSL negotiation.

```
<?xml version="1.0" encoding="utf-8"?>
<Autodiscover
xmlns="http://schemas.microsoft.com/exchange/autodiscover/mobilesync/responseschema/2006">
    <Response>
        <Culture>en:us</Culture>
        <User>
            <DisplayName>Chris Gray</DisplayName>
            <EMailAddress>chris@woodgrovebank.com</EMailAddress>
        </User>
        <Action>
            <Settings>
                <Server>
                    <Type>MobileSync</Type>
                    <Url>
                        https://loandept.woodgrovebank.com/Microsoft-Server-ActiveSync
                    </Url>
                    <Name>
                 https://loandept.woodgrovebank.com/Microsoft-Server-ActiveSync
              </Name>
                </Server>
                <Server>
                    <Type>CertEnroll</Type>
                    <Url>https://cert.woodgrovebank.com/CertEnroll</Url>
```

Release: February 10, 2014

```
                <Name />
                <ServerData>CertEnrollTemplate</ServerData>
            </Server>
        </Settings>
    </Action>
</Response>
</Autodiscover>
```

4.2.5   Response - Case Framework Error

The following example shows an **Autodiscover** command response (section 2.2.2.1) that indicates either the provider cannot be found, or the **autodiscover:AcceptableResponseSchema** (section 2.2.3.1) value cannot be matched.

The error code 600 means an invalid request was sent to the service, and the error code 601 means that a provider could not be found to handle the **autodiscover:AcceptableResponseSchema** value that was indicated.

```
<?xml version="1.0" encoding="utf-8"?>
<Autodiscover
xmlns ="http://schemas.microsoft.com/exchange/autodiscover/mobilesync/responseschema/2006">
    <Response>
        <Error Time="16:56:32.6164027" Id="1054084152">
            <ErrorCode>600</ErrorCode>
            <Message>Invalid Request</Message>
            <DebugData />
        </Error>
    </Response>
</Autodiscover>
```

4.2.6   Response – Case Framework Default

For unauthenticated requests, the server can create and serve a static page with contents to aid in troubleshooting errors, such as the following.

```
<?xml version="1.0" encoding="utf-8"?>
<Autodiscover>
  <Account>
    <AccountType>default email</AccountType>
    <Action>settings</Action>
    <Image>http://www.abcd.com/def.jpg</Image>
    <ServiceHome>http://www.microsoft.com</ServiceHome>
    <RedirectUrl>...</RedirectUrl>

    <Protocol>
      <Type>POP</Type>
      <Server>popserverFQDN</Server>
      <Port>110</Port>
    </Protocol>

    <Protocol>
      <Type>SMTP</Type>
      <Server>smtpserverFQDN</Server>
      <Port>25</Port>
    </Protocol>
```

```
   <Protocol>
     <Type>IMAP</Type>
     <Server>imapserver1FQDN</Server>
   </Protocol>

   <Protocol>
     <Type>IMAP</Type>
     <Server>imapserver2FQDN</Server>
     <Port>143</Port>
   </Protocol>
 </Account>
</Autodiscover>
```

4.3   Setting Device Information by Using the Settings Command

The following sections show how to set device information by using the **Settings** command (section 2.2.2.16).

4.3.1   Request

The following example shows the **Settings** command request (section 2.2.2.16) sent from the client to the server to set device information.

```
<?xml version="1.0" encoding="utf-8"?>
<Settings xmlns="Settings">
    <DeviceInformation>
        <Set>
            <Model>...</Model>
            <IMEI>...</IMEI>
            <FriendlyName>...</FriendlyName>
            <OS>...</OS>
            <OSLanguage>...</OSLanguage>
            <PhoneNumber>...</PhoneNumber>
            <MobileOperator>...</MobileOperator>
            <UserAgent>...</UserAgent>
        </Set>
    </DeviceInformation>
</Settings>
```

4.3.2   Response

The following example shows the **Settings** command response (section 2.2.2.16) sent from the server to the client.

```
<?xml version="1.0" encoding="utf-8"?>
<Settings xmlns="Settings">
    <Status>1</Status>
    <DeviceInformation>
        <Set>
            <Status>...</Status>
        </Set>
    </DeviceInformation>
</Settings>
```

4.4   Synchronizing Folders by Using the FolderSync Command

The following sections show how to synchronize folders by using the **FolderSync** command (section 2.2.2.4).

4.4.1   Request

The following example shows the initial **FolderSync** command request (section 2.2.2.4) sent from the client to the server. The **folderhierarchy:SyncKey** element (section 2.2.3.166.2) in the request is set to 0 (zero), because this is the first **FolderSync** command issued. Any subsequent **FolderSync** requests will contain the **folderhierarchy:SyncKey** value that was returned in the previous **FolderSync** response.

```
<?xml version="1.0" encoding="utf-8"?>
<FolderSync xmlns="FolderHierarchy">
  <SyncKey>0</SyncKey>
</FolderSync>
```

4.4.2   Response

The following example shows the initial **FolderSync** command response (section 2.2.2.4) sent from the server to the client. The **folderhierarchy:SyncKey** element (section 2.2.3.166.2) value has been incremented in the response, and this value will be used in the next **FolderSync** request, **FolderCreate** request (section 2.2.2.2), **FolderDelete** request (section 2.2.2.3), or **FolderUpdate** request (section 2.2.2.5). Each folder in the collection is added to the client using an **folderhierarchy:Add** element (section 2.2.3.7.1), which indicates the server ID of the folder, the parent ID of the folder, a display name, and the type of the folder, which indicates whether it is an email, calendar, task, or other type of folder.

```
<?xml version="1.0" encoding="utf-8"?>
<FolderSync xmlns="FolderHierarchy">
  <Status>1</Status>
  <SyncKey>1</SyncKey>
  <Changes>
    <Count>13</Count>
    <Add>
      <ServerId>1</ServerId>
      <ParentId>0</ParentId>
      <DisplayName>Calendar</DisplayName>
      <Type>8</Type>
    </Add>
    <Add>
      <ServerId>2</ServerId>
      <ParentId>0</ParentId>
      <DisplayName>Contacts</DisplayName>
      <Type>9</Type>
    </Add>
    <Add>
      <ServerId>3</ServerId>
      <ParentId>0</ParentId>
      <DisplayName>Deleted Items</DisplayName>
      <Type>4</Type>
    </Add>
    <Add>
      <ServerId>4</ServerId>
      <ParentId>0</ParentId>
```

Release: February 10, 2014

```
      <DisplayName>Drafts</DisplayName>
      <Type>3</Type>
    </Add>
    <Add>
      <ServerId>5</ServerId>
      <ParentId>0</ParentId>
      <DisplayName>Inbox</DisplayName>
      <Type>2</Type>
    </Add>
    <Add>
      <ServerId>12</ServerId>
      <ParentId>5</ParentId>
      <DisplayName>NewFolder</DisplayName>
      <Type>12</Type>
    </Add>
    <Add>
      <ServerId>6</ServerId>
      <ParentId>0</ParentId>
      <DisplayName>Journal</DisplayName>
      <Type>11</Type>
    </Add>
    <Add>
      <ServerId>7</ServerId>
      <ParentId>0</ParentId>
      <DisplayName>Junk E-Mail</DisplayName>
      <Type>12</Type>
    </Add>
    <Add>
      <ServerId>8</ServerId>
      <ParentId>0</ParentId>
      <DisplayName>Notes</DisplayName>
      <Type>10</Type>
    </Add>
    <Add>
      <ServerId>9</ServerId>
      <ParentId>0</ParentId>
      <DisplayName>Outbox</DisplayName>
      <Type>6</Type>
    </Add>
    <Add>
      <ServerId>10</ServerId>
      <ParentId>0</ParentId>
      <DisplayName>Sent Items</DisplayName>
      <Type>5</Type>
    </Add>
    <Add>
      <ServerId>11</ServerId>
      <ParentId>0</ParentId>
      <DisplayName>Tasks</DisplayName>
      <Type>7</Type>
    </Add>
    <Add>
      <ServerId>RI</ServerId>
      <ParentId>0</ParentId>
      <DisplayName>RecipientInfo</DisplayName>
      <Type>19</Type>
    </Add>
  </Changes>
</FolderSync>
```

4.5 Synchronizing Data by Using the Sync Command

The following sections show how to accomplish several different tasks by using the **Sync** command (section 2.2.2.19).

4.5.1 Downloading Current Information from the Server

The following sections show how a client synchronizes an email folder by using the **Sync** command (section 2.2.2.19).

4.5.1.1 Request

The following example shows the **Sync** command request (section 2.2.2.19) sent from the client to the server to synchronize an email folder. The request asks that deleted items be moved to the Deleted Items folder. The request also asks for changes on the server to be included in the response.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync">
    <Collections>
        <Collection>
            <SyncKey>6</SyncKey>
            <CollectionId>1</CollectionId>
            <DeletesAsMoves/>
            <GetChanges/>
            <Options> ... </Options>
        </Collection>
    </Collections>
</Sync>
```

4.5.1.2 Response

The following example shows the **Sync** command response (section 2.2.2.19) sent from the server to the client. The response contains the new synchronization key and the items to be added, deleted, and changed on the client.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync">
    <Collections>
        <Collection>
            <SyncKey>7</SyncKey>
            <CollectionId>1</CollectionId>
            <Status>1</Status>
            <Commands>
                <Add>...</Add>
                <Delete>...</Delete>
                <Change>...</Change>
                <Fetch>...</Fetch>
            </Commands>
        </Collection>
    </Collections>
```

Release: February 10, 2014

```
    </Sync>
```

4.5.2 Fetching an Email by Using the ServerId

The following sections show how a client retrieves an item from the server by using the **Sync** command (section 2.2.2.19) and the item's server ID (**airsync:ServerId** (section 2.2.3.151.6)).

4.5.2.1 Request

The following example shows the **Sync** command request (section 2.2.2.19) sent from the client to the server to retrieve an email.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync">
…
    <Commands>
        <Fetch >
            <ServerId>1:14</ServerId>
        </Fetch >
    </Commands>
…
</Sync>
```

4.5.2.2 Response

The following example shows the **Sync** command response (section 2.2.2.19) sent from the server to the client. The response contains the server ID, status, and application data of the requested item.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync">
…
    <Responses>
        <Fetch>
            <ServerId>1:14</ServerId>
            <Status>1</Status>
            <ApplicationData>...</ApplicationData>
        </Fetch>
    </Responses>
…
</Sync>
```

4.5.3 Uploading New ApplicationData to the Server

The following sections show how a client uploads new application data to the server by using the **Sync** command (section 2.2.2.19).

4.5.3.1 Request

The following example shows the **Sync** command request (section 2.2.2.19) sent from the client to the server to add a contact.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:contacts="Contacts" xmlns="AirSync">
…
    <Commands>
        <Add>
            <ClientId>123</ClientId>
            <ApplicationData>
                <contacts:Email1Address>anat@fourthcoffee.com</contacts:Email1Address>
                <contacts:FirstName>Anat</contacts:FirstName>
                <contacts:MiddleName>W</contacts:MiddleName>
                <contacts:LastName>Kerry</contacts:LastName>
                <contacts:Title>Sr Marketing Manager</contacts:Title>
            </ApplicationData>
        </Add>
    </Commands>
…
</Sync>
```

4.5.3.2 Response

The following example shows the **Sync** command response (section 2.2.2.19) sent from the server to the client. The response shows that the synchronization was successful and that the new item from the client, identified by the **airsync:ClientId** element (section 2.2.3.28.2), was added to the collection on the server. The server also assigns a permanent ID for the newly added item in the **airsync:ServerId** element (section 2.2.3.151.6). After the client receives a successful response, the client uses this server ID for any future **airsync:Change** (section 2.2.3.24) or **airsync:Delete** (section 2.2.3.42.2) commands for this item.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync">
…
    <Responses>
        <Add>
            <ClientId>123</ClientId>
            <ServerId>4:1</ServerId>
            <Status>1</Status>
        </Add>
    </Responses>
…
</Sync>
```

4.5.4 Updating ApplicationData on the Server

The following sections show how a client updates application data on the server by using the **Sync** command (section 2.2.2.19).

4.5.4.1 Request

The following example shows the **Sync** command request (section 2.2.2.19) sent from the client to the server to modify the contact that is identified by the server ID.

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<Sync xmlns:contacts="Contacts" xmlns="AirSync">
…
    <Commands>
        <Change>
            <ServerId>3:1</ServerId>
            <ApplicationData>
                <contacts:Email1Address>anat@fourthcoffee.com</contacts:Email1Address>
                <contacts:FirstName>Anat</contacts:FirstName>
            </ApplicationData>
        </Change>
    </Commands>
…
</Sync>
```

4.5.4.2 Response

The following example shows the **Sync** command response (section 2.2.2.19) sent from the server to the client. The response shows that the change that is identified by the **Sync** request of the client succeeded and supplies the synchronization key and collection ID of the changed item.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync">
…
    <Collections>
        <Collection>
            <SyncKey>4</SyncKey>
            <CollectionId>1</CollectionId>
            <Status>1</Status>
        </Collection>
    </Collections>
…
</Sync>
```

4.5.5 Deleting an Item from the Server

The following sections show how a client deletes an email message from the server by using the **Sync** command (section 2.2.2.19).

4.5.5.1 Request

The following example shows the **Sync** command request (section 2.2.2.19) sent from the client to the server to delete the email message that is identified by the **airsync:ServerId** element (section 2.2.3.151.6).

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync">
  <Collections>
    <Collection>
      <SyncKey>1537401391</SyncKey>
      <CollectionId>5</CollectionId>
      <DeletesAsMoves>1</DeletesAsMoves>
      <GetChanges>1</GetChanges>
      <WindowSize>512</WindowSize>
```

```
            <Commands>
              <Delete>
                <ServerId>5:1</ServerId>
              </Delete>
            </Commands>
          </Collection>
        </Collections>
      </Sync>
```

4.5.5.2 Response

The following example shows the **Sync** command response (section 2.2.2.19) sent from the server to the client. The response includes the incremented **airsync:SyncKey** element (section 2.2.3.166.4) value, an **airsync:Status** element (section 2.2.3.162.16) value of 1, indicating the success of the message deletion, and the **airsync:CollectionId** element (section 2.2.3.30.5) value of the updated collection.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync">
  <Collections>
    <Collection>
      <SyncKey>609636509</SyncKey>
      <CollectionId>5</CollectionId>
      <Status>1</Status>
    </Collection>
  </Collections>
</Sync>
```

4.5.6 Identifying Acceptance of Partial Collections

The following example shows a **Sync** command request (section 2.2.2.19) that includes the **Partial** element (section 2.2.3.121). The **Partial** element indicates that the current **Sync** request does not include all of the settings for all the collections to be synchronized, and that the server is to use the values from the previous **Sync** request.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync">
    <Collections>
        <Collection>
            <SyncKey>1723058747</SyncKey>
            <CollectionId>10</CollectionId>
        </Collection>
    </Collections>
    <Wait>8</Wait>
    <Partial/>
</Sync>
```

4.5.7 Identifying Acceptance of MIME Content

The following sections show how to use the **airsync:MIMESupport** element (section 2.2.3.100.3), the **airsync:MIMETruncation** element (section 2.2.3.101), and the **airsyncbase:Type** element (as described in [MS-ASAIRS] section 2.2.2.22.4) to identify acceptance of MIME content on the client.

4.5.7.1 Sync Request With Support for MIME Content

The following example shows a **Sync** command request (section 2.2.2.19) that enables MIME support for email items that are returned from the server to the client, and requests that all MIME data of an email item over 4,096 characters be truncated when returned from the server to the client.

```xml
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync">
  <Collections>
    <Collection>
      <SyncKey>2</SyncKey>
      <CollectionId>1</CollectionId>
      <DeletesAsMoves/>
      <GetChanges/>
      <Options>
        <MIMETruncation>1</MIMETruncation>
        <MIMESupport>1</MIMESupport>
      </Options>
    </Collection>
  </Collections>
</Sync>
```

4.5.7.2 Sync Response with MIME Content

The following example shows a **Sync** command response (section 2.2.2.19) with MIME content sent from the server to the client.

```xml
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:email="Email" xmlns:airsyncbase="AirSyncBase" xmlns="AirSync">
  …
  <Add>
    <ServerId>1:3</ServerId>
    <ApplicationData>
      <email:To>"Dag Rovik" &lt;dag@contoso.com&gt;</email:To>
      <email:From>"Jan Kotas" &lt;jan@contoso.com&gt;</email:From>
      <email:Subject>RE: Presentation</email:Subject>
      <email:DateReceived>2004-11-12T00:45:06.000Z</email:DateReceived>
      <email:DisplayTo>Dag Rovik</email:DisplayTo>
      <email:Importance>1</email:Importance>
      <email:Read>0</email:Read>
      <email:Importance>1</email:Importance>
      <email:Read>0</email:Read>
      <airsyncbase:Body>
              <airsyncbase:Type>4</airsyncbase:Type>
              <airsyncbase:EstimatedDataSize>1</airsyncbase:EstimatedDataSize>
              <airsyncbase:Truncated>5408</airsyncbase:Truncated>
              <airsyncbase:Data>
                 <!--Content removed -->
              </airsyncbase:Data>
      </airsyncbase:Body>
      <email:MessageClass>IPM.Note.SMIME.MultipartSigned</email:MessageClass>
      <email:InternetCPID>20127</email:InternetCPID>
      <email:Flag/>
      <email:ContentClass>urn:content-classes:message</email:ContentClass>
      <airsyncbase:NativeBodyType>1</airsyncbase:NativeBodyType>
    </ApplicationData>
```

```
    </Add>
    …
  </Sync>
```

4.5.7.3   Sync Request with BodyPreference and MIME Support

The following example shows a **Sync** command request (section 2.2.2.19) that enables MIME
support for email items that are returned from the server to the client.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync" xmlns:airsyncbase="AirSyncBase">
  <Collections>
    <Collection>
      <SyncKey>2</SyncKey>
      <CollectionId>17</CollectionId>
      <DeletesAsMoves/>
      <GetChanges/>
      <Options>
        <airsyncbase:BodyPreference>
          <airsyncbase:Type>4</airsyncbase:Type>
          <airsyncbase:TruncationSize>512</airsyncbase:TruncationSize>
        </airsyncbase:BodyPreference>
        <MIMESupport>1</MIMESupport>
      </Options>
    </Collection>
  </Collections>
</Sync>
```

4.5.7.4   Sync Response with MIME Support

The following example shows a **Sync** command response (section 2.2.2.19) with MIME content sent
from the server to the client.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync" xmlns:email="Email" xmlns:email2="Email2"
xmlns:airsyncbase="AirSyncBase">
  <Collections>
    <Collection>
      <SyncKey>3</SyncKey>
      <CollectionId>17</CollectionId>
      <Status>1</Status>
      <Commands>
        <Change>
          <ServerId>17:11</ServerId>
          <ApplicationData>
            <email:To>"Leonor Marques" &lt;leonor@contoso.com&gt;</email:To>
            <email:From>"Wei Yu" &lt;wei@contoso.com&gt;</email:From>
            <email:Subject>opaque s + e </email:Subject>
            <email:DateReceived>2007-02-01T06:42:46.015Z</email:DateReceived>
            <email:DisplayTo>Leonor Marques</email:DisplayTo>
            <email:ThreadTopic>opaque s + e</email:ThreadTopic>
            <email:Importance>1</email:Importance>
            <email:Read>1</email:Read>
            <airsyncbase:Attachments>
              <airsyncbase:Attachment>
```

```
              <airsyncbase:DisplayName>smime.p7m</airsyncbase:DisplayName>
              <airsyncbase:FileReference>17%3a11%3a0</airsyncbase:FileReference>
              <airsyncbase:Method>1</airsyncbase:Method>
              <airsyncbase:EstimatedDataSize>9340</airsyncbase:EstimatedDataSize>
            </airsyncbase:Attachment>
          </airsyncbase:Attachments>
          <airsyncbase:Body>
            <airsyncbase:Type>4</airsyncbase:Type>
            <airsyncbase:EstimatedDataSize>13814</airsyncbase:EstimatedDataSize>
            <airsyncbase:Truncated>1</airsyncbase:Truncated>
            <airsyncbase:Data>Received: from contoso.com ([157.55.97.121])
            By contoso.com ([157.55.97.121]) with mapi;
            Wed, 31 Jan 2007 22:42:45 -0800
            From: Wei Yu &lt;wei@contoso.com&gt;
            To: Leonor &lt;leonor@contoso.com&gt;
            Content-Class: urn:content-classes:message
            Date: Wed, 31 Jan 2007 22:42:41 -0800
            Subject: opaque s + e
            Thread-Topic: opaque s + e
            Thread-Index: AcdFzCv5tyCXieBuTd2I5APpEvS+iQ==
            Message-ID:
            &lt;3AA64EB47EA90</airsyncbase:Data>
          </airsyncbase:Body>
          <email:MessageClass>IPM.Note.SMIME</email:MessageClass>
          <email:InternetCPID>20127</email:InternetCPID>
          <email:Flag/>
          <email:ContentClass>urn:content-classes:message</email:ContentClass>
          <airsyncbase:NativeBodyType>1</airsyncbase:NativeBodyType>
          <email2:ConversationId>…</email2:ConversationId>
          <email2:ConversationIndex>…</email2:ConversationIndex>
        </ApplicationData>
      </Change>
    </Commands>
  </Collection>
 </Collections>
</Sync>
```

4.5.8   Identifying That More Content is Ready for Download

The following example shows a **Sync** command response (section 2.2.2.19) message that indicates that more content is available for download from the server. Because the content exceeded the value indicated by the **airsync:WindowSize** element (section 2.2.3.183) in the request, the **airsync:MoreAvailable** element (section 2.2.3.106) is included in the response, indicating that additional content is available on the server for download.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync">
…
    <Collection>
        <SyncKey>2</SyncKey>
        <CollectionId>1</CollectionId>
        <Status>1</Status>
        <Commands>
             ...
        </Commands>
        <MoreAvailable/>
    </Collection>
```

```
…
</Sync>
```

4.5.9 Synchronizing the Calendar Folder

The following sections show how a client performs the initial synchronization of the Calendar folder by using the **Sync** command (section 2.2.2.19).

4.5.9.1 Initial Request

The following example shows the initial **Sync** command request (section 2.2.2.19) sent from the client to the server used to retrieve a nonzero **airsync:SyncKey** (section 2.2.3.166.4) value for the collection. The nonzero **airsync:SyncKey** value is used by a second **Sync** command request to retrieve the contents of the collection. The **airsync:CollectionId** element (section 2.2.3.30.5) identifies the Calendar folder as the collection being synchronized.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync">
  <Collections>
    <Collection>
      <SyncKey>0</SyncKey>
      <CollectionId>1</CollectionId>
      <DeletesAsMoves>1</DeletesAsMoves>
      <WindowSize>100</WindowSize>
    </Collection>
  </Collections>
</Sync>
```

4.5.9.2 Initial Response

The following example shows the initial **Sync** command response (section 2.2.2.19) used to synchronize the Calendar folder. The **airsync:SyncKey** (section 2.2.3.166.4) value returned in the response is nonzero and is used in the next request to retrieve the contents of the Calendar folder.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync">
  <Collections>
    <Collection>
      <SyncKey>1024910360</SyncKey>
      <CollectionId>1</CollectionId>
      <Status>1</Status>
    </Collection>
  </Collections>
</Sync>
```

4.5.9.3 Second Request

The following example shows the second **Sync** command request (section 2.2.2.19) used to synchronize the Calendar folder. The nonzero **airsync:SyncKey** element (section 2.2.3.166.4) value from the previous **Sync** response is included in this request to retrieve the contents of the Calendar folder.

```xml
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync">
  <Collections>
    <Collection>
      <SyncKey>1024910360</SyncKey>
      <CollectionId>1</CollectionId>
      <DeletesAsMoves>1</DeletesAsMoves>
      <GetChanges>1</GetChanges>
      <WindowSize>100</WindowSize>
    </Collection>
  </Collections>
</Sync>
```

4.5.9.4   Second Response

The following example shows the **Sync** command response (section 2.2.2.19) used to synchronize the contents of the Calendar folder.

```xml
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:calendar="Calendar" xmlns:airsyncbase="AirSyncBase" xmlns="AirSync">
  <Collections>
    <Collection>
      <SyncKey>1024910365</SyncKey>
      <CollectionId>1</CollectionId>
      <Status>1</Status>
      <Commands>
        <Add>
          <ServerId>1:1</ServerId>
          <ApplicationData>

<calendar:TimeZone>aAEAACgARwBNAFQALQAwADYAOgAwADAAKQAgAEMAZQBuAHQAcgBhAGwAIABUAGkAbQBlACAAKA
BVAFMAIAAmACAAQwAAAAsAAAABAAIAAAAAAAAAAAAAACgARwBNAFQALQAwADYAOgAwADAAKQAgAEMAZQBuAHQAcgBhAGw
AIABUAGkAbQBlACAAKAABVAFMAIAAmACAAQwAAAMAAAACAAIAAAAAAAAxP///w==</calendar:TimeZone>
          <calendar:DtStamp>20091020T161724Z</calendar:DtStamp>
          <calendar:StartTime>20091020T163000Z</calendar:StartTime>
          <calendar:Subject>Status Meeting</calendar:Subject>

<calendar:UID>040000008200E00074C5B7101A82E0080000000039F2FBCEA051CA010000000000000000001000000
0D9221B7FF439514E87E0D2FE13295CE2</calendar:UID>
          <calendar:OrganizerName>Kim Truelsen</calendar:OrganizerName>
          <calendar:OrganizerEmail>kim@contoso.com</calendar:OrganizerEmail>
          <calendar:Attendees>
            <calendar:Attendee>
              <calendar:Email>toby@contoso.com</calendar:Email>
              <calendar:Name>Toby Nixon</calendar:Name>
              <calendar:AttendeeType>1</calendar:AttendeeType>
            </calendar:Attendee>
          </calendar:Attendees>
          <calendar:Location>My office</calendar:Location>
          <calendar:EndTime>20091020T173000Z</calendar:EndTime>
          <calendar:Recurrence>
            <calendar:Type>1</calendar:Type>
            <calendar:Interval>1</calendar:Interval>
            <calendar:DayOfWeek>4</calendar:DayOfWeek>
          </calendar:Recurrence>
          <airsyncbase:Body>
            <airsyncbase:Type>3</airsyncbase:Type>
            <airsyncbase:EstimatedDataSize>275</airsyncbase:EstimatedDataSize>
```

```
                        <airsyncbase:Truncated>1</airsyncbase:Truncated>
                    </airsyncbase:Body>
                    <calendar:Sensitivity>0</calendar:Sensitivity>
                    <calendar:BusyStatus>1</calendar:BusyStatus>
                    <calendar:AllDayEvent>0</calendar:AllDayEvent>
                    <calendar:Reminder>15</calendar:Reminder>
                    <calendar:Exceptions>
                      <calendar:Exception>
                        <calendar:DtStamp>20091020T162543Z</calendar:DtStamp>
                        <calendar:Location>Your office</calendar:Location>
                        <airsyncbase:Body>
                          <airsyncbase:Type>3</airsyncbase:Type>
                          <airsyncbase:EstimatedDataSize>273</airsyncbase:EstimatedDataSize>
                          <airsyncbase:Truncated>1</airsyncbase:Truncated>
                        </airsyncbase:Body>
                        <calendar:Categories />
                        <calendar:ExceptionStartTime>20091020T163000Z</calendar:ExceptionStartTime>
                        <calendar:BusyStatus>2</calendar:BusyStatus>

  <calendar:AppointmentReplyTime>20091020T180257Z</calendar:AppointmentReplyTime>
                        <calendar:ResponseType>3</calendar:ResponseType>
                      </calendar:Exception>
                      <calendar:Exception>
                        <calendar:DtStamp>20091027T161136Z</calendar:DtStamp>
                        <calendar:StartTime>20091027T163000Z</calendar:StartTime>
                        <calendar:EndTime>20091027T173000Z</calendar:EndTime>
                        <calendar:Categories />
                        <calendar:ExceptionStartTime>20091027T163000Z</calendar:ExceptionStartTime>
                        <calendar:BusyStatus>2</calendar:BusyStatus>

  <calendar:AppointmentReplyTime>20091027T161136Z</calendar:AppointmentReplyTime>
                        <calendar:ResponseType>3</calendar:ResponseType>
                      </calendar:Exception>
                    </calendar:Exceptions>
                    <calendar:MeetingStatus>3</calendar:MeetingStatus>
                    <airsyncbase:NativeBodyType>3</airsyncbase:NativeBodyType>
                    <calendar:ResponseRequested>1</calendar:ResponseRequested>
                    <calendar:ResponseType>2</calendar:ResponseType>
                  </ApplicationData>
                </Add>
                <Add>
                  <ServerId>1:2</ServerId>
                  <ApplicationData>

<calendar:TimeZone>4AEAACgARwBNAFQALQAwADgAOgAwADAAKQAgAFAAYBjAGkAZgBpAGMAIABUAGkAbQBlACAAKA
BVAFMAIAAmACAAQwAAAAsAAAABAAIAAAAAAAAAAAACgARwBNAFQALQAwADgAOgAwADAAKQAgAFAAYBjAGkAZgBpAGMA
AIABUAGkAbQBlACAAKABVAFMAIAAmACAAQwAAAAMAAACAAIAAAAAAAAAxP///w==</calendar:TimeZone>
                    <calendar:DtStamp>20091027T161226Z</calendar:DtStamp>
                    <calendar:StartTime>20091029T163000Z</calendar:StartTime>
                    <calendar:Subject>Planning meeting</calendar:Subject>

<calendar:UID>040000008200E00074C5B7101A82E008000000007D5639462057CA010000000000000000001000000
065870766FB4D69479C217C16F9AD5E5B</calendar:UID>
                    <calendar:OrganizerName>Kim Truelsen</calendar:OrganizerName>
                    <calendar:OrganizerEmail>kim@contoso.com</calendar:OrganizerEmail>
                    <calendar:Location />
                    <calendar:EndTime>20091029T173000Z</calendar:EndTime>
                    <airsyncbase:Body>
                      <airsyncbase:Type>3</airsyncbase:Type>
```

```
                <airsyncbase:EstimatedDataSize>238</airsyncbase:EstimatedDataSize>
                <airsyncbase:Truncated>1</airsyncbase:Truncated>
            </airsyncbase:Body>
            <calendar:Sensitivity>0</calendar:Sensitivity>
            <calendar:BusyStatus>2</calendar:BusyStatus>
            <calendar:AllDayEvent>0</calendar:AllDayEvent>
            <calendar:Reminder>15</calendar:Reminder>
            <calendar:MeetingStatus>0</calendar:MeetingStatus>
            <airsyncbase:NativeBodyType>3</airsyncbase:NativeBodyType>
            <calendar:ResponseRequested>1</calendar:ResponseRequested>
            <calendar:ResponseType>1</calendar:ResponseType>
          </ApplicationData>
        </Add>
      </Commands>
    </Collection>
  </Collections>
</Sync>
```

4.5.10 Empty Sync Request and Response

This section demonstrates a scenario in which an empty **Sync** request (section 2.2.2.19.1) and an empty **Sync** response (section 2.2.2.19.2) are exchanged between the client and server.

The scenario begins when a **Sync** request is issued by the client and indicates that there are no pending changes to report to the server. The **Sync** request is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync">
  <Collections>
    <Collection>
      <SyncKey>560109801</SyncKey>
      <CollectionId>5</CollectionId>
      <DeletesAsMoves>1</DeletesAsMoves>
      <GetChanges>1</GetChanges>
      <WindowSize>512</WindowSize>
    </Collection>
  </Collections>
  <HeartbeatInterval>60</HeartbeatInterval>
  <WindowSize>512</WindowSize>
</Sync>
```

When the server receives this **Sync** request and determines that it contains no changes, it caches the request for future use. The server then responds to the **Sync** request with an empty **Sync** response when no changes or errors have occurred on the server. The empty **Sync** response is as follows:

```
HTTP/1.1 200 OK
Date: Fri, 10 Apr 2009 20:32:39 GMT
Content-Length: 0
```

When the client receives the empty **Sync** response, it can in turn send an empty **Sync** request if there are no pending changes. The empty **Sync** request is as follows:

```
POST /Microsoft-Server-
ActiveSync?Cmd=Sync&User=DeviceUser&DeviceId=v140Device&DeviceType=SmartPhone HTTP/1.1
Content-Type: application/vnd.ms-sync.wbxml
MS-ASProtocolVersion: 14.0
X-MS-PolicyKey: 2401271238
User-Agent: ASOM
Host: contoso.com
```

The exchange of the empty **Sync** requests and responses continues until a change is detected on either the client or server, at which time a **Sync** request or response with an XML payload is sent.

4.5.11   Receiving a Substitute Meeting Invitation Email Message

The following example shows a **Sync** command response (section 2.2.2.19) that contains a substitute meeting invitation email message, as specified in section 3.1.5.5.1. In this example, Jason Carlson has configured his calendar to give Toby Nixon delegate access. Toby receives copies of all meeting requests sent to Jason. David Jones has sent a meeting request to Jason with the following information:

Meeting request field	Value
Subject	"Meet for lunch"
Start time	"2012-02-23T16:00:00.000Z"
End time	"2012-02-23T17:00:00.000Z"
Location	"Cafeteria downstairs"
Body	"Let's meet for lunch to discuss upcoming projects."

The server sends a **Sync** command response that contains the following.

```
<ApplicationData xmlns="AirSync" xmlns:email="Email" xmlns:email2="Email2"
xmlns:airsyncbase="AirSyncBase">
  <email:To>"Toby Nixon" &lt;toby@contoso.com&gt;</email:To>
  <email:From>"David Jones" &lt;david@contoso.com&gt;</email:From>
  <email:Subject>Meeting request for Jason Carlson: Meet for lunch</email:Subject>
  <email:DateReceived>2012-02-23T15:51:02.277Z</email:DateReceived>
  <email:DisplayTo>Toby Nixon</email:DisplayTo>
  <email:ThreadTopic>Meet for lunch</email:ThreadTopic>
  <email:Importance>1</email:Importance>
  <email:Read>1</email:Read>
  <airsyncbase:Body>
    <airsyncbase:Type>3</airsyncbase:Type>
    <airsyncbase:EstimatedDataSize>429</airsyncbase:EstimatedDataSize>
    <airsyncbase:Truncated>1</airsyncbase:Truncated>
  </airsyncbase:Body>
  <email:MessageClass>IPM.Note</email:MessageClass>
  <email:InternetCPID>28591</email:InternetCPID>
  <email:Flag />
  <email:ContentClass>urn:content-classes:message</email:ContentClass>
  <airsyncbase:NativeBodyType>3</airsyncbase:NativeBodyType>
  <email2:ConversationId>...</email2:ConversationId>
  <email2:ConversationIndex>...</email2:ConversationIndex>
  <email:Categories />
```

```
</airsync:ApplicationData>
```

The body of the item retrieved via the **ItemOperations** command (section 2.2.2.8) has the following value:

```
When: Thursday, February 23, 2012 11:00 AM-12:00 PM. (UTC-05:00) Eastern Time (US & Canada)
Where: Cafeteria downstairs


*~*~*~*~*~*~*~*~*~*


Let's meet for lunch to discuss upcoming projects.
```

4.6 Sending Email Messages by Using the SendMail Command

The following sections show how to send email messages by using the **SendMail** command (section 2.2.2.15).

4.6.1 Request

The following example shows the **SendMail** command request (section 2.2.2.15) sent from the client to the server to send an email message. The **composemail:ClientId** element (section 2.2.3.28.1) represents the unique message ID of the message to be sent. The **composemail:SaveInSentItems** element (section 2.2.3.144) indicates to the server that a copy of the message is placed in the Sent Items folder, and the **composemail:Mime** element (section 2.2.3.99) contains the message contents.

```
<?xml version="1.0" encoding="utf-8"?>
<SendMail xmlns="ComposeMail">
  <ClientId>633916348086136194</ClientId>
  <SaveInSentItems />
  <Mime>From: testuser1
To: testuser2
Cc:
Bcc:
Subject:
MIME-Version: 1.0
Content-Type: text/plain; charset="iso-8859-1"
Content-Transfer-Encoding: 7bit
X-MimeOLE: Produced By Microsoft MimeOLE V6.00.2900.3350
This is the email body content.</Mime>
</SendMail>
```

4.6.2 Response

The following example shows the successful **SendMail** command response (section 2.2.2.15) sent from the server to the client.

```
HTTP/1.1 200 OK
Date: Wed, 24 Feb 2010 16:33:31 GMT
Content-Length: 0
```

4.7 Replying to Email Messages by Using the SmartReply Command

The following sections show how to reply to email messages by using the **SmartReply** command (section 2.2.2.18).

4.7.1 Request

The following example shows the **SmartReply** command request (section 2.2.2.18) sent from the client to the server to reply to an email message. The **composemail:ClientId** element (section 2.2.3.28.1) represents the unique message ID of the reply message to be sent. The **composemail:SaveInSentItems** element (section 2.2.3.144) indicates to the server that a copy of the message is placed in the Sent Items folder. The **composemail:FolderId** element (section 2.2.3.69) identifies the folder that contains the message that is being responded to, the **composemail:ItemId** element (section 2.2.3.84) identifies the message being responded to, and the **composemail:Mime** element (section 2.2.3.99) contains the message contents of the reply message.

```
<?xml version="1.0" encoding="utf-8"?>
<SmartReply xmlns="ComposeMail">
  <ClientId>d7b99822-685a-4a40-8dfb-87a114926986</ClientId>
  <SaveInSentItems />
  <Source>
    <FolderId>5</FolderId>
    <ItemId>5:10</ItemId>
  </Source>
  <Mime>From: testuser1
To: "Test User 2" &lt;testuser2@contoso.com&gt;
Cc:
Bcc:
Subject: RE: Lunch
MIME-Version: 1.0
Content-Type: text/plain; charset="iso-8859-1"
Content-Transfer-Encoding: 7bit
X-MimeOLE: Produced By Microsoft MimeOLE V6.00.2900.3350
This is the body of the smart reply message.</Mime>
</SmartReply>
```

4.7.2 Response

The following example shows the successful **SmartReply** command response (section 2.2.2.18) sent from the server to the client. A successful **SmartReply** response contains no XML content.

```
HTTP/1.1 200 OK
Date: Wed, 22 Feb 2010 18:19:26 GMT
Content-Length: 0
```

4.8 Pinging the Server for Updates by Using the Ping Command

The following sections show how to query the server for updates by using the **Ping** command (section 2.2.2.11).

4.8.1 Ping Command Request

The following example shows the **Ping** command request (section 2.2.2.11) sent from the client to the server to query for new email received in the folder that has an **ping:Id** (section 2.2.3.81) value of 5.

```
<?xml version="1.0" encoding="utf-8"?>
<Ping xmlns="Ping">
  <HeartbeatInterval>80</HeartbeatInterval>
  <Folders>
    <Folder>
      <Id>5</Id>
      <Class>Email</Class>
    </Folder>
  </Folders>
</Ping>
```

4.8.2 Ping Command Response

4.8.2.1 Typical Response

The following example shows a typical **Ping** command response (section 2.2.2.11) when the heartbeat interval that was indicated by the client has expired and there were no changes in any of the identified folders.

```
<?xml version="1.0" encoding="utf-8"?>
<Ping xmlns="Ping">
  <Status>1</Status>
</Ping>
```

4.8.2.2 Response – Changes Found

The following example shows a **Ping** command response (section 2.2.2.11) that indicates that changes have occurred in two folders that were being monitored. The client responds to this information by synchronizing the folders identified in the response (before reissuing the next **Ping** command request).

```
<?xml version="1.0" encoding="utf-8"?>
<Ping xmlns="Ping">
  <Status>2</Status>
  <Folders>
    <Folder>1234</Folder>
    <Folder>5678</Folder>
  </Folders>
</Ping>
```

4.8.2.3 Response – HeartbeatInterval Error

The following example shows a **Ping** command response (section 2.2.2.11) to a **Ping** command request that indicated a heartbeat interval outside the acceptable range. The returned heartbeat interval is either the minimum or maximum allowed value. The client responds to this information by comparing the requested interval with the returned interval and determines whether the requested heartbeat interval was either too great or too small.

```
<?xml version="1.0" encoding="utf-8"?>
<Ping xmlns="Ping">
  <Status>5</Status>
  <HeartbeatInterval>60</HeartbeatInterval>
</Ping>
```

4.8.2.4   Response – Folder Error

The following example shows a **Ping** command response (section 2.2.2.11) to a **Ping** command request where the number of folders that was indicated was greater than the maximum number of folders that are allowed to be monitored. The maximum number of folders that are allowed to be monitored is returned in the **ping:MaxFolders** element (section 2.2.3.92).

```
<?xml version="1.0" encoding="utf-8"?>
<Ping xmlns="Ping">
  <Status>6</Status>
  <MaxFolders>200</MaxFolders>
</Ping>
```

4.9   Retrieving Item Estimates by Using the GetItemEstimate Command

The following sections show how to retrieve the number of items in a folder that require synchronization by using the **GetItemEstimate** command (section 2.2.2.7).

4.9.1   Request

The following example shows the **GetItemEstimate** command request (section 2.2.2.7) sent from the client to the server to retrieve the number of items in a folder that require synchronization. The **airsync:SyncKey** element (section 2.2.3.166.3) in the request is set to the value returned by the last **Sync** command response (section 2.2.2.19). The **getitemestimate:CollectionId** element (section 2.2.3.30.1) identifies the folder to query, and the **airsync:Options** (section 2.2.3.115.1) elements indicate any time filters or specific content class items to retrieve.

```
<?xml version="1.0" encoding="utf-8"?>
<GetItemEstimate xmlns="GetItemEstimate" xmlns:airsync="AirSync">
  <Collections>
    <Collection>
      <airsync:SyncKey>785971245</airsync:SyncKey>
      <CollectionId>5</CollectionId>
      <airsync:Options>
        <airsync:FilterType>0</airsync:FilterType>
        <airsync:Class>Email</airsync:Class>
      </airsync:Options>
    </Collection>
  </Collections>
</GetItemEstimate>
```

4.9.2   Response

The following example shows the **GetItemEstimate** command response (section 2.2.2.7) sent from the server to the client. A **getitemestimate:Status** element (section 2.2.3.162.6) value of 1 is
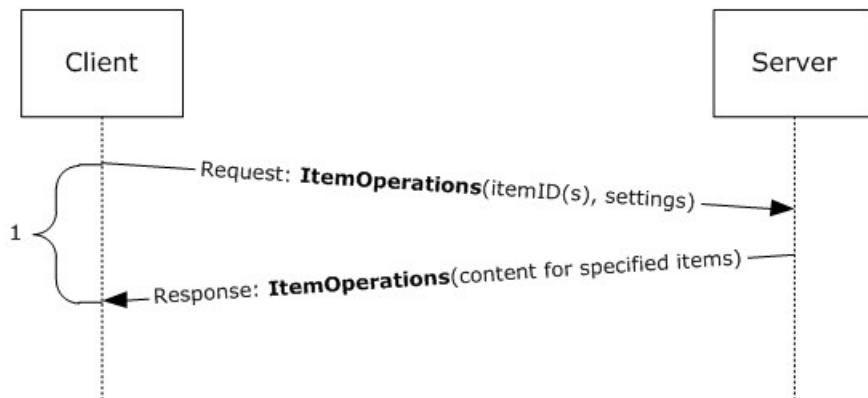
returned to indicate that the **GetItemEstimate** request (section 2.2.2.7) was completed successfully. The **getitemestimate:CollectionId** element (section 2.2.3.30.1) identifies the folder that was queried, and the **getitemestimate:Estimate** element (section 2.2.3.62) indicates the number of items that require synchronization in the folder.

```xml
<?xml version="1.0" encoding="utf-8"?>
<GetItemEstimate xmlns="GetItemEstimate">
  <Response>
    <Status>1</Status>
    <Collection>
      <CollectionId>5</CollectionId>
      <Estimate>1</Estimate>
    </Collection>
  </Response>
</GetItemEstimate>
```

4.10   Fetching Email and Attachments by Using the ItemOperations Command

The **ItemOperations** command (section 2.2.2.8) enables the client to retrieve PIM items and attachments (in addition to document library items and search results) outside the **Sync** command (section 2.2.2.19) context.

The following sections contain examples that focus on retrieval of items and attachments, following a simple request and response model. The following figure shows the request and response model used in fetching email and attachments.

Figure 6: Fetching Email

4.10.1   Fetching an Email Item

The following sections show how the client retrieves an email message by using the **ItemOperations** command (section 2.2.2.8).

4.10.1.1   Request

The following example shows the **ItemOperations** command request (section 2.2.2.8) sent from the client to the server to retrieve an email message.

```xml
<?xml version="1.0" encoding="utf-8"?>
```

```
<ItemOperations xmlns:airsync="AirSync"
xmlns:airsyncbase="AirSyncBase" xmlns="ItemOperations">
  <Fetch>
    <Store>Mailbox</Store>
    <airsync:CollectionId>7</airsync:CollectionId>
    <airsync:ServerId>7:1</airsync:ServerId>
    <Options>
      <airsyncbase:BodyPreference>
        <airsyncbase:Type>1</airsyncbase:Type>
        <airsyncbase:TruncationSize>5120</airsyncbase:TruncationSize>
        <airsyncbase:AllOrNone>0</airsyncbase:AllOrNone>
      </airsyncbase:BodyPreference>
    </Options>
  </Fetch>
</ItemOperations>
```

4.10.1.2 Response

The following example shows the **ItemOperations** command response (section 2.2.2.8) sent from the server to the client.

```
<?xml version="1.0" encoding="utf-8"?><ItemOperations
xmlns:airsync="AirSync" xmlns:email="Email" xmlns:email2="Email2"
xmlns="ItemOperations">
  <Status>1</Status>
  <Response>
    <Fetch>
      <Status>1</Status>
      <airsync:CollectionId>7</airsync:CollectionId>
      <airsync:ServerId>7:1</airsync:ServerId>
      <airsync:Class>Email</airsync:Class>
      <Properties>
        <email:To>"deviceuser" &lt;anat@contoso.com&gt;</email:To>
        <email:Cc>"deviceuser3" &lt;chris@contoso.com.com&gt;</email:Cc>
        <email:From>"deviceuser2" &lt;dag@contoso.com&gt;
        </email:From>
        <email:Subject>Subject</email:Subject>
        <email:DateReceived>2007-05-08T17:29:07.890Z
        </email:DateReceived>
        <email:DisplayTo>DeviceUserDisplayName</email:DisplayTo>
        <email:ThreadTopic>Subject</email:ThreadTopic>
        <email:Importance>1</email:Importance>
        <email:Read>0</email:Read>
        <airsyncbase:Body>
          <airsyncbase:Type>1</airsyncbase:Type>
          <airsyncbase:EstimatedDataSize>20
          </airsyncbase:EstimatedDataSize>
          <airsyncbase:Data>Body as plain text</airsyncbase:Data>
        </airsyncbase:Body>
        <email:MessageClass>IPM.Note</email:MessageClass>
        <email:InternetCPID>28591</email:InternetCPID>
        <email:Flag />
        <email:ContentClass>urn:content-classes:message
        </email:ContentClass>
        <airsyncbase:NativeBodyType>1</airsyncbase:NativeBodyType>
        <email2:ConversationId>…</email2:ConversationId>
        <email2:ConversationIndex>…</email2:ConversationIndex>
```

```
            </Properties>
         </Fetch>
      </Response>
   </ItemOperations>
```

4.10.2   Fetching a MIME Email Item

The following sections show how the client retrieves a MIME email message by using the
airsync:MIMESupport option (section 2.2.3.100.1).

4.10.2.1   Request

The following example shows the **ItemOperations** command request (section 2.2.2.8) sent from
the client to the server to retrieve a MIME email message.

```
<?xml version="1.0" encoding="utf-8"?>
<ItemOperations xmlns="ItemOperations" xmlns:airsync="AirSync"
xmlns:airsyncbase="AirSyncBase">
   <Fetch>
      <Store>Mailbox</Store>
      <airsync:CollectionId>17</airsync:CollectionId>
      <airsync:ServerId>17:11</airsync:ServerId>
      <Options>
         <airsync:MIMESupport>1</airsync:MIMESupport>
         <airsyncbase:BodyPreference>
            <airsyncbase:Type>4</airsyncbase:Type>
         </airsyncbase:BodyPreference>
      </Options>
   </Fetch>
</ItemOperations>
```

4.10.2.2   Response

The following example shows the **ItemOperations** command response (section 2.2.2.8) sent from
the server to the client.

```
<?xml version="1.0" encoding="utf-8"?>
<ItemOperations xmlns="ItemOperations" xmlns:airsync="AirSync" xmlns:email="Email"
xmlns:email2="Email2" xmlns:airsyncbase="AirSyncBase">
   <Status>1</Status>
   <Response>
      <Fetch>
         <Status>1</Status>
         <airsync:CollectionId>17</airsync:CollectionId>
         <airsync:ServerId>17:11</airsync:ServerId>
         <airsync:Class>Email</airsync:Class>
         <Properties>
            <email:To>"Anat Kerry" &lt;anat@contoso.com&gt;</email:To>
            <email:From>"Chris Gray" &lt;chris@contoso.com&gt;</email:From>
            <email:Subject>opaque s + e</email:Subject>
            <email:DateReceived>2007-02-01T06:42:46.015Z</email:DateReceived>
            <email:DisplayTo>Anat Kerry</email:DisplayTo>
            <email:ThreadTopic>opaque s + e</email:ThreadTopic>
            <email:Importance>1</email:Importance>
```

```
                <email:Read>1</email:Read>
                <airsyncbase:Attachments>
                    <airsyncbase:Attachment>
                        <airsyncbase:DisplayName>smime.p7m</airsyncbase:DisplayName>

<airsyncbase:FileReference>RgAAAAA4u8%2fWvU8lQ7GtLlC7V9V3BwCdyWYIRkOHRp2ozB%2f0DXQsAHgM%2bwAF
AAA6pk60fqkEQbWH4Wm%2bnjh7AHgNBA%2bgAAAJ%3a0</airsyncbase:FileReference>
                        <airsyncbase:Method>1</airsyncbase:Method>
                        <airsyncbase:EstimatedDataSize>9340</airsyncbase:EstimatedDataSize>
                    </airsyncbase:Attachment>
                </airsyncbase:Attachments>
                <airsyncbase:Body>
                    <airsyncbase:Type>4</airsyncbase:Type>
                    <airsyncbase:EstimatedDataSize>13813</airsyncbase:EstimatedDataSize>
                    <airsyncbase:Data>Received: from contoso.com ([157.55.97.121])
 by contoso.com ([157.55.97.121]) with mapi;
 Wed, 31 Jan 2007 22:42:45 -0800
From: Chris Gray &lt;chris@contoso.com&gt;
To: Anat Kerry &lt;anat@contoso.com&gt;
Content-Class: urn:content-classes:message
Date: Wed, 31 Jan 2007 22:42:41 -0800
Subject: opaque s + e
Thread-Topic: opaque s + e
Thread-Index: AcdFzCv5tyCXieBuTd2I5APpEvS+iQ==
Message-ID:
 &lt;3AA64EB47EA90441B587E169BE9E387B780D00C326@contoso.com&gt;
Content-Language: en-US
X-MS-Exchange-Organization-AuthAs: Internal
X-MS-Exchange-Organization-AuthMechanism: 04
X-MS-Exchange-Organization-AuthSource:
 contoso.com
X-MS-Has-Attach: yes
X-MS-Exchange-Organization-SCL: -1
X-MS-TNEF-Correlator:
acceptlanguage: en-US
Content-Type: application/x-pkcs7-mime; smime-type=enveloped-data;
name="smime.p7m"
Content-Disposition: attachment; filename="smime.p7m"
Content-Transfer-Encoding: base64
MIME-Version: 1.0

MIAGCSqGSIb3DQEHA6CAMIACAQAxggJEMIIBHgIBADCBhjB4MRMwEQYKCZImiZPyLGQBGRYDY29t
MRkwFwYKCZImiZPyLGQBGRYJbWljcm9zb2Z0MRYwFAYKCZImiZPyLGQBGRYGZXh0ZXN0MR0wGwYK
CZImiZPyLGQBGRYNamluZ2h1YWMwMURPTTEPMA0GA1UEAxMGVGVzdENBAgonJIo2AAAAAAHMA0G
(Large section of sample data removed)
                    </airsyncbase:Data>
                </airsyncbase:Body>
                <email:MessageClass>IPM.Note.SMIME</email:MessageClass>
                <email:InternetCPID>20127</email:InternetCPID>
                <email:Flag/>
                <email:ContentClass>urn:content-classes:message</email:ContentClass>
                <airsyncbase:NativeBodyType>1</airsyncbase:NativeBodyType>
                <email2:ConversationId>…</email2:ConversationId>
                <email2:ConversationIndex>…</email2:ConversationIndex>
            </Properties>
        </Fetch>
    </Response>
</ItemOperations>
```

4.10.3 Fetching an Email Item with a LongId

The following sections show how the client retrieves an email message by using the **search:LongId** element (section 2.2.3.89.3). The client uses the **Search** command (section 2.2.2.14) to get the **search:LongId** value corresponding to the message, and then uses the **itemoperations:Fetch** element (section 2.2.3.63.1) with the **search:LongId** element to retrieve the message.

4.10.3.1 Search Request

The following example shows the **Search** command request (section 2.2.2.14) sent from the client to the server to search for emails containing the text "Sales Totals".

```
<?xml version="1.0" encoding="utf-8"?>
<Search xmlns="Search" xmlns:airsync="AirSync"
xmlns:email="Email">
  <Store>
    <Name>Mailbox</Name>
    <Query>
      <And>
        <airsync:Class>Email</airsync:Class>
        <airsync:CollectionId>7</airsync:CollectionId>
        <FreeText>Sales Totals</FreeText>
      </And>
    </Query>
    <Options>
      <RebuildResults />
      <Range>0-4</Range>
    </Options>
  </Store>
</Search>
```

4.10.3.2 Search Response

The following example shows the **Search** command response (section 2.2.2.14) sent from the server to the client. The response includes email data for email that contains the string "Sales Totals". Included with the results is the **search:LongId** element (section 2.2.3.89.3) that uniquely identifies the search result.

```
<?xml version="1.0" encoding="utf-8"?><Search xmlns:airsync="AirSync"
xmlns:email="Email" xmlns:email2="Email2" xmlns:airsyncbase="AirSyncBase"
xmlns="Search">
  <Status>1</Status>
  <Response>
    <Store>
      <Status>1</Status>
      <Result>
        <airsync:Class>Email</airsync:Class>
        <LongId>RgAAAACYWCHnyBZ%2fTq8bujFmR1EPBwBzyWfENpcEQ7zU
        NyaWwM4BAAAA8FxEAABzyWfENpcEQ7zUNyaWwM4BAAAA8HACAAAJ</LongId>
        <airsync:CollectionId>7</airsync:CollectionId>
        <Properties>
          <email:To>"deviceuser" &lt;anat@contoso.com&gt;
          </email:To>
          <email:From>"deviceuser2" &lt;chris@contoso.com&gt;
          </email:From>
          <email:Subject>Sales Totals for April</email:Subject>
```

```
                    <email:DateReceived>2007-05-08T17:29:07.890Z
                    </email:DateReceived>
                    <email:DisplayTo>DeviceUserDisplayName</email:DisplayTo>
                    <email:ThreadTopic>Subject</email:ThreadTopic>
                    <email:Importance>1</email:Importance>
                    <email:Read>1</email:Read>
                    <airsyncbase:Body>
                       <airsyncbase:Type>1</airsyncbase:Type>
                       <airsyncbase:EstimatedDataSize>6
                       </airsyncbase:EstimatedDataSize>
                       <airsyncbase:Truncated>1</airsyncbase:Truncated>
                    </airsyncbase:Body>
                    <email:MessageClass>IPM.Note</email:MessageClass>
                    <email:InternetCPID>28591</email:InternetCPID>
                    <email:Flag />
                    <email:ContentClass>urn:content-classes:message
                    </email:ContentClass>
                    <airsyncbase:NativeBodyType>1</airsyncbase:NativeBodyType>
                    <email2:ConversationId>…</email2:ConversationId>
                    <email2:ConversationIndex>…</email2:ConversationIndex>
                 </Properties>
              </Result>
              <Range>0-0</Range>
              <Total>1</Total>
           </Store>
        </Response>
     </Search>
```

4.10.3.3   ItemOperations Request

The following example shows the **ItemOperations** command request (section 2.2.2.8) sent from the client to the server to retrieve the item found by the **Search** command (section 2.2.2.14). The **itemoperations:Fetch** element (section 2.2.3.63.1) includes the **search:LongId** element (section 2.2.3.89.3) that was returned in the **Search** command response.

```
<?xml version="1.0" encoding="utf-8"?>
<ItemOperations xmlns:airsync="AirSync"
xmlns:airsyncbase="AirSyncBase" xmlns:search="Search"
xmlns="ItemOperations">
  <Fetch>
    <Store>Mailbox</Store>
    <search:LongId>RgAAAACYWCHnyBZ%2fTq8bujFmR1EPBwBzyWfENpc
    EQ7zUNyaWwM4BAAAA8FxEAABzyWfENpcEQ7zUNyaWwM4BAAAA8HA
    CAAAJ</search:LongId>
    <Options>
      <airsyncbase:BodyPreference>
        <airsyncbase:Type>1</airsyncbase:Type>
      </airsyncbase:BodyPreference>
    </Options>
  </Fetch>
</ItemOperations>
```

4.10.3.4 ItemOperations Response

The following example shows the **ItemOperations** command response (section 2.2.2.8) sent from the server to the client. The response includes the **itemoperations:Fetch** element (section 2.2.3.63.1), which contains the complete email for the message.

```
<?xml version="1.0" encoding="utf-8"?><ItemOperations
xmlns:airsync="AirSync" xmlns:email="Email" xmlns:email2="Email2"
xmlns="ItemOperations">
  <Status>1</Status>
  <Response>
    <Fetch>
      <Status>1</Status>
      <search:LongId>RgAAAACYWCHnyBZ%2fTq8bujFmR1EPBwBzyWfENpcEQ7zU
        NyaWwM4BAAAA8FxEAABzyWfENpcEQ7zUNyaWwM4BAAAA8HACAAAJ</search:LongId>
      <airsync:Class>Email</airsync:Class>
      <Properties>
        <email:To>"deviceuser" &lt;anat@contoso.com&gt;</email:To>
        <email:From>"deviceuser2" &lt;chris@contoso.com&gt;
        </email:From>
        <email:Subject>Sales Totals for April</email:Subject>
        <email:DateReceived>2007-05-08T17:29:07.890Z
        </email:DateReceived>
        <email:DisplayTo>DeviceUserDisplayName</email:DisplayTo>
        <email:ThreadTopic>Subject</email:ThreadTopic>
        <email:Importance>1</email:Importance>
        <email:Read>1</email:Read>
        <airsyncbase:Body>
          <airsyncbase:Type>1</airsyncbase:Type>
          <airsyncbase:EstimatedDataSize>20
          </airsyncbase:EstimatedDataSize>
          <airsyncbase:Data>Income generated by the sales department
          in April can be attributed to the following...
          </airsyncbase:Data>
        </airsyncbase:Body>
        <email:MessageClass>IPM.Note</email:MessageClass>
        <email:InternetCPID>28591</email:InternetCPID>
        <email:Flag />
        <email:ContentClass>urn:content-classes:message
        </email:ContentClass>
        <airsyncbase:NativeBodyType>1</airsyncbase:NativeBodyType>
        <email2:ConversationId>…</email2:ConversationId>
        <email2:ConversationIndex>…</email2:ConversationIndex>
      </Properties>
    </Fetch>
  </Response>
</ItemOperations>
```

4.10.4 Fetching an Attachment

The following sections show how the client retrieves a new message with an attachment. The client uses the **Sync** command (section 2.2.2.19) to synchronize a new message with an attachment to the client and then uses the **ItemOperations** command (section 2.2.2.8) to retrieve the attachment.

In the XML schema code, HTML strings are escaped by using "<" and ">". However, as these values are passed over the wire, they are passed in their original HTML format, as "<" and ">".

4.10.4.1   Sync Request

The following example shows the **Sync** command request (section 2.2.2.19) sent from the client to
the server to synchronize a new message with an attachment to the client.

```xml
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:airsyncbase="AirSyncBase" xmlns="AirSync">
  <Collections>
    <Collection>
      <SyncKey>1</SyncKey>
      <CollectionId>7</CollectionId>
      <DeletesAsMoves />
      <GetChanges />
    </Collection>
  </Collections>
</Sync>
```

4.10.4.2   Sync Response

The following example shows the **Sync** command response (section 2.2.2.19) sent from the server
to the client.

```xml
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:email="Email"
xmlns:airsyncbase="AirSyncBase" xmlns="AirSync">
  <Collections>
    <Collection>
      <SyncKey>2</SyncKey>
      <CollectionId>7</CollectionId>
      <Status>1</Status>
      <Commands>
        <Add>
          <ServerId>7:1</ServerId>
          <ApplicationData>
            <email:To>"deviceuser" &lt;anat@contoso.com&gt;
            </email:To>
            <email:From>"deviceuser2" &lt;chris@contoso.com&gt;
            </email:From>
            <email:Subject>Email with Attachment</email:Subject>
            <email:DateReceived>2007-05-08T17:57:22.890Z
            </email:DateReceived>
            <email:DisplayTo>deviceuser</email:DisplayTo>
            <email:ThreadTopic>Email with Attachment
            </email:ThreadTopic>
            <email:Importance>1</email:Importance>
            <email:Read>0</email:Read>
            <airsyncbase:Attachments>
              <airsyncbase:Attachment>
                <airsyncbase:DisplayName>ActiveSyncClient_
                AcceptingMeetingRequest.JPG</airsyncbase:DisplayName>
                <airsyncbase:FileReference>7%3a1%3a0
                </airsyncbase:FileReference>
                <airsyncbase:Method>1</airsyncbase:Method>
                <airsyncbase:EstimatedDataSize>18790
                </airsyncbase:EstimatedDataSize>
              </airsyncbase:Attachment>
```

```
            </airsyncbase:Attachments>
            <airsyncbase:Body>
              <airsyncbase:Type>2</airsyncbase:Type>
              <airsyncbase:EstimatedDataSize>58
              </airsyncbase:EstimatedDataSize>
              <airsyncbase:Truncated>1</airsyncbase:Truncated>
              <airsyncbase:Data>&lt;html&gt;</airsyncbase:Data>
            </airsyncbase:Body>
            <email:MessageClass>IPM.Note</email:MessageClass>
            <email:InternetCPID>28591</email:InternetCPID>
            <email:Flag />
            <email:ContentClass>urn:content-classes:message
            </email:ContentClass>
            <airsyncbase:NativeBodyType>1</airsyncbase:NativeBodyType>
          </ApplicationData>
        </Add>
      </Commands>
    </Collection>
  </Collections>
</Sync>
```

4.10.4.3   ItemOperation Request

The following example shows the **ItemOperations** command request (section 2.2.2.8) sent from the client to the server to retrieve the attachment.

```
<?xml version="1.0" encoding="utf-8"?>
<ItemOperations xmlns:airsyncbase="AirSyncBase"
xmlns="ItemOperations">
  <Fetch>
    <Store>Mailbox</Store>
      <airsyncbase:FileReference>7%3a1%3a0</airsyncbase:FileReference>
  </Fetch>
</ItemOperations>
```

4.10.4.4   ItemOperation Response

The following example shows the **ItemOperations** command response (section 2.2.2.8) sent from the server to the client.

```
<?xml version="1.0" encoding="utf-8"?>
<ItemOperations xmlns:airsyncbase="AirSyncBase"
xmlns="ItemOperations">
  <Status>1</Status>
  <Response>
    <Fetch>
      <Status>1</Status>
      <airsyncbase:FileReference>7%3a1%3a0</airsyncbase:FileReference>
      <Properties>
        <airsyncbase:ContentType>text/plain
        </airsyncbase:ContentType>
        <Data>U291cmNlIERlcG90IFN5c3RlbSBSZXF1aXJlbWVudHMNClRv
        IHJ1biBTb3VyY2UgRGVwb3QsIHlvdXIgY29tcHV0ZXIgbXVzdCBt
        WV0IGNlcnRhaW4gaGFyZHdhcmUgYW5kIHNvZnR3YXJlIHJlcXVpc
        mVtZW50cy4gVG8gcnVuIHRoZSBTb3VyY2UgRGVwb3QgY2xpZW50L
```

[MS-ASCMD] — v20140130
Exchange ActiveSync: Command Reference Protocol

Copyright © 2014 Microsoft Corporation.

Release: February 10, 2014

```
        CBjaGVjayB0aGUgZm9sbG93aW5GGxpc3Qgb2YgbWluaW1bSBw
        bGF0Zm9ybS1zcGVjaWaWZpYyByZXF1aXJlbWVudHMgZm9yIFNvdXJj
        ZSBEZXBvdC4gDQoNClNvZnR3YXJlIFJlcXVpcmVtZW50cw0KV2luZG9
        3cyBYUCBhbmQgV2luZG93cyAyMDAwIJYgSW50ZWWwb3IgQVhQ
        NjQNCk5UNCAtIEludGVsIGVsIA0KV2luZG93cyA5eCCWIEludGVsDQ
        pIYXJkd2FyZSBSZXF1aXJlbWVudHMNCkludGVsIENsaWVudA0K
        DQp4ODYgcHJvY2Vzc29yDQpQZW50aXVtIG9yIGJldHRlciANCkV
        ub3VnaCBoYXJkIGRpc2sgc3BhY2UgdG8gaG9sZCB0aGUgZmlsZX
        MNCkFYDY0IENsaWVudA0KDQpBbHBoYSBDbGllbnQgc3lzdGVtI
        HJ1bm5pbmcgYW4gQVhQNjQgZW11bGF0b3IuIA0KSG93IHRvDQoN
        Ckluc3RhbGwgU291cmNlIERlcG90DQoNCg0KLS0tLS0tLS0tLS0t
        LS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0
        tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0NCg0KU2VuZCB
        1cyBmZWVkYmFjayBhbmQgc3VnZ2VzdGlvbnMgb24gdGhpcyBZW
        xwIHRvcGljIG9yIHRoaXMgcHJvZHVjdC4NCg0KQ29udGFjdCBzdX
        Bwb3J0IGZvciBhc3Npc3RhbmNlLg0KDQoNCmMuIDIwMDQuIE1pY3
        Jvc29mdCBDb25maWRlbnRpYWwgDQpMYXN0IHVwZGF0ZWQgMy8xO
        S8yMDA0IA==</Data>
      </Properties>
    </Fetch>
  </Response>
</ItemOperations>
```

4.10.5 Fetching an Email Item With a Multipart Response

The following sections show how the client requests a multipart response (as described in section 2.2.2.8.1) to the **ItemOperations** command (as described in section 2.2.2.8).

4.10.5.1 Request

The following example shows the **ItemOperations** command request (as described in section 2.2.2.8) sent from the client to the server to retrieve an email message. The request contains the MS-ASAcceptMultiPart header, described in [MS-ASHTTP] section 2.2.1.1.2.3, set to "T", indicating that the client will accept a multipart response.

```
POST /Microsoft-Server-ActiveSync?Cmd=ItemOperations
  &User=deviceuser&DeviceId=v140Device&DeviceType=PocketPC
Content-Type: application/vnd.ms-sync
MS-ASProtocolVersion: 14.1
MS-ASAcceptMultiPart: T

<?xml version="1.0" encoding="utf-8"?>
<ItemOperations xmlns:airsync="AirSync"
xmlns:airsyncbase="AirSyncBase" xmlns="ItemOperations">
  <Fetch>
    <Store>Mailbox</Store>
    <airsync:CollectionId>5</airsync:CollectionId>
    <airsync:ServerId>5:1</airsync:ServerId>
    <Options>
      <airsyncbase:BodyPreference>
        <airsyncbase:Type>1</airsyncbase:Type>
        <airsyncbase:TruncationSize>5120</airsyncbase:TruncationSize>
        <airsyncbase:AllOrNone>0</airsyncbase:AllOrNone>
      </airsyncbase:BodyPreference>
    </Options>
  </Fetch>
```

```
</ItemOperations>
```

4.10.5.2 Response

The following example shows the multipart **ItemOperations** command response (as described in section 2.2.2.8.1) sent from the server to the client.

```
HTTP/1.1 200 OK
Content-Type: application/vnd.ms-sync.multipart
```

```
0200000014000000F401000080200001300000003016A000014454D033100014E464D033100010000520335000014
D03353A3100015003456D61696C000100144B000256032244657669636520557365722220203C646576696365757365
7240636F6E746F736F2E636F6D3E00015803224465766963652055736572222203C646576696365757365724240636F6
E746F736F2E636F6D3E0001540354657374204D756C746970617274205265737074706F6E736500014F03323031322D30
352D30385432303A31353A34352E3231315A00015103446576696365520557365720001750354657374204D756C6746
9706172742052657370706F6E73650001520331000155033000010011E4F5003746573742E7478740001510352674141
41414143333064496F6B2532624A685461696925 04766417173414B427744326C516E4B2532666262415149526455
74230396B253262484141414141414156414144326C516E4B2532666262415149526455774230396B253262484141
4634714125326253414141414A25336130000152033100014C033733000101014A46033100014C03313900010014510
3310001010002530349504D2E4E6F74650001790332383539310013A7C0375726E3A636F6E74656E742D636C6173
7365733A6573737361676500010011560332000100164 9C310ADA57BDA90934B4CAE83B1A1AFDDEAD5014AC305CD2
D5758360101010101546869732069732074686520626F6479 2E0D0A
```

Interpreting the binary body of the response as a **MultiPartResponse** structure (as described in section 2.2.2.8.1.1) results in the following:

PartCount (4 bytes): 02000000 = 2

PartsMetaData[0] (8 bytes): 14000000F4010000 = **Offset**: 14000000 = 20, **Length**: F4010000 = 500

PartsMetaData[1] (8 bytes): 0802000013000000 = **Offset:** 08020000 = 520, **Length:** 13000000 = 19

The rest of the body makes up the **Parts** field of the **MultiPartResponse** structure.

The first part, described by the first **PartMetaData** structure (as described in section 2.2.2.8.1.1.1) in the **PartsMetaData** field, contains the WBXML response. The data starts at 20 bytes from the start of the body, and is 500 bytes long.

```
03016A000014454D033100014E464D033100010000520335000014D03353A3100015003456D61696C000100144B000
2560322446576696365205573657222203C646576696365757365724240636F6E746F736F2E636F6D3E0001580322446
57669636520557365722220203C646576696365757365724240636F6E746F736F2E636F6D3E0001540354657374204D7
56C746970617274205265737074706F6E736500014F03323031322D30352D30385432303A31353A34352E3231315A0001
510344657669636520557365720001750354657374204D756C746970617274205265737074706F6E73650001520331000
1550330000100114E4F5003746573742E7478740001510352674141414143333064496F6B2532624A685461696250
04766417173414B427744326C516E4B2532666262415149526455774230396B2532624841414141414141564141443
26C516E4B2532666262415149526455774230396B25326248414146634714125326253414141414A25336130000152033
3100014C033733000101014A46033100014C03313900010014510331000101000253034950 4D2E4E6F7465000179 0
332383539310013A7C0375726E3A636F6E74656E742D636C6173 7365733A6573737361676500010011560332000 1
001649C310ADA57BDA90934B4CAE83B1A1AFDDEAD5014AC305CD2D5758360101010101
```

This decodes into the following XML:

```
<?xml version="1.0" encoding="utf-8"?>
<ItemOperations xmlns="ItemOperations" xmlns:airsync="AirSync" xmlns:email="Email"
```

```
  xmlns:airsyncbase="AirSyncBase" xmlns:email2="Email2">
  <Status>1</Status>
  <Response>
    <Fetch>
      <Status>1</Status>
      <airsync:CollectionId>5</airsync:CollectionId>
      <airsync:ServerId>5:1</airsync:ServerId>
      <airsync:Class>Email</airsync:Class>
      <Properties>
        <email:To>"Device User" &lt;deviceuser@contoso.com&gt;</email:To>
        <email:From>"Device User" &lt;deviceuser@contoso.com&gt;</email:From>
        <email:Subject>Test Multipart Response</email:Subject>
        <email:DateReceived>2012-05-08T20:15:45.211Z</email:DateReceived>
        <email:DisplayTo>Device User</email:DisplayTo>
        <email:ThreadTopic>Test Multipart Response</email:ThreadTopic>
        <email:Importance>1</email:Importance>
        <email:Read>0</email:Read>
        <airsyncbase:Attachments>
          <airsyncbase:Attachment>
            <airsyncbase:DisplayName>test.txt</airsyncbase:DisplayName>
            <airsyncbase:FileReference>RgAAAAC30dIok%2bJhTaibPGfAqsAKBwD2lQnK
              %2fbbAQIRdUwB09k%2bHAAAAAAAVAAD2lQnK%2fbbAQIRdUwB09k%2bHAAF4qA
              %2bSAAAJ%3a0</airsyncbase:FileReference>
            <airsyncbase:Method>1</airsyncbase:Method>
            <airsyncbase:EstimatedDataSize>73</airsyncbase:EstimatedDataSize>
          </airsyncbase:Attachment>
        </airsyncbase:Attachments>
        <airsyncbase:Body>
          <airsyncbase:Type>1</airsyncbase:Type>
          <airsyncbase:EstimatedDataSize>19</airsyncbase:EstimatedDataSize>
          <Part>1</Part>
        </airsyncbase:Body>
        <email:MessageClass>IPM.Note</email:MessageClass>
        <email:InternetCPID>28591</email:InternetCPID>
        <email:Flag />
        <email:ContentClass>urn:content-classes:message</email:ContentClass>
        <airsyncbase:NativeBodyType>2</airsyncbase:NativeBodyType>
        <email2:ConversationId>
          <![CDATA[ADA57BDA90934B4CAE83B1A1AFDDEAD5]]>
        </email2:ConversationId>
        <email2:ConversationIndex><![CDATA[CD2D575836]]></email2:ConversationIndex>
      </Properties>
    </Fetch>
  </Response>
</ItemOperations>
```

The **itemoperations:Part** element (described in section 2.2.3.120) is present as a child of the **airsyncbase:Body** element (described in [MS-ASAIRS] section 2.2.2.4). This indicates that the part described by the second **PartMetaData** structure in the **PartsMetaData** field contains the data for the body of the email item. The **airsyncbase:Body** element has an **airsyncbase:Type** child element (described in [MS-ASAIRS] section 2.2.2.22.1) with a value of 1, which indicates that the body is plain text.

The data starts at 520 bytes from the start of the body, and is 19 bytes long.

```
5468697320697320746865206F64792E0D0A
```

Interpreting this data as plain text results in the following value:

```
This is the body.CRLF
```

4.11   Searching for an Item in the Mailbox by Using the Search Command

The following sections show how the client performs keyword searches by using the **Search** command (section 2.2.2.14) and forwards search results for items in the mailbox by using the **SmartForward** command (section 2.2.2.17).

4.11.1   Keyword Search

In the following example, the client searches the Inbox folder in the mailbox by using the keyword "Presentation". The client requests the first five results and requests that text bodies be returned for the results. Note that the content of the **search:FreeText** element (section 2.2.3.73) is not case-sensitive.

4.11.1.1   Request

The following example shows the **Search** command request (section 2.2.2.14) sent from the client to the server to search the Inbox folder by using the keyword "Presentation".

```
<?xml version="1.0" encoding="utf-8"?>
<Search xmlns="Search" xmlns:airsync="AirSync">
<Store>
  <Name>Mailbox</Name>
    <Query>
      <And>
        <airsync:CollectionId>7</airsync:CollectionId>
        <FreeText>Presentation</FreeText>
      </And>
    </Query>
    <Options>
      <RebuildResults />
      <Range>0-4</Range>
      <DeepTraversal/>
    </Options>
  </Store>
</Search>
```

4.11.1.2   Response

The following example shows the **Search** command response (section 2.2.2.14) sent from the server to the client.

```
<?xml version="1.0" encoding="utf-8"?><Search xmlns:airsync="AirSync"
xmlns:email="Email" xmlns:airsyncbase="AirSyncBase"
xmlns:email2="Email2" xmlns="Search">
<Status>1</Status>
  <Response>
    <Store>
      <Status>1</Status>
      <Result>
        <airsync:Class>Email</airsync:Class>
```

```
                    <LongId>RgAAAACYWCHnyBZ%2fTq8bujFmR1EPBwBzyWfENpcEQ7
                    zUNyaWwM4BAAAA8FxEAABzyWfENpcEQ7zUNyaWwM4BAAAA8HACAAAJ</LongId>
                    <airsync:CollectionId>7</airsync:CollectionId>
                    <Properties>
                      <email:To>"deviceuser" &lt;anat@contoso.com&gt;
                      </email:To>
                      <email:From>"deviceuser2"&lt;chris@contoso.com&gt;
                      </email:From>
                      <email:Subject>Presentation</email:Subject>
                      <email:DateReceived>2007-05-08T17:41:58.000Z
                      </email:DateReceived>
                      <email:DisplayTo>DeviceUserDisplayName</email:DisplayTo>
                      <email:ThreadTopic>Presentation</email:ThreadTopic>
                      <email:Importance>1</email:Importance>
                      <email:Read>1</email:Read>
                      <airsyncbase:Body>
                        <airsyncbase:Type>1</airsyncbase:Type>
                        <airsyncbase:EstimatedDataSize>6
                        </airsyncbase:EstimatedDataSize>
                        <airsyncbase:Truncated>1</airsyncbase:Truncated>
                      </airsyncbase:Body>
                      <email:MessageClass>IPM.Note</email:MessageClass>
                      <email:InternetCPID>28591</email:InternetCPID>
                      <email:Flag />
                      <email:ContentClass>urn:content-classes:message
                      </email:ContentClass>
                      <airsyncbase:NativeBodyType>1</airsyncbase:NativeBodyType>
                      <email2:ConversationId>Í÷Žo•7D´½•¹t (îD</email2:ConversationId>
                      <email2:ConversationIndex>Ë&#x1E;Õ~%</email2:ConversationIndex>
                      <email2:LastVerbExecuted>3</email2:LastVerbExecuted>
                      <email2:LastVerbExecutionTime>2010-07-
                 12T19:15:25.446Z</email2:LastVerbExecutionTime>
                      <email:Categories/>
                    </Properties>
                  </Result>
                  <Range>0-0</Range>
                  <Total>1</Total>
                </Store>
              </Response>
            </Search>
```

4.11.2   Forward a Search Result

In the following example, the client forwards an item in the search results by using the
composemail:LongId value (section 2.2.3.89.4) of the item.

4.11.2.1   Request

The following example shows the **SmartForward** command request (section 2.2.2.17) sent from
the client to the server to forward an item in the search results. The **composemail:LongId** element
value (section 2.2.3.89.4) identifies the item being forwarded.

```
<?xml version="1.0" encoding="utf-8"?>
<SmartForward xmlns="ComposeMail">
  <ClientId>634145285982398784</ClientId>
  <Source>
```

```
<LongId>RgAAAAA%2fygDboRrKQ6odHngUY8KtBwCwZCmLZplDQJqLAuQKTJwjAAAAjGzeAACwZCmLZplDQJqLAuQKTJw
jAAAAjmvbAAAJ</LongId>
  </Source>
  <Mime>From: anat@contoso.com
To: chris@contoso.com
Cc:
Bcc:
Subject: Presentation
MIME-Version: 1.0
Content-Type: text/plain; charset="iso-8859-1"
Content-Transfer-Encoding: 7bit
X-MimeOLE: Produced By Microsoft MimeOLE V6.00.2900.3350
This is the content of the forwarded message.</Mime>
</SmartForward>
```

4.11.2.2 Response

The following example shows the **SmartForward** command response (section 2.2.2.17) sent from the server to the client. The response indicates that the **SmartForward** command request was successful.

```
HTTP/1.1 200 OK
Date: Mon, 12 Jul 2010 19:15:26 GMT
Content-Length: 0
```

4.11.3 Keyword Search with MIMESupport

In the following example, the client searches the Inbox folder in the mailbox by using the keyword "text". The client requests MIME data for all results.

4.11.3.1 Request

The following example shows the **Search** command request (section 2.2.2.14) sent from the client to the server to search the Inbox folder by using the keyword "text".

```
<?xml version="1.0" encoding="utf-8"?>
<Search xmlns="Search" xmlns:airsyncbase="AirSyncBase" xmlns:airsync="AirSync">
   <Store>
      <Name>Mailbox</Name>
      <Query>
         <And>
            <airsync:Class>Email</airsync:Class>
            <FreeText>text</FreeText>
         </And>
      </Query>
      <Options>
         <RebuildResults/>
         <DeepTraversal/>
         <Range>0-999</Range>
         <airsyncbase:BodyPreference>
            <airsyncbase:Type>4</airsyncbase:Type>
            <airsyncbase:TruncationSize>1024</airsyncbase:TruncationSize>
            <airsyncbase:Preview>120</airsyncbase:Preview>
         </airsyncbase:BodyPreference>
```

```
        <airsync:MIMESupport>2</airsync:MIMESupport>
      </Options>
    </Store>
  </Search>
```

4.11.3.2 Response

The following example shows the **Search** command response (section 2.2.2.14) sent from the server to the client.

```
<?xml version="1.0" encoding="utf-8"?>
<Search xmlns:airsync="AirSync" xmlns:email="Email" xmlns:airsyncbase="AirSyncBase"
xmlns="Search">
    <Status>1</Status>
    <Response>
        <Store>
            <Status>1</Status>
            <Result>
                <airsync:Class>Email</airsync:Class>

<LongId>RgAAAAaty%2f%2b4QxHTJOZnIR0P9qkBwA6pk60fqkEQbWH4Wm%2bnjh7AJKAUQo6AAA6pk60fqkEQbWH4Wm
%2bnjh7AJKAURrEAAAJ</LongId>
                <airsync:CollectionId>6</airsync:CollectionId>
                <Properties>
                    <email:To>"Chris Gray" &lt;chris@contoso.com></email:To>
                    <email:From>"Jan Kotas" &lt;jan@contoso.com></email:From>
                    <email:Subject>Subject</email:Subject>
                    <email:DateReceived>2007-04-02T19:20:32.000Z</email:DateReceived>
                    <email:DisplayTo>Chris Gray</email:DisplayTo>
                    <email:Read>1</email:Read>
                    <airsyncbase:Body>
                        <airsyncbase:Type>4</airsyncbase:Type>
                        <airsyncbase:Preview>The beginning of this message</airsyncbase:Preview>
                        <airsyncbase:EstimatedDataSize>2288</airsyncbase:EstimatedDataSize>
                        <airsyncbase:Truncated>1</airsyncbase:Truncated>
                        <airsyncbase:Data>Received: from 157.55.97.120 ([157.55.97.120]) by
contoso.com ([157.55.97.121]) with Microsoft Exchange Server HTTP-DAV ; Mon, 2 Apr 2007
19:20:32 +0000 From: Jan Kotas &lt;jan@contoso.com> To: Chris Gray &lt;chris@contoso.com>
Content-Class: urn:content-classes:message Date: Mon, 27 Apr 1998 13:05:29 -0700 Subject:
Subject Thread-Topic: Topic Message-ID:
&lt;3AA64EB47EA90441B587E169BE9E387B9280511AC4@contoso.com> Accept-Language: en-US X-MS-Has-
Attach: X-MS-TNEF-Correlator: Content-Type: text/plain; charset="iso-8859-1" Content-
Transfer-Encoding: quoted-printable MIME-Version: 1.0
Body12345678901234567890123456789012345678901234567890123456789012345678901=
234567890123456789012345678901234567890123456789012345678901234567890123456=
789012345678901234567890123456789012345678901234567890123456789012345678901=
234567890123456789012345678901234567890123456789012345678901234567890123456789</airsyncbase:Data>
                    </airsyncbase:Body>
                    <email:MessageClass>IPM.Note</email:MessageClass>
                    <email:InternetCPID>28591</email:InternetCPID>
                    <email:Flag/>
                    <email:ContentClass>urn:content-classes:message</email:ContentClass>
                    <airsyncbase:NativeBodyType>1</airsyncbase:NativeBodyType>
                </Properties>
            </Result>
            <Range>0-0</Range>
            <Total>1</Total>
        </Store>
```

```
        </Response>
    </Search>
```

4.12   Searching the Global Address List by Using the Search Command

The following sections show how to search the GAL by using the **Search** command (section 2.2.2.14).

4.12.1   Request

The following example shows a **Search** command request (section 2.2.2.14) sent from the client to the server. The **search:Name** element (section 2.2.3.110.2) identifies the GAL as the store to search, and the **search:Query** element (section 2.2.3.129) identifies the search string as "Anat". Within the **search:Options** element (section 2.2.3.115.4), the **search:Range** element (section 2.2.3.130.2) indicates that a maximum of two results can be returned in the response, the **search:RebuildResults** element (section 2.2.3.131) indicates that the server will rebuild the search folder (2), and the **search:DeepTraversal** element (section 2.2.3.41) indicates that all subfolders are searched.

```
<?xml version="1.0" encoding="utf-8"?>
<Search xmlns="Search">
  <Store>
    <Name>GAL</Name>
    <Query>Anat</Query>
    <Options>
        <Range>0-1</Range>
        <RebuildResults/>
        <DeepTraversal/>
    </Options>
  </Store>
</Search>
```

4.12.2   Response

The following example shows the **Search** command response (section 2.2.2.14) sent from the server to the client. A **search:Status** element (section 2.2.3.162.12) value of 1 is returned to indicate that the search was successful. The **search:Result** elements (section 2.2.3.142.2) contain the GAL entries for the first two results that met the search criteria.

```
<?xml version="1.0" encoding="utf-8"?>
<Search xmlns:gal="Gal" xmlns="Search">
  <Status>1</Status>
  <Response>
    <Store>
      <Status>1</Status>
      <Result>
        <Properties>
          <gal:DisplayName>Anat Kerry</gal:DisplayName>
          <gal:Phone>+1 (301) 5550156 X8376</gal:Phone>
          <gal:Office>Bldg36/6163</gal:Office>
          <gal:Title>SDE</gal:Title>
          <gal:Company>Contoso</gal:Company>
          <gal:Alias>anatk</gal:Alias>
          <gal:FirstName>Anat</gal:FirstName>
```

```
              <gal:LastName>Kerry</gal:LastName>
              <gal:EmailAddress>anatk@contoso.com</gal:EmailAddress>
            </Properties>
          </Result>
          <Result>
            <Properties>
              <gal:DisplayName>Anat Gray</gal:DisplayName>
              <gal:Phone>+1 (953) 5550195 </gal:Phone>
              <gal:Office>Bldg2/2710</gal:Office>
              <gal:Title>SDET</gal:Title>
              <gal:Company>Contoso</gal:Company>
              <gal:Alias>anatg</gal:Alias>
              <gal:FirstName>Anat</gal:FirstName>
              <gal:LastName>Gray</gal:LastName>
              <gal:MobilePhone>+1 (953) 5550167</gal:MobilePhone>
              <gal:EmailAddress>anatg@contoso.com</gal:EmailAddress>
            </Properties>
          </Result>
          <Range>0-1</Range>
          <Total>11</Total>
        </Store>
      </Response>
    </Search>
```

4.13   Working with Folders

The following sections show how to create folders by using the **FolderCreate** command (section 2.2.2.2), delete folders by using the **FolderDelete** command (section 2.2.2.3), update folders by using the **FolderUpdate** command (section 2.2.2.5), and empty folder contents by using the **ItemOperations** command (section 2.2.2.8).

4.13.1   Creating Folders by Using the FolderCreate Command

The following sections show how to create a folder by using the **FolderCreate** command (section 2.2.2.2).

4.13.1.1   Request

The following example shows a **FolderCreate** command request (section 2.2.2.2) sent from the client to the server. The **folderhierarchy:SyncKey** element (section 2.2.3.166.1) in the request is set to 1, which is the **folderhierarchy:SyncKey** value returned in the last **FolderSync** response (section 4.4.2). The **folderhierarchy:ParentId** element (section 2.2.3.119.1) indicates that this is a child folder of the Inbox folder, the **folderhierarchy:Type** element (section 2.2.3.170.2) indicates that this is an email folder, and the **folderhierarchy:DisplayName** element (section 2.2.3.47.2) identifies the display name of the folder as "NewFolder".

```
<?xml version="1.0" encoding="utf-8"?>
<FolderCreate xmlns="FolderHierarchy">
  <SyncKey>1</SyncKey>
  <ParentId>5</ParentId>
  <DisplayName>NewFolder</DisplayName>
  <Type>12</Type>
</FolderCreate>
```

4.13.1.2 Response

The following example shows the **FolderCreate** command response (section 2.2.2.2) sent from the server to the client. A **folderhierarchy:Status** element (section 2.2.3.162.2) value of 1 is returned to indicate that the new folder creation was successful. The **folderhierarchy:SyncKey** element (section 2.2.3.166.1) value has been incremented in the response, and this value is to be used in the next **FolderSync** request (section 2.2.2.19), **FolderCreate** request (section 2.2.2.2), **FolderDelete** request (section 2.2.2.3), or **FolderUpdate** request (section 2.2.2.5). The **folderhierarchy:ServerId** element (section 2.2.3.151.1) value indicates the ID of the new folder.

```
<?xml version="1.0" encoding="utf-8"?>
<FolderCreate xmlns="FolderHierarchy">
  <Status>1</Status>
  <SyncKey>2</SyncKey>
  <ServerId>13</ServerId>
</FolderCreate>
```

4.13.2 Deleting Folders by Using the FolderDelete Command

The following sections show how to delete a folder by using the **FolderDelete** command (section 2.2.2.3).

4.13.2.1 Request

The following example shows a **FolderDelete** command request (section 2.2.2.3) that is sent from the client to the server to delete a folder. The **folderhierarchy:SyncKey** element (section 2.2.3.166.1) value in the request is set to 2, which is the **folderhierarchy:SyncKey** element value returned in the last **FolderCreate** response (section 4.13.1.2). The **folderhierarchy:ServerId** element (section 2.2.3.151.2) value identifies which folder to delete.

```
<?xml version="1.0" encoding="utf-8"?>
<FolderDelete xmlns="FolderHierarchy">
  <SyncKey>2</SyncKey>
  <ServerId>13</ServerId>
</FolderDelete>
```

4.13.2.2 Response

The following example shows the **FolderDelete** command response (section 2.2.2.3) sent from the server to the client. A **folderhierarchy:Status** element (section 2.2.3.162.3) value of 1 is returned to indicate that the folder deletion was successful. The **folderhierarchy:SyncKey** element (section 2.2.3.166.1) value has been incremented in the response, and this value is to be used in the next **FolderSync** request (section 2.2.2.19), **FolderCreate** request (section 2.2.2.2), **FolderDelete** request, or **FolderUpdate** request (section 2.2.2.5).

```
<?xml version="1.0" encoding="utf-8"?>
<FolderDelete xmlns="FolderHierarchy">
  <Status>1</Status>
  <SyncKey>3</SyncKey>
</FolderDelete>
```

4.13.3   Updating Folders by Using the FolderUpdate Command

The following sections show how to update a folder by using the **FolderUpdate** command (section 2.2.2.5).

4.13.3.1   Request

The following example shows a **FolderUpdate** command request (section 2.2.2.5) sent from the client to the server to update a folder. This request is changing the display name of the folder to "NewName". The **folderhierarchy:SyncKey** element (section 2.2.3.166.1) value in the request is set to 3, which is the **folderhierarchy:SyncKey** element value returned in the last **FolderSync** response (section 4.13.2.2). The **folderhierarchy:ServerId** element (section 2.2.3.151.4) indicates the ID of the folder whose name is being updated, and the **folderhierarchy:ParentID** element (section 2.2.3.119.3) indicates the ID of the parent folder.

```
<?xml version="1.0" encoding="utf-8"?>
<FolderUpdate xmlns="FolderHierarchy">
  <SyncKey>3</SyncKey>
  <ServerId>14</ServerId>
  <ParentId>5</ParentId>
  <DisplayName>NewName</DisplayName>
</FolderUpdate>
```

4.13.3.2   Response

The following example shows the **FolderUpdate** command response (section 2.2.2.5) sent from the server to the client. A **folderhierarchy:Status** element (section 2.2.3.162.5) element value of 1 is returned to indicate that the folder update was successful. The **folderhierarchy:SyncKey** element (section 2.2.3.166.1) value has been incremented in the response, and this value is to be used in the next **FolderSync** request (section 2.2.2.4), **FolderCreate** request (section 2.2.2.2), **FolderDelete** request (section 2.2.2.3), or **FolderUpdate** request.

```
<?xml version="1.0" encoding="utf-8"?>
<FolderUpdate xmlns="FolderHierarchy">
  <Status>1</Status>
  <SyncKey>4</SyncKey>
</FolderUpdate>
```

4.13.4   Emptying Folder Contents by Using the ItemOperations Command

The following sections show how to empty the contents of a folder by using the **ItemOperations** command (section 2.2.2.8).

4.13.4.1   Request

The following example shows an **ItemOperations** command request (section 2.2.2.8) sent from the client to the server to delete the contents of a folder. The **itemoperations:EmptyFolderContents** element (section 2.2.3.55) contains the **airsync:CollectionId** element (section 2.2.3.30.2) value that identifies the folder whose contents are deleted.

```
<?xml version="1.0" encoding="utf-8"?>
<ItemOperations xmlns:airsync="AirSync" xmlns="ItemOperations">
  <EmptyFolderContents>
```

```
        <airsync:CollectionId>15</airsync:CollectionId>
    </EmptyFolderContents>
  </ItemOperations>
```

4.13.4.2 Response

The following example shows the **ItemOperations** command response (section 2.2.2.8) sent from
the server to the client. An **itemoperations:Status** element (section 2.2.3.162.7) value of 1 is
returned to indicate that the folder deletion was successful. The
itemoperations:EmptyFolderContents element (section 2.2.3.55) and **airsync:CollectionId**
element (section 2.2.3.30.2) provide confirmation of the folder whose contents were deleted.

```
<?xml version="1.0" encoding="utf-8"?>
<ItemOperations xmlns:airsync="AirSync" xmlns="ItemOperations">
  <Status>1</Status>
  <Response>
    <EmptyFolderContents>
      <Status>1</Status>
      <airsync:CollectionId>15</airsync:CollectionId>
    </EmptyFolderContents>
  </Response>
</ItemOperations>
```

4.14   Moving Items to Another Folder by Using the MoveItems Command

The following sections show how to move items to another folder by using the **MoveItems**
command (section 2.2.2.10).

4.14.1   Request

The following example shows a **MoveItems** command request (section 2.2.2.10) sent from the
client to the server to move an item from one folder to another. The **move:SrcMsgId** element
(section 2.2.3.160) identifies the item to move. The **move:SrcFldId** element (section 2.2.3.159)
identifies the folder ID of the folder that currently contains the item, and the **move:DstFldId**
element (section 2.2.3.49.2) identifies the destination folder.

```
<?xml version="1.0" encoding="utf-8"?>
<MoveItems xmlns="Move">
    <Move>
        <SrcMsgId>5:1</SrcMsgId>
        <SrcFldId>5</SrcFldId>
        <DstFldId>14</DstFldId>
    </Move>
</MoveItems>
```

4.14.2   Response

The following example shows the **MoveItems** command response (section 2.2.2.10) sent from the
server to the client. A **move:Status** element (section 2.2.3.162.9) value of 3 is returned to indicate
that the move operation was successful. The **move:SrcMsgId** element (section 2.2.3.160)
identifies the original ID of the item to move, and the **move:DstMsgId** element (section 2.2.3.50)
identifies the new ID of the item that was moved.

```
<?xml version="1.0" encoding="utf-8"?>
<MoveItems xmlns="Move">
  <Response>
    <SrcMsgId>5:1</SrcMsgId>
    <Status>3</Status>
    <DstMsgId>14:1</DstMsgId>
  </Response>
</MoveItems>
```

4.15   Creating Meetings

When a user creates an appointment or meeting on the client, the calendar item is added to the server by using the **Sync** command (section 2.2.2.19). If the meeting has attendees, the client uses the **SendMail** command (section 2.2.2.15) to send meeting requests to the attendees. When the attendee synchronizes her Inbox folder, the **Sync** response (section 2.2.2.19) from the server adds the new meeting request to the attendee's Inbox folder. When the attendee synchronizes their Calendar folder, the **Sync** response from the server adds the new calendar item to the attendee's Calendar folder.

The following sections contain examples that show how a new meeting is uploaded to the server, how the client sends the meeting request, how the new meeting request is added to the attendees Inbox folder, and how the new meeting is added to the attendees Calendar folder.

The **Sync** request command used to create an appointment is the same as the **Sync** request used to create a meeting, except that when creating an appointment, the initial calendar item uploaded to the server does not have any attendees.

4.15.1   Uploading a Meeting to the Server

The following sections show how to upload a meeting to the server by using the **Sync** command (section 2.2.2.19).

4.15.1.1   Request

The following example shows a **Sync** command request (section 2.2.2.19) sent from the client to the server to upload a meeting to the Calendar folder on the server. The **Sync** request contains a new meeting that has one attendee.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:calendar="Calendar" xmlns="AirSync">
    <Collections>
        <Collection>
            <SyncKey>509910681</SyncKey>
            <CollectionId>1</CollectionId>
            <GetChanges>1</GetChanges>
            <Commands>
                <Add>
                    <ClientId>1</ClientId>
                    <ApplicationData>

<calendar:TimeZone>4AEAACgARwBNAFQALQAwADgAOgAwADAAKQAgAFAAYQBjAGkAZgBpAGMAIABUAGkAbQBlACAAKA
BVAFMAIAAmACAAQwAAAAsAAAABAAIAAAAAAAAAAAAAAACgARwBNAFQALQAwADgAOgAwADAAKQAgAFAAYQBjAGkAZgBpAGM
AIABUAGkAbQBlACAAKABVAFMAIAAmACAAQwAAAMAAAACAAIAAAAAAAAxP///w==</calendar:TimeZone>
                    <calendar:DtStamp>20100406T170219Z</calendar:DtStamp>
                    <calendar:StartTime>20100503T090000Z</calendar:StartTime>
                    <calendar:Subject>TestMeeting</calendar:Subject>
```

```
<calendar:UID>040000008200E00074C5B7101A82E0080000000036BD76EAAAD5CA01000000000000000001000000
0C45185F686A5D542B20BF2CE2F477D55</calendar:UID>
                              <calendar:Attendees>
                                  <calendar:Attendee>
                                      <calendar:Email>testuser2@contoso.com</calendar:Email>
                                      <calendar:Name>Test User 2</calendar:Name>
                                      <calendar:AttendeeStatus>0</calendar:AttendeeStatus>
                                      <calendar:AttendeeType>1</calendar:AttendeeType>
                                  </calendar:Attendee>
                              </calendar:Attendees>
                              <calendar:Location>My Office</calendar:Location>
                              <calendar:EndTime>20100503T100000Z</calendar:EndTime>
                              <calendar:Sensitivity>0</calendar:Sensitivity>
                              <calendar:BusyStatus>1</calendar:BusyStatus>
                              <calendar:AllDayEvent>0</calendar:AllDayEvent>
                          </ApplicationData>
                      </Add>
                  </Commands>
              </Collection>
          </Collections>
      </Sync>
```

4.15.1.2 Response

The following example shows a **Sync** command response (section 2.2.2.19) sent from the server to the client for the Calendar folder. The **airsync:SyncKey** element (section 2.2.3.166.4) value is incremented and a **airsync:Status** element (section 2.2.3.162.16) value of 1 is returned to indicate that the **Sync** command was successful. The **airsync:ServerId** element (section 2.2.3.151.6) value is the ID for the new meeting.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync">
  <Collections>
    <Collection>
      <SyncKey>993351228</SyncKey>
      <CollectionId>1</CollectionId>
      <Status>1</Status>
      <Responses>
        <Add>
          <ClientId>1</ClientId>
          <ServerId>1:3</ServerId>
          <Status>1</Status>
        </Add>
      </Responses>
    </Collection>
  </Collections>
</Sync>
```

4.15.2 Sending Meeting Requests

The following sections show how to send a meeting request to an attendee's Inbox folder by using the **SendMail** command (section 2.2.2.15).

4.15.2.1 Request

The following example shows a **SendMail** command request (section 2.2.2.15) sent from the meeting organizer's client to the server. The **SendMail** command request contains the meeting request for the new calendar item uploaded to the server in section 4.15.1.1.

As described in [RFC2447] section 3.4, meeting requests have a content type of "text/calendar" with the *method* parameter set to "REQUEST". The details of the meeting request are sent in iCalendar format, as described in [MS-OXCICAL].

```
<?xml version="1.0" encoding="utf-8"?>
<SendMail xmlns="ComposeMail">
<ClientId>3</ClientId>
<SaveInSentItems/>
<Mime>X-MimeOLE: Produced By Microsoft Exchange V6.5.7226.0
Received: by contoso.com
     id &lt;01C4F431.619431CA@contoso.com&gt;; Tue, 6 Apr 2010 14:52:31 -0800
MIME-Version: 1.0
Content-Type: multipart/alternative;
     boundary="----_=_NextPart_001_01C4F431.619431CA"
Content-class: urn:content-classes:calendarmessage
Subject: TestMeeting
Date: Tue, 6 Apr 2010 17:02:19 -0800
Message-ID: &lt;1E1FC9DA1767ED44872F4E17AC17E8F7011CF265@contoso.com&gt;
X-MS-Has-Attach:
X-MS-TNEF-Correlator:
Thread-Topic: dD)g_&lt;X`WS
Thread-Index: AcT0MWkdH8lvfizmRrysZICO2nAkVQAAAAFg
From: chris@contoso.com
To: kim@contoso.com

This is a multi-part message in MIME format.

------_=_NextPart_001_01C4F431.619431CA
Content-Type: text/html;
     charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable

&lt;!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN"&gt;
&lt;HTML&gt;
&lt;HEAD&gt;
&lt;META HTTP-EQUIV=3D"Content-Type" CONTENT=3D"text/html; =
charset=3Diso-8859-1"&gt;
&lt;META NAME=3D"Generator" CONTENT=3D"MS Exchange Server version =
6.5.7232.36"&gt;
&lt;TITLE&gt;TestMeeting&lt;/TITLE&gt;
&lt;/HEAD&gt;
&lt;BODY&gt;
&lt;!-- Converted from text/rtf format --&gt;

&lt;P&gt;&lt;B&gt;&lt;FONT SIZE=3D2 FACE=3D"System"&gt;This is a meeting
request.&lt;/FONT&gt;&lt;/B&gt;
&lt;/P&gt;

&lt;/BODY&gt;
&lt;/HTML&gt;
------_=_NextPart_001_01C4F431.619431CA
Content-class: urn:content-classes:calendarmessage
```

Release: February 10, 2014

```
Content-Type: text/calendar;
    method=REQUEST;
    name="meeting.ics"
Content-Transfer-Encoding: 8bit

BEGIN:VCALENDAR
METHOD:REQUEST
PRODID:Microsoft CDO for Microsoft Exchange
VERSION:14.1
BEGIN:VTIMEZONE
TZID:(GMT-08.00) Pacific Time (US &amp; Canada)/Tijuana
X-MICROSOFT-CDO-TZID:13
BEGIN:STANDARD
DTSTART:16010101T020000
TZOFFSETFROM:-0700
TZOFFSETTO:-0800
RRULE:FREQ=YEARLY;WKST=MO;INTERVAL=1;BYMONTH=11;BYDAY=1SU
END:STANDARD
BEGIN:DAYLIGHT
DTSTART:16010101T020000
TZOFFSETFROM:-0800
TZOFFSETTO:-0700
RRULE:FREQ=YEARLY;WKST=MO;INTERVAL=1;BYMONTH=3;BYDAY=2SU
END:DAYLIGHT
END:VTIMEZONE
BEGIN:VEVENT
DTSTAMP:20100406T170219
DTSTART;TZID="(GMT-08.00) Pacific Time (US &amp; Canada)/Tijuana":20100503T090000
SUMMARY:TestMeeting
UID:140000008200E00074C5B7101A82E00800000000E03FFF5AEEF3C401000000000000000
 010000000972B1A80D193D54E8DD185652818A128
ATTENDEE;ROLE=REQ-PARTICIPANT;PARTSTAT=NEEDS-
ACTION;RSVP=TRUE;CN="testuser2":MAILTO:testuser2@contoso.com
ORGANIZER;CN="testuser1":MAILTO:testuser1@contoso.com
LOCATION:My Office\N
DTEND;TZID="(GMT-08.00) Pacific Time (US &amp; Canada)/Tijuana":20100503T100000
DESCRIPTION:Test meeting\N
SEQUENCE:0
PRIORITY:5
CLASS:
CREATED:20100406T170219
LAST-MODIFIED:20100406T170219Z
STATUS:CONFIRMED
TRANSP:OPAQUE
X-MICROSOFT-CDO-BUSYSTATUS:BUSY
X-MICROSOFT-CDO-INSTTYPE:0
X-MICROSOFT-CDO-INTENDEDSTATUS:BUSY
X-MICROSOFT-CDO-ALLDAYEVENT:FALSE
X-MICROSOFT-CDO-IMPORTANCE:1
X-MICROSOFT-CDO-OWNERAPPTID:-2673127979
BEGIN:VALARM
ACTION:DISPLAY
DESCRIPTION:REMINDER
TRIGGER;RELATED=START:-PT00H15M00S
END:VALARM
END:VEVENT
END:VCALENDAR

------_=_NextPart_001_01C4F431.619431CA--</Mime>
```

```
    </SendMail>
```


4.15.2.2   Response

The following example shows a successful **SendMail** command response (section 2.2.2.15) sent from the server to the client. The **SendMail** command response has no XML body (Content-Length: 0) when the **SendMail** command completes successfully.

```
    HTTP/1.1 200 OK
    Date: Tue, 06 Apr 2010 19:11:25 GMT
    Content-Length: 0
```


4.15.3   Adding a Meeting Request to the Attendee's Inbox Folder

The following sections show how to add a meeting request to an attendee's Inbox folder by using the **Sync** command (section 2.2.2.19).

4.15.3.1   Request

The following example shows a **Sync** command request (section 2.2.2.19) for the Inbox folder of the attendee.

```
    <?xml version="1.0" encoding="utf-8"?>
    <Sync xmlns:rm="RightsManagement" xmlns="AirSync">
      <Collections>
        <Collection>
          <SyncKey>52305585</SyncKey>
          <CollectionId>5</CollectionId>
          <DeletesAsMoves>1</DeletesAsMoves>
          <GetChanges>1</GetChanges>
          <WindowSize>512</WindowSize>
          <Options>
            <rm:RightsManagementSupport>0</rm:RightsManagementSupport>
          </Options>
        </Collection>
      </Collections>
    </Sync>
```


4.15.3.2   Response

The following example shows a **Sync** command response (section 2.2.2.19) sent from the server to the client for the Inbox folder of the attendee. The **airsync:SyncKey** element (section 2.2.3.166.4) value is incremented and a **airsync:Status** element (section 2.2.3.162.16) value of 1 is returned to indicate that the **Sync** command was successful. The meeting request is added to the attendee's Inbox folder using the contents of the **airsync:ApplicationData** element (section 2.2.3.11).

```
    <?xml version="1.0" encoding="utf-8"?>
    <Sync xmlns:email="Email" xmlns:email2="Email2" xmlns:airsyncbase="AirSyncBase"
    xmlns="AirSync">
      <Collections>
        <Collection>
          <SyncKey>1744744472</SyncKey>
```

```xml
<CollectionId>5</CollectionId>
<Status>1</Status>
<Commands>
  <Add>
    <ServerId>5:1</ServerId>
    <ApplicationData>
      <email:To>"Anat Kerry" &lt;anat@contoso.com&gt;</email:To>
      <email:From>"Chris Gray" &lt;chris@contoso.com&gt;</email:From>
      <email:Subject>TestMeeting</email:Subject>
      <email:DateReceived>2010-04-06T17:02:26.332Z</email:DateReceived>
      <email:DisplayTo>Anat Kerry</email:DisplayTo>
      <email:ThreadTopic>TestMeeting</email:ThreadTopic>
      <email:Importance>1</email:Importance>
      <email:Read>0</email:Read>
      <airsyncbase:Body>
        <airsyncbase:Type>3</airsyncbase:Type>
        <airsyncbase:EstimatedDataSize>432</airsyncbase:EstimatedDataSize>
        <airsyncbase:Truncated>1</airsyncbase:Truncated>
      </airsyncbase:Body>
      <email:MessageClass>IPM.Schedule.Meeting.Request</email:MessageClass>
      <email:MeetingRequest>
        <email:AllDayEvent>0</email:AllDayEvent>
        <email:StartTime>2010-05-03T16:00:00.000Z</email:StartTime>
        <email:DtStamp>2010-04-06T17:02:19.499Z</email:DtStamp>
        <email:EndTime>2010-05-03T17:00:00.000Z</email:EndTime>
        <email:InstanceType>0</email:InstanceType>
        <email:Location>My Office</email:Location>
        <email:Organizer>"Chris Gray" &lt;chris@contoso.com&gt;</email:Organizer>
        <email:Reminder>900</email:Reminder>
        <email:ResponseRequested>1</email:ResponseRequested>
        <email:Sensitivity>0</email:Sensitivity>
        <email:BusyStatus>2</email:BusyStatus>
```

```xml
        <email2:MeetingMessageType>1</email2:MeetingMessageType>
      </email:MeetingRequest>
      <email:InternetCPID>28591</email:InternetCPID>
      <email:Flag />
      <email:ContentClass>urn:content-classes:calendarmessage</email:ContentClass>
      <airsyncbase:NativeBodyType>3</airsyncbase:NativeBodyType>
      <email2:ConversationId>…</email2:ConversationId>
      <email2:ConversationIndex>…</email2:ConversationIndex>
      <email:Categories />
    </ApplicationData>
  </Add>
</Commands>
</Collection>
</Collections>
</Sync>
```

4.15.4 Adding a Meeting to the Attendee's Calendar Folder

The following sections show how to add a meeting to an attendee's Calendar folder by using the **Sync** command (section 2.2.2.19).

4.15.4.1 Request

The following example shows a **Sync** command request (section 2.2.2.19) sent from the client to the server for the Calendar folder of the attendee.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:rm="RightsManagement" xmlns="AirSync">
  <Collections>
    <Collection>
      <SyncKey>151676483</SyncKey>
      <CollectionId>1</CollectionId>
      <DeletesAsMoves>1</DeletesAsMoves>
      <GetChanges>1</GetChanges>
      <WindowSize>512</WindowSize>
      <Options>
        <rm:RightsManagementSupport>0</rm:RightsManagementSupport>
      </Options>
    </Collection>
  </Collections>
</Sync>
```

4.15.4.2 Response

The following example shows a **Sync** command response (section 2.2.2.19) sent from the server to the client for the Calendar folder of the attendee. The **airsync:SyncKey** element (section 2.2.3.166.4) value is incremented and a **airsync:Status** element (section 2.2.3.162.16) value of 1 is returned to indicate that the **Sync** command was successful. The meeting is added to the attendees Calendar folder using the contents of the **airsync:ApplicationData** element (section 2.2.3.11). Because the attendee has not responded to the meeting request yet, the **calendar:ResponseType** element (as described in [MS-ASCAL] section 2.2.2.38) value is 5, indicating that the meeting request has not been responded to. When the user accepts the meeting request and then synchronizes the Calendar folder again, the **calendar:ResponseType** value will be changed to 3, to indicate that the meeting was accepted.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:calendar="Calendar" xmlns:airsyncbase="AirSyncBase" xmlns="AirSync">
  <Collections>
    <Collection>
      <SyncKey>1761460822</SyncKey>
      <CollectionId>1</CollectionId>
      <Status>1</Status>
      <Commands>
        <Add>
          <ServerId>1:1</ServerId>
          <ApplicationData>

<calendar:TimeZone>4AEAACgARwBNAFQALQAwADgAOgAwADAAKQAgAFAAYQBjAGkAZgBpAGMAIABUAGkAbQBlACAAKA
BVAFMAIAAmACAAQwAAAAsAAAABAAIAAAAAAAAAAAAACgARwBNAFQALQAwADgAOgAwADAAKQAgAFAAYQBjAGkAZgBpAGM
AIABUAGkAbQBlACAAKABVAFMAIAAmACAAQwAAAAMAAAACAAIAAAAAAAAxP///w==</calendar:TimeZone>
            <calendar:DtStamp>20100406T170219Z</calendar:DtStamp>
```

```
                <calendar:StartTime>20100503T090000Z</calendar:StartTime>
                <calendar:Subject>TestMeeting</calendar:Subject>

    <calendar:UID>040000008200E00074C5B7101A82E00800000000122CD5EB1DB7CA0100000000000000000001000000
    0CA93B43442E5314A8AF02D7E133D17EF</calendar:UID>
                <calendar:OrganizerName>Chris Gray</calendar:OrganizerName>
                <calendar:OrganizerEmail>chris@contoso.com</calendar:OrganizerEmail>
                <calendar:Attendees>
                  <calendar:Attendee>
                    <calendar:Email>anat@contoso.com</calendar:Email>
                    <calendar:Name>Anat Kerry</calendar:Name>
                    <calendar:AttendeeType>1</calendar:AttendeeType>
                  </calendar:Attendee>
                </calendar:Attendees>
                <calendar:Location>My Office</calendar:Location>
                <calendar:EndTime>20100503T100000Z</calendar:EndTime>
                <airsyncbase:Body>
                  <airsyncbase:Type>3</airsyncbase:Type>
                  <airsyncbase:EstimatedDataSize>275</airsyncbase:EstimatedDataSize>
                  <airsyncbase:Truncated>1</airsyncbase:Truncated>
                </airsyncbase:Body>
                <calendar:Sensitivity>0</calendar:Sensitivity>
                <calendar:BusyStatus>1</calendar:BusyStatus>
                <calendar:AllDayEvent>0</calendar:AllDayEvent>
                <calendar:Reminder>15</calendar:Reminder>
                <calendar:MeetingStatus>3</calendar:MeetingStatus>
                <airsyncbase:NativeBodyType>3</airsyncbase:NativeBodyType>
                <calendar:ResponseRequested>1</calendar:ResponseRequested>
                <calendar:ResponseType>5</calendar:ResponseType>
              </ApplicationData>
            </Add>
          </Commands>
        </Collection>
      </Collections>
    </Sync>
```

4.16   Responding to Meeting Requests by Using the MeetingResponse Command

The following sections show how to respond to an exception or single instance of a recurring meeting by using the **MeetingResponse** command (section 2.2.2.9) and the **meetingresponse:InstanceId** element (section 2.2.3.83.1).

4.16.1   Request

The following example shows a **MeetingReponse** command request (section 2.2.2.9) sent from the client to the server to respond to two instances of recurring meetings. The first meeting is identified by using the **meetingresponse:CollectionId** element (section 2.2.3.30.3), **meetingresponse:RequestId** element (section 2.2.3.138), and the **meetingresponse:InstanceId** element (section 2.2.3.83.1), while the second meeting is identified by using the **search:LongId** element (section 2.2.3.89) and **meetingresponse:InstanceId** element.

```
<?xml version="1.0" encoding="utf-8"?>
<MeetingResponse xmlns:search="Search" xmlns="MeetingResponse">
  <Request>
```

```
    <UserResponse>2</UserResponse>
    <CollectionId>19</CollectionId>
    <RequestId>19:39</RequestId>
    <InstanceId>20090907T07:00:00.000Z</InstanceId>
  </Request>
  <Request>
    <UserResponse>1</UserResponse>
    <search:LongId>RgAAAju…  …ALS8pQAARL</search:LongId>
    <InstanceId>20091210T17:30:00.000Z</InstanceId>
  </Request>
</MeetingResponse>
```

4.16.2   Response

The following example shows the **MeetingResponse** command response (section 2.2.2.9) sent from the server to the client. The response includes the **meetingresponse:RequestId** element (section 2.2.3.138) that was included in the request message, as well as the **meetingresponse:CalendarId** element (section 2.2.3.18). The response message is the same whether or not the **meetingresponse:InstanceId** element (section 2.2.3.83.1) is included in the request.

```
<?xml version="1.0" encoding="utf-8"?>
<MeetingResponse xmlns:search="Search" xmlns="MeetingResponse">
  <Result>
    <RequestId>19:39</RequestId>
    <Status>1</Status>
    <CalendarId>19:43</CalendarId>
  </Result>
  <Result>
    <search:LongId>RgAAAju…  …ALS8pQAARL</search:LongId>
    <Status>1</Status>
    <CalendarId>19:47</CalendarId>
  </Result>
</MeetingResponse>
```

4.17   Resolving Recipients and Retrieving Free/Busy Data by Using the ResolveRecipients Command

The following sections show how to resolve recipients and retrieve free/busy data by using the **ResolveRecipients** command (section 2.2.2.13).

4.17.1   Request

The following example shows a **ResolveRecipients** command request (section 2.2.2.13) sent from the client to the server to retrieve GAL and contact information by using ANR. In this example, the client sends a request to retrieve recipient information for recipients containing the word "Testers".

```
<?xml version="1.0" encoding="utf-8"?>
<ResolveRecipients xmlns="ResolveRecipients">
  <To>Testers</To>
  <Options>
    <CertificateRetrieval>3</CertificateRetrieval>
```

```
        <MaxCertificates>99</MaxCertificates>
        <MaxAmbiguousRecipients>99</MaxAmbiguousRecipients>
        <Picture>
            <MaxSize>1024</MaxSize>
            <MaxPictures>10</MaxPictures>
        </Picture>
     </Options>
  </ResolveRecipients>
```

4.17.2   Response for a GAL Entry

The following example shows a **ResolveRecipients** command response (section 2.2.2.13) that contains two recipients that are being returned to the client. In the "Testers" distribution list, there are three recipients but only two have valid certificates. The value of the **resolverecipients:Type** element (section 2.2.3.170.4) indicates that the recipient is a GAL entry.

```
    <?xml version="1.0" encoding="utf-8"?>
    <ResolveRecipients xmlns="ResolveRecipients">
      <Status>1</Status>
      <Response>
        <To>Testers</To>
        <Status>1</Status>
        <RecipientCount>2</RecipientCount>
        <Recipient>
          <Type>1</Type>
          <DisplayName>Testers</DisplayName>
          <EmailAddress>testers@example.com</EmailAddress>
          <Certificates>
            <Status>1</Status>
            <CertificateCount>2</CertificateCount>
            <RecipientCount>3</RecipientCount>
            <MiniCertificate>AAAAAEfXfBA=</MiniCertificate>
          </Certificates>
          <Picture>
            <Status>173</Status>
          </Picture>
        </Recipient>
        <Recipient>
         ...
        </Recipient>
      </Response>
    <ResolveRecipients>
```

4.17.3   Response for a Contact Entry

The following example shows a **ResolveRecipients** command response (section 2.2.2.13) for a contact entry. The value of the **resolverecipients:Type** element (section 2.2.3.170.4) indicates that the recipient is a contact entry.

```
    <?xml version="1.0" encoding="utf-8"?>
    <ResolveRecipients xmlns="ResolveRecipients">
      <Status>1</Status>
      <Response>
        <To>Contact</To>
```

```
        <Status>1</Status>
        <RecipientCount>1</RecipientCount>
        <Recipient>
          <Type>2</Type>
          <DisplayName>Chris Gray</DisplayName>
          <EmailAddress>chris@contoso.com</EmailAddress>
        </Recipient>
      </Response>
   </ResolveRecipients>
```

4.17.4   Retrieving Free/Busy Data By Using the ResolveRecipients Command

The following sections show how to retrieve the free/busy data for two users and two distribution lists by using the **ResolveRecipients** command (section 2.2.2.13).

4.17.4.1   Request to Retrieve Free/Busy Data

The following example shows a **ResolveRecipients** command request (section 2.2.2.13) sent from the client to the server to resolve two recipients and two distribution lists and retrieve their free/busy information for a two day period.

```
<?xml version="1.0" encoding="utf-8"?>
<ResolveRecipients xmlns="ResolveRecipients">
  <To>all@contoso.com</To>
  <To>chris@contoso.com</To>
  <To>Anat</To>
  <To>myPersonalDistributionList</To>
  <Options>
    <MaxAmbiguousRecipients>2</MaxAmbiguousRecipients>
    <Availability>
        <StartTime>2008-12-01T08:00:00.000Z</StartTime>
        <EndTime>2008-12-03T08:00:00.000Z</EndTime>
    </Availability>
  </Options>
</ResolveRecipients>
```

4.17.4.2   Response with MergedFreeBusy Data

The following example shows the **ResolveRecipients** command response (section 2.2.2.13) sent from the server to the client in response to the request described in section 4.17.4.1.

As shown in the example, the free/busy data for the all@contoso.com distribution list could not be retrieved and **resolverecipients:Status** element (section 2.2.3.162.11) value of 162 is returned. The free/busy data for Chris Gray is returned successfully. Two ambiguous recipient suggestions are returned for "Anat", neither of which contain the **resolverecipients:Availability** element (section 2.2.3.16), as it is returned only when an exact match is found. And, the personal distribution list returns a variety of successful and unsuccessful queries.

```
<?xml version="1.0" encoding="utf-8"?>
<ResolveRecipients xmlns="ResolveRecipients">
  <Status>1</Status>
  <Response>
    <To>all@contoso.com</To>
```

```
    <Status>1</Status>
    <RecipientCount>1</RecipientCount>
    <Recipient>
      <Type>1</Type>
      <DisplayName>All Contoso Full Time Employees</DisplayName>
      <EmailAddress>all@contoso.com</EmailAddress>
      <Availability>
        <Status>162</Status>
      </Availability>
    </Recipient>
  </Response>
  <Response>
    <To>ryan@contoso.com</To>
    <Status>1</Status>
    <RecipientCount>1</RecipientCount>
    <Recipient>
      <Type>1</Type>
      <DisplayName>Chris Gray</DisplayName>
      <EmailAddress>chris@contoso.com</EmailAddress>
      <Availability>
        <Status>1</Status>
        <MergedFreeBusy>0020000000000000000000000000100200220000010000000</MergedFreeBusy>
      </Availability>
    </Recipient>
  </Response>
  <Response>
    <To>tom</To>
    <Status>3</Status>
    <RecipientCount>30</RecipientCount>
    <Recipient>
      <Type>2</Type>
      <DisplayName>Anat Kerry</DisplayName>
      <EmailAddress>anatk@contoso.com</EmailAddress>
    </Recipient>
    <Recipient>
      <Type>1</Type>
      <DisplayName>Anat Reding</DisplayName>
      <EmailAddress>anetr@contoso.com</EmailAddress>
    </Recipient>
  </Response>
  <Response>
    <To>myPersonalDistributionList</To>
    <Status>1</Status>
    <RecipientCount>4</RecipientCount>
    <Recipient>
      <Type>2</Type>
      <DisplayName>chris@fourthcoffee.com</DisplayName>
      <EmailAddress>chris@fourthcoffee.com</EmailAddress>
      <Availability>
        <Status>162</Status>
      </Availability>
    </Recipient>
    <Recipient>
      <Type>1</Type>
      <DisplayName>Anet Reding</DisplayName>
      <EmailAddress>anetr@contoso.com</EmailAddress>
      <Availability>
        <Status>161</Status>
      </Availability>
    </Recipient>
```

```
        </Recipient>
        <Recipient>
          <Type>2</Type>
          <DisplayName>Dag Rovik</DisplayName>
          <EmailAddress>dag@contoso.com</EmailAddress>
          <Availability>
            <Status>1</Status>
            <MergedFreeBusy>33333333333333333333300000010020022000000010000000</MergedFreeBusy>
          </Availability>
        </Recipient>
        <Recipient>
          <Type>2</Type>
          <DisplayName>fabrice@fourthcoffee.com</DisplayName>
          <EmailAddress>fabrice@fourthcoffee.com</EmailAddress>
          <Availability>
            <Status>162</Status>
          </Availability>
        </Recipient>
      </Response>
  </ResolveRecipients>
```

4.18   Retrieving and Changing OOF Settings by Using the Settings Command

The following sections show how to retrieve and change OOF settings by using the **Settings** command (section 2.2.2.16).

4.18.1   Retrieving OOF Settings

The following sections show how the client requests the user's OOF settings by using the **settings:Get** element (section 2.2.3.75) and indicates the body type to use for the OOF message format.

4.18.1.1   Request

The following example shows a **Settings** command request (section 2.2.2.16) sent from the client to the server to retrieve OOF settings. The client requests that the messages be returned in plain text format, so all OOF messages will be formatted as such.

```
<?xml version="1.0" encoding="utf-8"?>
<Settings xmlns="Settings">
  <Oof>
    <Get>
      <BodyType>Text</BodyType>
    </Get>
  </Oof>
</Settings>
```

4.18.1.2   Response

The following example shows the **Settings** command response (section 2.2.2.16) sent from the server to the client.

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<Settings xmlns="Settings">
<Status>1</Status>
  <Oof>
    <Status>1</Status>
    <Get>
      <OofState>2</OofState>
      <StartTime>2007-05-08T10:45:51.250Z</StartTime>
      <EndTime>2007-05-11T10:45:51.250Z</EndTime>
      <OofMessage>
        <AppliesToInternal />
        <Enabled>1</Enabled>
        <ReplyMessage>Internal OOF Message</ReplyMessage>
       <BodyType>Text</BodyType>
      </OofMessage>
      <OofMessage>
        <AppliesToExternalKnown />
        <Enabled>1</Enabled>
        <ReplyMessage>External OOF Message</ReplyMessage>
         <BodyType>Text</BodyType>
      </OofMessage>
      <OofMessage>
      <AppliesToExternalUnknown /><Enabled>0</Enabled>
      <ReplyMessage>External OOF Message</ReplyMessage>
      <BodyType>Text</BodyType>
     </OofMessage>
    </Get>
  </Oof>
</Settings>
```

4.18.2   Turning On the OOF Message

The following sections show how the client enables the OOF message. The client updates the OOF status by using the **settings:Set** element (section 2.2.3.152).

4.18.2.1   Request

The following example shows a **Settings** command request (section 2.2.2.16) sent from the client to the server to enable the OOF message.

```
<?xml version="1.0" encoding="utf-8"?>
<Settings xmlns="Settings">
  <Oof>
    <Set>
      <OofState>2</OofState>
      <OofMessage>
        <AppliesToInternal/>
        <Enabled>1</Enabled>
        <ReplyMessage> &lt;html&gt;&lt;head&gt;&lt;meta
        http-equiv="Content-Type" content="text/html;
        charset=utf-8"&gt;&lt;style&gt;@font-face
        {font-family:Verdana}p.MsoNormal, li.MsoNormal,
        div.MsoNormal {margin:0in; margin-bottom:.0001pt;
        font-size:10.0pt; font-family:Verdana} a:link,
        span.MsoHyperlink {color:blue; text-
        decoration:underline}a:visited,
        span.MsoHyperlinkFollowed {color:purple;
        text-decoration:underline} span.EmailStyle17
```

```
             {font-family:Arial; color:windowtext} @page Section1
             {margin:1.0in 1.25in 1.0in 1.25in} div.Section1 {}
             &lt;/style&gt; &lt;/head&gt; &lt;body lang="EN-US"
             link="blue" vlink="purple"&gt; &lt;div class="Section1"&gt;
             &lt;p class="MsoNormal"&gt;&lt;font size="2"
             face="Arial"&gt;&lt;span style="font-size:10.0pt;
             font-family:Arial"&gt;I'll be out of the office
             today.&lt;/span&gt;&lt;/font&gt;&lt;/p&gt; &lt;/div&gt;
             &lt;/body&gt;&lt;/html&gt;</ReplyMessage>
             <BodyType>HTML</BodyType>
        </OofMessage>
        <OofMessage>
          <AppliesToExternalKnown/>
          <Enabled>0</Enabled>
        </OofMessage>
        <OofMessage>
          <AppliesToExternalUnknown/>
          <Enabled>0</Enabled>
        </OofMessage>
      </Set>
    </Oof>
  </Settings>
```

4.18.2.2 Response

The following example shows the **Settings** command response (section 2.2.2.16) sent from the
server to the client. The response contains a **settings:Status** element (section 2.2.3.162.14) value
of 1, to indicate that OOF functionality was successfully enabled.

```
<?xml version="1.0" encoding="utf-8"?>
<Settings xmlns="Settings">
  <Status>1</Status>
  <Oof>
    <Status>1</Status>
  </Oof>
</Settings>
```

4.18.3 Turning Off the OOF Message

The following sections show how the client disables the OOF message. The client updates the OOF
status by using the **settings:Set** element (section 2.2.3.152).

4.18.3.1 Request

The following example shows a **Settings** command request (section 2.2.2.16) sent from the client
to the server to disable the OOF message.

```
<?xml version="1.0" encoding="utf-8"?>
<Settings xmlns="Settings">
  <Oof>
    <Set>
      <OofState>0</OofState>
    </Set>
  </Oof>
```

```
  </Settings>
```

4.18.3.2   Response

The following example shows the **Settings** command response (section 2.2.2.16) sent from the server to the client. The response contains a **settings:Status** element (section 2.2.3.162.14) value of 1, to indicate that OOF functionality was successfully disabled.

```
<?xml version="1.0" encoding="utf-8"?>
<Settings xmlns="Settings">
  <Status>1</Status>
  <Oof>
    <Status>1</Status>
  </Oof>
</Settings>
```

4.19   Validating Certificates by Using the ValidateCert Command

The following sections show how to validate certificates by using the **ValidateCert** command (section 2.2.2.20).

4.19.1   Request

The following example shows a **ValidateCert** command request (section 2.2.2.20) sent from the client to the server to validate certificates. The **validatecert:CertificateChain** element (section 2.2.3.20) contains all of the certificates in a certificate chain, and the **validatecert:Certificate** elements (section 2.2.3.23.2) contain the individual certificate values. The **validatecert:CheckCrl** element (section 2.2.3.26) is set to 1 (**TRUE**), indicating that the server cannot ignore an unverifiable revocation status.

```
<?xml version="1.0" encoding="utf-8"?>
<ValidateCert xmlns="ValidateCert">
  <CertificateChain>
  <!—Certificate values have been truncated for example purposes -->
    <Certificate>MIICYjCCAcugAwIBAgIUYGs8jZbX0Vxj/0CIrh8…</Certificate>
    <Certificate>MIIB8zCCAVygAwIBAgIUdhWamYEKM9eaFVFSylR…</Certificate>
    <Certificate>MIIB8zCCAVygAwIBAgIU9uwT6UARSuw1KdJmYN6…</Certificate>
    <Certificate>MIIB8zCCAVygAwIBAgIUB0959dCBM5WSLg7NuM4…</Certificate>
    <Certificate>MIIB8zCCAVygAwIBAgIUGFjVCBrvrguSaNxziWN…</Certificate>
  </CertificateChain>
  <Certificates>
    <Certificate>MIICYjCCAcugAwIBAgIUYGs8jZbX0VxjObu4nw0…</Certificate>
  </Certificates>
  <CheckCRL>1</CheckCRL>
</ValidateCert>
```

4.19.2   Response

The following example shows the **ValidateCert** command response (section 2.2.2.20) sent from the server to the client. A **validatecert:Status** element (section 2.2.3.162.17) value of 1 is returned to indicate that the certificate validation was successful.

```
<?xml version="1.0" encoding="utf-8"?>
<ValidateCert xmlns="ValidateCert">
    <Status>1</Status>
    <Certificate>
        <Status>1</Status>
    </Certificate>
</ValidateCert>
```

4.20   Retrieving User Information by Using the Settings Command

The following sections show how to retrieve user information by using the **Settings** command
(section 2.2.2.16).

4.20.1   Request

The following example shows a **Settings** command request (section 2.2.2.16) sent from the client
to the server to retrieve user information.

```
<?xml version="1.0" encoding="utf-8"?>
<Settings>
    <UserInformation>
        <Get/>
    </UserInformation>
</Settings>
```

4.20.2   Response

The following example shows the **Settings** command response (section 2.2.2.16) sent from the
server to the client.

```
<?xml version="1.0" encoding="utf-8"?>
<Settings>
    <Status>1</Status>
    <UserInformation>
        <Status>1</Status>
        <Get>
            <EmailAddresses>
                <SMTPAddress>chris@contoso.com</SMTPAddress>
            </EmailAddresses>
        </Get>
    </UserInformation>
</Settings>
```

4.21   Setting a Device Password by Using the Settings Command

The following sections show how to set a device password by using the **Settings** command (section
2.2.2.16).

4.21.1   Request

The following example shows a **Settings** command request (section 2.2.2.16) sent from the client
to the server to set the device password.

```
<?xml version="1.0" encoding="utf-8"?>
<Settings>
    <DevicePassword>
        <Set>
            <Password>bar</Password>
        </Set>
    </DevicePassword>
</Settings>
```

4.21.2   Response

The following example shows the **Settings** command response (section 2.2.2.16) sent from the server to the client. A **settings:Status** element (section 2.2.3.162.14) value of 1 indicates that the password was set successfully.

```
<?xml version="1.0" encoding="utf-8"?>
<Settings>
    <Status>1</Status>
    <DevicePassword>
        <Set>
            <Status>...</Status>
        </Set>
    </DevicePassword>
</Settings>
```

4.22   Accessing Documents on File Shares and URIs by Using the Search and ItemOperations Commands

The following sections show how to retrieve an item from a Windows SharePoint Services or UNC site.
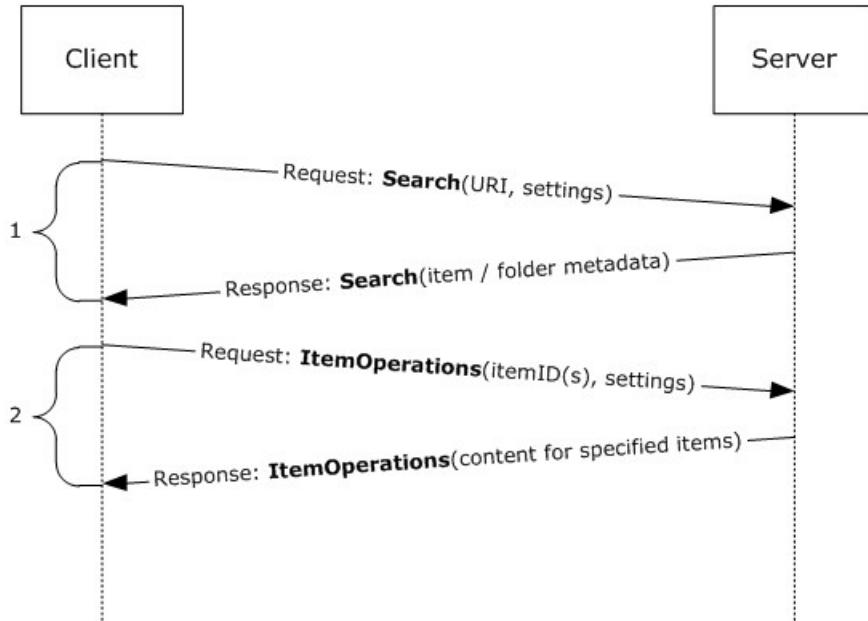
First, the client issues a **Search** command request (section 2.2.2.14), indicating the link to the folder. The server will return folder/item metadata, indicating the ID, file name, size, creation date, last modified date, whether the item is a folder, and whether the item is hidden. For instructions on completing this task, see section 4.22.1.

Next, the client issues an **ItemOperations** command request (section 2.2.2.8), including the ID from the item metadata. For instructions on completing this task, see section 4.22.2.

In issuing the **ItemOperations** command request, the following are considerations for the client pertaining to the size of the file to be retrieved:

- Does the client want to have the item content returned inline in the WBXML, or as separate body parts in the HTTP response? Using WBXML might be easier to implement, but might consume more memory on the device, depending on how the response parser on the device is implemented.

- What is the maximum number of bytes of item content that the client wants to have returned in one response? (Successive requests can be used to obtain the remaining content.)

The following figure shows the request and response pattern that is used to find and retrieve an item located on a Windows SharePoint Services or UNC site.

Figure 7: Finding and retrieving an item from a file share or UNC site

4.22.1   Issuing a Search for Item Metadata

As illustrated by the figure in section 4.22, the client first issues a **Search** request (section 2.2.2.14) to the server to retrieve metadata about the item (if the URI points to an item) or the items (if the URI points to a folder). The client then does the following:

- Indicates that the client is searching a document library store by using the **search:Name** element (section 2.2.3.110.2).

- Provides the URI as the **search:Value** element (section 2.2.3.180) in an **search:EqualTo** query (section 2.2.3.59).

- Indicates the range of results that the client wants to have returned in the response.

In this example, the client is attempting to retrieve metadata for the files in a UNC share.

4.22.1.1   Request

The following example shows the **Search** command request (section 2.2.2.14) sent from the client to the server to search for item metadata.

```
<?xml version="1.0" encoding="utf-8"?>
<Search xmlns:documentlibrary="DocumentLibrary"
xmlns="Search">
  <Store>
    <Name>DocumentLibrary</Name>
    <Query>
      <EqualTo>
        <documentlibrary:LinkId/>
        <Value>\\somehost\directory</Value>
      </EqualTo>
```

```
      </Query>
      <Options>
        <Range>0-999</Range>
      </Options>
    </Store>
  </Search>
```

4.22.1.2 Response

The following example shows the **Search** command response (section 2.2.2.14) sent from the server to the client. The response contains the metadata for the folder and items. The very first node in the response is the top-level node, followed by its child nodes (if any).

```
<?xml version="1.0" encoding="utf-8"?>
<Search xmlns:documentlibrary="DocumentLibrary" xmlns="Search">
  <Status>1</Status>
  <Response>
    <Store>
      <Status>1</Status>
      <Result>
        <Properties>
          <documentlibrary:LinkId>\\somehost\directory
          </documentlibrary:LinkId>
          <documentlibrary:DisplayName>directory
          </documentlibrary:DisplayName>
          <documentlibrary:IsFolder>1
          </documentlibrary:IsFolder>
          <documentlibrary:CreationDate>2007-05-08T17:28:15.375Z
          </documentlibrary:CreationDate>
          <documentlibrary:LastModifiedDate>2007-05-08T17:28:15.406Z
          </documentlibrary:LastModifiedDate>
          <documentlibrary:IsHidden>0</documentlibrary:IsHidden>
        </Properties>
      </Result>
      <Result>
        <Properties>
          <documentlibrary:LinkId>\\somehost\directory\resource
          </documentlibrary:LinkId>
          <documentlibrary:DisplayName>resource
          </documentlibrary:DisplayName>
          <documentlibrary:IsFolder>1</documentlibrary:IsFolder>
          <documentlibrary:CreationDate>2004-03-02T12:34:56.123Z
          </documentlibrary:CreationDate>
          <documentlibrary:LastModifiedDate>2005-04-03T12:34:56.345Z
          </documentlibrary:LastModifiedDate>
          <documentlibrary:IsHidden>0</documentlibrary:IsHidden>
        </Properties>
      </Result>
      <Result>
        <Properties>
          <documentlibrary:LinkId>\\somehost\directory\TestFile.txt
          </documentlibrary:LinkId>
          <documentlibrary:DisplayName>TestFile.txt
          </documentlibrary:DisplayName>
          <documentlibrary:IsFolder>0</documentlibrary:IsFolder>
          <documentlibrary:CreationDate>2004-03-02T12:34:56.123Z
          </documentlibrary:CreationDate>
```

```
                  <documentlibrary:LastModifiedDate>2005-04-03T12:34:56.345Z
                  </documentlibrary:LastModifiedDate>
                  <documentlibrary:IsHidden>0</documentlibrary:IsHidden>
                  <documentlibrary:ContentLength>88
                  </documentlibrary:ContentLength>
                  <documentlibrary:ContentType>text/plain
                  </documentlibrary:ContentType>
              </Properties>
          </Result>
          <Range>0-2</Range>
          <Total>3</Total>
        </Store>
      </Response>
   </Search>
```

4.22.2   Fetching an Item Based on Metadata

When a document library is used to provide item or folder metadata, the client can retrieve a file
within a document library by using the **ItemOperations** command (section 2.2.2.8) and indicating
the **documentlibrary:LinkId** value (section 2.2.3.48) corresponding to the item.

4.22.2.1   Request

The following example shows an **ItemOperations** command request (section 2.2.2.8) sent from the
client to the server to retrieve a file within a document library. The **itemoperations:Range** element
(section 2.2.3.130.1) indicates that the client only requests bytes from 10 through 19 of the item
returned in the response.

```
<?xml version="1.0" encoding="utf-8"?>
<ItemOperations xmlns:documentlibrary="DocumentLibrary"
xmlns="ItemOperations">
   <Fetch>
      <Store>DocumentLibrary</Store>
      <documentlibrary:LinkId>\\somehost\directory\
      ActiveSyncDocumentFetch.txt</documentlibrary:LinkId>
      <Options>
        <Range>10-19</Range>
      </Options>
   </Fetch>
</ItemOperations>
```

4.22.2.2   Response

The following example shows the **ItemOperations** command response (section 2.2.2.8) that
contains the requested item. The binary content of the file is encoded with base64 encoding and is
included in the **itemoperations:Data** element (section 2.2.3.39.1).

```
<?xml version="1.0" encoding="utf-8"?>
<ItemOperations xmlns:documentlibrary="DocumentLibrary"
xmlns="ItemOperations">
   <Status>1</Status>
   <Response>
      <Fetch>
         <Status>1</Status>
```

```
          <documentlibrary:LinkId>\\somehost\directory\
          ActiveSyncDocumentFetch.txt</documentlibrary:LinkId>
          <Properties>
            <Range>10-19</Range>
            <Total>26</Total>
            <Data>S0xNTk9QUVJTVA==</Data>
            <Version>2005-04-03T12:34:56.345Z</Version>
          </Properties>
        </Fetch>
      </Response>
    </ItemOperations>
```

4.23   Using the Supported Element and Ghosted Elements in the Sync Command

The following sections show how to use ghosted contact elements and the **airsync:Supported** element (section 2.2.3.164) in the **Sync** command (section 2.2.2.19). To demonstrate the effect of using the **airsync:Supported** element, the same contact is modified with the same **Sync** command request in two separate scenarios: once without using the **airsync:Supported** element in the initial **Sync** command request, and once with using the **airsync:Supported** element in the initial **Sync** command request.

For both scenarios, there is a single contact that initially has the following field values.

Contact field	Value
First name	Elvin
Last name	Sills
Company name	Contoso, Inc.
Job title	Assistant Manager
Department	Accounting
Office location	123
Email address	elvin@contoso.com

4.23.1   Modifying a Contact with no Ghosted Elements

In this scenario, the client does not include the **airsync:Supported** element in the initial **Sync** command request for the default Contacts folder. This causes all of the elements that can be ghosted to not be ghosted for all items in that folder, as specified in section 2.2.3.164.

First, the client sends the following initial **Sync** command request.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync">
  <Collections>
    <Collection>
      <SyncKey>0</SyncKey>
      <CollectionId>2</CollectionId>
    </Collection>
  </Collections>
```

```
      </Sync>
```

The server sends the following response.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync">
  <Collections>
    <Collection>
      <SyncKey>1545750018</SyncKey>
      <CollectionId>2</CollectionId>
      <Status>1</Status>
    </Collection>
  </Collections>
</Sync>
```

The client then synchronizes the contents of the default Contacts folder by using the following **Sync** command request.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync">
  <Collections>
    <Collection>
      <SyncKey>1545750018</SyncKey>
      <CollectionId>2</CollectionId>
    </Collection>
  </Collections>
</Sync>
```

The server sends the following response.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync" xmlns:airsyncbase="AirSyncBase" xmlns:contacts="Contacts">
  <Collections>
    <Collection>
      <SyncKey>1593447001</SyncKey>
      <CollectionId>2</CollectionId>
      <Status>1</Status>
      <Commands>
        <Add>
          <ServerId>2:1</ServerId>
          <ApplicationData>
            <airsyncbase:Body>
              <airsyncbase:Type>1</airsyncbase:Type>
              <airsyncbase:EstimatedDataSize>0</airsyncbase:EstimatedDataSize>
              <airsyncbase:Truncated>1</airsyncbase:Truncated>
            </airsyncbase:Body>
            <contacts:Department>Accounting</contacts:Department>
            <contacts:Email1Address>"elvin@contoso.com"
                &lt;elvin@contoso.com&gt;</contacts:Email1Address>
            <contacts:FileAs>Sills, Elvin</contacts:FileAs>
            <contacts:FirstName>Elvin</contacts:FirstName>
            <contacts:CompanyName>Contoso, Inc.</contacts:CompanyName>
            <contacts:LastName>Sills</contacts:LastName>
            <contacts:JobTitle>Assistant Manager</contacts:JobTitle>
            <contacts:OfficeLocation>123</contacts:OfficeLocation>
```

```
                <airsyncbase:NativeBodyType>1</airsyncbase:NativeBodyType>
             </ApplicationData>
          </Add>
       </Commands>
     </Collection>
   </Collections>
 </Sync>
```

The client modifies the contact (3) using the **airsync:Change** element (section 2.2.3.24) in a **Sync** command request. The client updates the values of the **contacts:JobTitle** ([MS-ASCNTC] section 2.2.2.42) and **contacts:OfficeLocation** ([MS-ASCNTC] section 2.2.2.49) elements, and does not include the **contacts:CompanyName** element ([MS-ASCNTC] section 2.2.2.22). The client sends the following **Sync** command request.

```
 <?xml version="1.0" encoding="utf-8"?>
 <Sync xmlns="AirSync" xmlns:airsyncbase="AirSyncBase" xmlns:contacts="Contacts">
   <Collections>
     <Collection>
       <SyncKey>1593447001</SyncKey>
       <CollectionId>2</CollectionId>
       <Commands>
         <Change>
           <ServerId>2:1</ServerId>
           <ApplicationData>
             <contacts:Department>Accounting</contacts:Department>
             <contacts:Email1Address>"elvin@contoso.com"
                 &lt;elvin@contoso.com&gt;</contacts:Email1Address>
             <contacts:FileAs>Sills, Elvin</contacts:FileAs>
             <contacts:FirstName>Elvin</contacts:FirstName>
             <contacts:LastName>Sills</contacts:LastName>
             <contacts:JobTitle>Manager</contacts:JobTitle>
             <contacts:OfficeLocation>456</contacts:OfficeLocation>
           </ApplicationData>
         </Change>
       </Commands>
     </Collection>
   </Collections>
 </Sync>
```

The server sends the following response.

```
 <?xml version="1.0" encoding="utf-8"?>
 <Sync xmlns="AirSync">
   <Collections>
     <Collection>
       <SyncKey>1020675135</SyncKey>
       <CollectionId>2</CollectionId>
       <Status>1</Status>
     </Collection>
   </Collections>
 </Sync>
```

On the server, the contact (3) now has the following field values.

Contact field	Old value	New value
First name	Elvin	Elvin
Last name	Sills	Sills
Company name	Contoso, Inc.	
Job title	Assistant Manager	Manager
Department	Accounting	Accounting
Office location	123	456
Email address	elvin@contoso.com	elvin@contoso.com

Since the **contacts:CompanyName** element was not ghosted, and the element was not included within the **airsync:Change** element, the value for this element was deleted.

4.23.2   Modifying a Contact with Ghosted Elements

In this scenario, the client includes the **airsync:Supported** element in the initial **Sync** command request for the default Contacts folder. This causes all of the elements not included as child elements of the **airsync:Supported** element that can be ghosted to be ghosted for all items in that folder, as specified in section 2.2.3.164.

First, the client sends the following initial **Sync** command request.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync" xmlns:contacts="Contacts">
  <Collections>
    <Collection>
      <SyncKey>0</SyncKey>
      <CollectionId>2</CollectionId>
      <Supported>
        <contacts:JobTitle/>
        <contacts:OfficeLocation/>
      </Supported>
    </Collection>
  </Collections>
</Sync>
```

The server sends the following response.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync">
  <Collections>
    <Collection>
      <SyncKey>719108528</SyncKey>
      <CollectionId>2</CollectionId>
      <Status>1</Status>
    </Collection>
  </Collections>
</Sync>
```

Since the client included the **contacts:JobTitle element** ([MS-ASCNTC] section 2.2.2.42) and the **contacts:OfficeLocation** element ([MS-ASCNTC] section 2.2.2.49) as child elements of the **airsync:Supported** element, those elements are not ghosted. All other elements in the **Contacts** namespace are ghosted.

The client then synchronizes the contents of the default Contacts folder by using the following **Sync** command request.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync">
  <Collections>
    <Collection>
      <SyncKey>719108528</SyncKey>
      <CollectionId>2</CollectionId>
    </Collection>
  </Collections>
</Sync>
```

The server sends the following response.

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync" xmlns:airsyncbase="AirSyncBase" xmlns:contacts="Contacts">
  <Collections>
    <Collection>
      <SyncKey>846142473</SyncKey>
      <CollectionId>2</CollectionId>
      <Status>1</Status>
      <Commands>
        <Add>
          <ServerId>2:1</ServerId>
          <ApplicationData>
            <airsyncbase:Body>
              <airsyncbase:Type>1</airsyncbase:Type>
              <airsyncbase:EstimatedDataSize>0</airsyncbase:EstimatedDataSize>
              <airsyncbase:Truncated>1</airsyncbase:Truncated>
            </airsyncbase:Body>
            <contacts:Department>Accounting</contacts:Department>
            <contacts:Email1Address>"elvin@contoso.com"
                &lt;elvin@contoso.com&gt;</contacts:Email1Address>
            <contacts:FileAs>Sills, Elvin</contacts:FileAs>
            <contacts:FirstName>Elvin</contacts:FirstName>
            <contacts:CompanyName>Contoso, Inc.</contacts:CompanyName>
            <contacts:LastName>Sills</contacts:LastName>
            <contacts:JobTitle>Assistant Manager</contacts:JobTitle>
            <contacts:OfficeLocation>123</contacts:OfficeLocation>
            <airsyncbase:NativeBodyType>1</airsyncbase:NativeBodyType>
          </ApplicationData>
        </Add>
      </Commands>
    </Collection>
  </Collections>
</Sync>
```

The client modifies the contact (3) using the **airsync:Change** element (section 2.2.3.24) in a **Sync** command request. The client updates the values of the **contacts:JobTitle** and

contacts:OfficeLocation elements and does not include the **contacts:CompanyName** element ([MS-ASCNTC] section 2.2.2.22). The client sends the following **Sync** command request.

```xml
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync" xmlns:airsyncbase="AirSyncBase" xmlns:contacts="Contacts">
  <Collections>
    <Collection>
      <SyncKey>846142473</SyncKey>
      <CollectionId>2</CollectionId>
      <Commands>
        <Change>
          <ServerId>2:1</ServerId>
          <ApplicationData>
            <contacts:Department>Accounting</contacts:Department>
            <contacts:Email1Address>"elvin@contoso.com"
                &lt;elvin@contoso.com&gt;</contacts:Email1Address>
            <contacts:FileAs>Sills, Elvin</contacts:FileAs>
            <contacts:FirstName>Elvin</contacts:FirstName>
            <contacts:LastName>Sills</contacts:LastName>
            <contacts:JobTitle>Manager</contacts:JobTitle>
            <contacts:OfficeLocation>456</contacts:OfficeLocation>
          </ApplicationData>
        </Change>
      </Commands>
    </Collection>
  </Collections>
</Sync>
```

The server sends the following response.

```xml
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync">
  <Collections>
    <Collection>
      <SyncKey>195024309</SyncKey>
      <CollectionId>2</CollectionId>
      <Status>1</Status>
    </Collection>
  </Collections>
</Sync>
```

On the server, the contact (3) now has the following field values.

Contact field	Old value	New value
First name	Elvin	Elvin
Last name	Sills	Sills
Company name	Contoso, Inc.	Contoso, Inc.
Job title	Assistant Manager	Manager
Department	Accounting	Accounting
Office location	123	456

Contact field	Old value	New value
Email address	elvin@contoso.com	elvin@contoso.com

Since the **contacts:CompanyName** element was ghosted and the element was not included within the **airsync:Change** element, the value for this element was preserved.

4.24   Moving a Conversation by Using the ItemOperations Command

The following sections show how to move a conversation to another folder by using the **ItemOperations** command (section 2.2.2.8).

4.24.1   Request

The following example shows an **ItemOperations** command request (section 2.2.2.8) sent from the client to the server to move a conversation from one folder to another. The **itemoperations:ConversationId** element (section 2.2.3.35.1) identifies the conversation to move, the **itemoperations:DstFldId** element (section 2.2.3.49.1) identifies the destination folder, and the **itemoperations:MoveAlways** element (section 2.2.3.108) indicates that all future messages that are part of the conversation are to be moved to the destination folder as well.

```
<?xml version="1.0" encoding="utf-8"?>
<ItemOperations xmlns="ItemOperations">
  <Move>
    <DstFldId>15</DstFldId>
    <ConversationId>...</ConversationId>
    <Options>
      <MoveAlways/>
    </Options>
  </Move>
</ItemOperations>
```

4.24.2   Response

The following example shows the **ItemOperations** command response (section 2.2.2.8) sent from the server to the client. An **itemoperations:Status** element (section 2.2.3.162.7) value of 1 indicates that the move operation was successful. The **itemoperations:ConversationId** element (section 2.2.3.35.1) value is returned to confirm the conversation that was moved.

```
<?xml version="1.0" encoding="utf-8"?>
<ItemOperations xmlns="ItemOperations">
  <Status>1</Status>
  <Response>
    <Status>1</Status>
    <Move>
      <Status>1</Status>
      <ConversationId>...</ConversationId>
    </Move>
  </Response>
</ItemOperations>
```

5   Security

5.1   Security Considerations for Implementers

The client device honors all policies sent down by the server, or sends up the appropriate status codes indicating the non-success.

5.2   Index of Security Parameters

Security parameter	Section
Provision Command	2.2.2.12
ValidateCert Command	2.2.2.20

6 Appendix A: Full XML Schema

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

Schema name	Prefix	Section
AirSync namespace schema	airsync	6.1
Autodiscover MobileSync namespace schema	Mobilesync	6.2
Autodiscover request schema	autodiscover	6.3
Autodiscover response schema	autodiscover	6.4
ComposeMail namespace schema	composemail	6.5
FolderCreate request schema	folderhierarchy	6.6
FolderCreate response schema	folderhierarchy	6.7
FolderDelete request schema	folderhierarchy	6.8
FolderDelete response schema	folderhierarchy	6.9
FolderHierarchy namespace schema	folderhierarchy	6.10
FolderSync request schema	folderhierarchy	6.11
FolderSync response schema	folderhierarchy	6.12
FolderUpdate request schema	folderhierarchy	6.13
FolderUpdate response schema	folderhierarchy	6.14
GAL namespace schema	gal	6.15
GetItemEstimate request schema	getitemestimate	6.16
GetItemEstimate response schema	getitemestimate	6.17
ItemOperations namespace schema	itemoperations	6.18
ItemOperations request schema	itemoperations	6.19
ItemOperations response schema	itemoperations	6.20
MeetingResponse request schema	meetingresponse	6.21
MeetingResponse response schema	meetingresponse	6.22
MoveItems request schema	move	6.23
MoveItems response schema	move	6.24
Ping request schema	ping	6.25
Ping response schema	ping	6.26
ResolveRecipients request schema	resolverecipients	6.27

Release: February 10, 2014

Schema name	Prefix	Section
ResolveRecipients response schema	resolverecipients	6.28
Search namespace schema	search	6.29
Search request schema	search	6.30
Search response schema	search	6.31
SendMail request schema	composemail	6.32
SendMail response schema	composemail	6.33
Settings namespace schema	settings	6.34
Settings request schema	settings	6.35
Settings response schema	settings	6.36
SmartForward request schema	composemail	6.37
SmartForward response schema	composemail	6.38
SmartReply request schema	composemail	6.39
SmartReply response schema	composemail	6.40
Sync request schema	airsync	6.41
Sync response schema	airsync	6.42
ValidateCert request schema	validatecert	6.43
ValidateCert response schema	validatecert	6.44

6.1   AirSync Namespace Schema

This section contains the contents of the AirSync.xsd file. The additional files that this schema file requires to operate correctly are listed in the following table.

File name	Defining specification
AirSyncBase.xsd	[MS-ASAIRS] section 6
Calendar.xsd	[MS-ASCAL] section 6
Contacts.xsd	[MS-ASCNTC] section 6.1
Contacts2.xsd	[MS-ASCNTC] section 6.2
RightsManagement.xsd	[MS-ASRM] section 6

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:airsyncbase=
    "AirSyncBase" xmlns:calendar="Calendar" xmlns:contacts="Contacts"
    xmlns:contacts2="Contacts2" xmlns:rm="RightsManagement" xmlns="AirSync"
    targetNamespace="AirSync" elementFormDefault="qualified"
    attributeFormDefault="unqualified">
  <xs:import namespace="AirSyncBase" schemaLocation="AirSyncBase.xsd"/>
```

```
<xs:import namespace="Calendar" schemaLocation="Calendar.xsd"/>
<xs:import namespace="Contacts" schemaLocation="Contacts.xsd"/>
<xs:import namespace="Contacts2" schemaLocation="Contacts2.xsd"/>
<xs:import namespace="RightsManagement"
    schemaLocation="RightsManagement.xsd"/>
<xs:element name="SyncKey">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="64"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="ClientId">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="64"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="ServerId">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="64"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="Status" type="xs:unsignedByte"/>
<xs:element name="Class" type="xs:string"/>
<xs:element name="CollectionId">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="64"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="GetChanges" type="xs:boolean"/>
<xs:element name="MoreAvailable" type="airsyncbase:EmptyTag"/>
<xs:element name="WindowSize">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="512"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="FilterType">
  <xs:simpleType>
    <xs:restriction base="xs:unsignedByte">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="8"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="Conflict">
  <xs:simpleType>
    <xs:restriction base="xs:unsignedByte">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="1"/>
    </xs:restriction>
```

```
      </xs:simpleType>
    </xs:element>
    <xs:element name="DeletesAsMoves" type="xs:boolean"/>
    <xs:element name="Supported">
      <xs:complexType mixed="true">
        <xs:sequence minOccurs="0">
          <xs:choice maxOccurs="unbounded">
            <xs:group ref="calendar:GhostingProps"/>
            <xs:group ref="contacts:GhostingProps"/>
            <xs:group ref="contacts2:GhostingProps"/>
          </xs:choice>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="SoftDelete">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="ServerId"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="MIMESupport">
      <xs:simpleType>
        <xs:restriction base="xs:unsignedByte">
          <xs:minInclusive value="0"/>
          <xs:maxInclusive value="2"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="MIMETruncation">
      <xs:simpleType>
        <xs:restriction base="xs:unsignedByte">
          <xs:minInclusive value="0"/>
          <xs:maxInclusive value="8"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Wait">
      <xs:simpleType>
        <xs:restriction base="xs:integer">
          <xs:minInclusive value="1"/>
          <xs:maxInclusive value="59"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="Limit" type="xs:integer"/>
    <xs:element name="Partial" type="airsyncbase:EmptyTag"/>
    <xs:element name="ConversationMode" type="xs:boolean"/>
    <xs:element name="MaxItems">
      <xs:simpleType>
        <xs:restriction base="xs:integer">
          <xs:minInclusive value="1"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="HeartbeatInterval">
      <xs:simpleType>
        <xs:restriction base="xs:integer">
          <xs:minInclusive value="60"/>
```

```
              <xs:maxInclusive value="3540"/>
          </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="Options">
      <xs:complexType>
        <xs:choice maxOccurs="unbounded">
          <xs:element ref="FilterType" minOccurs="0"/>
          <xs:element ref="Class" minOccurs="0"/>
          <xs:element ref="airsyncbase:BodyPreference" minOccurs="0"
              maxOccurs="unbounded"/>
          <xs:element ref="airsyncbase:BodyPartPreference" minOccurs="0"/>
          <xs:element ref="Conflict" minOccurs="0"/>
          <xs:element ref="MIMESupport" minOccurs="0"/>
          <xs:element ref="MIMETruncation" minOccurs="0"/>
          <xs:element ref="MaxItems" minOccurs="0"/>
          <xs:element ref="rm:RightsManagementSupport" minOccurs="0"/>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:schema>
```

6.2 Autodiscover MobileSync Namespace Schema

This section contains the contents of the AutodiscoverMobileSync.xsd file.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns=
    "http://schemas.microsoft.com/exchange/autodiscover/mobilesync/responseschema/2006"
    targetNamespace=
    "http://schemas.microsoft.com/exchange/autodiscover/mobilesync/responseschema/2006"
    elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="Response">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Culture" type="xs:string" minOccurs="0"/>
        <xs:element name="User" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="DisplayName" minOccurs="0"/>
              <xs:element name="EMailAddress" minOccurs="1"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="Action" minOccurs="0">
          <xs:complexType>
            <xs:all>
              <xs:element name="Redirect" type="xs:string"
                  minOccurs="0"/>
              <xs:element name="Settings" minOccurs="0">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Server" maxOccurs="unbounded">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="Type" type="xs:string"
                              minOccurs="0"/>
```

```
                         <xs:element name="Url" type="xs:string"
                              minOccurs="0"/>
                         <xs:element name="Name" type="xs:string"
                              minOccurs="0"/>
                         <xs:element name="ServerData" type="xs:string"
                              minOccurs="0"/>
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="Error" minOccurs="0">
              <xs:complexType>
                <xs:all>
                  <xs:element name="Status" type="xs:string"
                       minOccurs="0"/>
                  <xs:element name="Message" type="xs:string"
                       minOccurs="0"/>
                  <xs:element name="DebugData" type="xs:string"
                       minOccurs="0"/>
                  <xs:element name="ErrorCode" type="xs:integer"
                       minOccurs="0"/>
                </xs:all>
              </xs:complexType>
            </xs:element>
          </xs:all>
        </xs:complexType>
      </xs:element>
      <xs:element name="Error" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ErrorCode" type="xs:integer"
                 minOccurs="0"/>
            <xs:element name="Message" type="xs:string"
                 minOccurs="0"/>
            <xs:element name="DebugData" type="xs:string"
                 minOccurs="0"/>
          </xs:sequence>
          <xs:attribute name="Time" type="xs:string"/>
          <xs:attribute name="Id" type="xs:unsignedInt"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

6.3 Autodiscover Request Schema

This section contains the contents of the AutodiscoverRequest.xsd file.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns=
    "http://schemas.microsoft.com/exchange/autodiscover/mobilesync/requestschema/2006"
    targetNamespace=
    "http://schemas.microsoft.com/exchange/autodiscover/mobilesync/requestschema/2006"
```

```
         elementFormDefault="qualified" attributeFormDefault="unqualified">
      <xs:element name="Autodiscover">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Request" type="RequestType"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="RequestType">
        <xs:sequence>
          <xs:sequence>
            <xs:element name="EMailAddress" type="xs:string"/>
            <xs:element name="AcceptableResponseSchema" type="xs:string"/>
          </xs:sequence>
        </xs:sequence>
      </xs:complexType>
    </xs:schema>
```

6.4   Autodiscover Response Schema

This section contains the contents of the AutodiscoverResponse.xsd file. The additional file that this schema file requires to operate correctly is listed in the following table.

File name	Defining section
AutodiscoverMobileSync.xsd	6.2

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns=
    "http://schemas.microsoft.com/exchange/autodiscover/responseschema/2006"
    targetNamespace=
    "http://schemas.microsoft.com/exchange/autodiscover/responseschema/2006"
    xmlns:mobilesync=
    "http://schemas.microsoft.com/exchange/autodiscover/mobilesync/responseschema/2006"
    elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import namespace=
      "http://schemas.microsoft.com/exchange/autodiscover/mobilesync/responseschema/2006"
      schemaLocation="AutodiscoverMobileSync.xsd"/>
  <xs:element name="Autodiscover">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="mobilesync:Response"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

6.5   ComposeMail Namespace Schema

This section contains the contents of the ComposeMail.xsd file. The additional file that this schema file requires to operate correctly is listed in the following table.

File name	Defining specification
AirSyncBase.xsd	[MS-ASAIRS] section 6

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:airsyncbase=
    "AirSyncBase" xmlns="ComposeMail" targetNamespace="ComposeMail"
    elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import namespace="AirSyncBase" schemaLocation="AirSyncBase.xsd"/>
  <xs:element name="ClientId">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:minLength value="1"/>
        <xs:maxLength value="40"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="AccountId" type="xs:string"/>
  <xs:element name="SaveInSentItems" type="airsyncbase:EmptyTag"/>
  <xs:element name="ReplaceMime" type="airsyncbase:EmptyTag"/>
  <xs:element name="Mime" type="xs:string"/>
  <xs:element name="Status" type="xs:integer"/>
  <xs:element name="Source">
    <xs:complexType>
      <xs:all>
        <xs:element name="FolderId" minOccurs="0">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:minLength value="1"/>
              <xs:maxLength value="64"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="ItemId" minOccurs="0">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:minLength value="1"/>
              <xs:maxLength value="64"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="LongId" minOccurs="0">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:minLength value="1"/>
              <xs:maxLength value="256"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="InstanceId" type="xs:string" minOccurs="0"/>
      </xs:all>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

6.6   FolderCreate Request Schema

This section contains the contents of the FolderCreateRequest.xsd file. The additional file that this schema file requires to operate correctly is listed in the following table.

Release: February 10, 2014

File name	Defining section
FolderHierarchy.xsd	6.10

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="FolderHierarchy"
    targetNamespace="FolderHierarchy" elementFormDefault="qualified"
    attributeFormDefault="unqualified">
  <xs:include schemaLocation="FolderHierarchy.xsd"/>
  <xs:element name="FolderCreate">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="SyncKey"/>
        <xs:element ref="ParentId"/>
        <xs:element name="DisplayName">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:maxLength value="256"/>
              <xs:minLength value="1"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="Type" type="xs:unsignedByte"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

6.7   FolderCreate Response Schema

This section contains the contents of the FolderCreateResponse.xsd file. The additional file that this schema file requires to operate correctly is listed in the following table.

File name	Defining section
FolderHierarchy.xsd	6.10

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="FolderHierarchy"
    targetNamespace="FolderHierarchy" elementFormDefault="qualified"
    attributeFormDefault="unqualified">
  <xs:include schemaLocation="FolderHierarchy.xsd"/>
  <xs:element name="FolderCreate">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Status"/>
        <xs:element ref="SyncKey" minOccurs="0"/>
        <xs:element ref="ServerId" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

6.8 FolderDelete Request Schema

This section contains the contents of the FolderDeleteRequest.xsd file. The additional file that this schema file requires to operate correctly is listed in the following table.

File name	Defining section
FolderHierarchy.xsd	6.10

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="FolderHierarchy"
    targetNamespace="FolderHierarchy" elementFormDefault="qualified"
    attributeFormDefault="unqualified">
  <xs:include schemaLocation="FolderHierarchy.xsd"/>
  <xs:element name="FolderDelete">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="SyncKey"/>
        <xs:element ref="ServerId"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

6.9 FolderDelete Response Schema

This section contains the contents of the FolderDeleteResponse.xsd file. The additional file that this schema file requires to operate correctly is listed in the following table.

File name	Defining section
FolderHierarchy.xsd	6.10

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="FolderHierarchy"
    targetNamespace="FolderHierarchy" elementFormDefault="qualified"
    attributeFormDefault="unqualified">
  <xs:include schemaLocation="FolderHierarchy.xsd"/>
  <xs:element name="FolderDelete">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Status"/>
        <xs:element ref="SyncKey" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

6.10 FolderHierarchy Namespace Schema

This section contains the contents of the FolderHierarchy.xsd file.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="FolderHierarchy"
    targetNamespace="FolderHierarchy" elementFormDefault="qualified"
```

```
            attributeFormDefault="unqualified">
   <xs:element name="ServerId">
     <xs:simpleType>
       <xs:restriction base="xs:string">
         <xs:maxLength value="64"/>
         <xs:minLength value="1"/>
       </xs:restriction>
     </xs:simpleType>
   </xs:element>
   <xs:element name="ParentId">
     <xs:simpleType>
       <xs:restriction base="xs:string">
         <xs:maxLength value="64"/>
         <xs:minLength value="1"/>
       </xs:restriction>
     </xs:simpleType>
   </xs:element>
   <xs:element name="Status" type="xs:unsignedByte"/>
   <xs:element name="SyncKey">
     <xs:simpleType>
       <xs:restriction base="xs:string">
         <xs:maxLength value="64"/>
         <xs:minLength value="1"/>
       </xs:restriction>
     </xs:simpleType>
   </xs:element>
 </xs:schema>
```

6.11   FolderSync Request Schema

This section contains the contents of the FolderSyncRequest.xsd file. The additional file that this schema file requires to operate correctly is listed in the following table.

File name	Defining section
FolderHierarchy.xsd	[6.10](#)

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="FolderHierarchy"
    targetNamespace="FolderHierarchy" elementFormDefault="qualified"
    attributeFormDefault="unqualified">
  <xs:include schemaLocation="FolderHierarchy.xsd"/>
  <xs:element name="FolderSync">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="SyncKey"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

6.12   FolderSync Response Schema

This section contains the contents of the FolderSyncResponse.xsd file. The additional file that this schema file requires to operate correctly is listed in the following table.

Release: February 10, 2014

File name	Defining section
FolderHierarchy.xsd	6.10

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="FolderHierarchy"
    targetNamespace="FolderHierarchy" elementFormDefault="qualified"
    attributeFormDefault="unqualified">
  <xs:include schemaLocation="FolderHierarchy.xsd"/>
  <xs:element name="FolderSync">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Status"/>
        <xs:element ref="SyncKey" minOccurs="0"/>
        <xs:element name="Changes" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Count" type="xs:unsignedInt" minOccurs="0"/>
              <xs:element name="Update" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element ref="ServerId"/>
                    <xs:element ref="ParentId"/>
                    <xs:element name="DisplayName" type="xs:string"/>
                    <xs:element name="Type" type="xs:integer"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="Delete" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element ref="ServerId"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="Add" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element ref="ServerId"/>
                    <xs:element ref="ParentId"/>
                    <xs:element name="DisplayName" type="xs:string"/>
                    <xs:element name="Type" type="xs:integer"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

6.13   FolderUpdate Request Schema

This section contains the contents of the FolderUpdateRequest.xsd file. The additional file that this schema file requires to operate correctly is listed in the following table.

File name	Defining section
FolderHierarchy.xsd	6.10

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="FolderHierarchy"
    targetNamespace="FolderHierarchy" elementFormDefault="qualified"
    attributeFormDefault="unqualified">
  <xs:include schemaLocation="FolderHierarchy.xsd"/>
  <xs:element name="FolderUpdate">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="SyncKey"/>
        <xs:element ref="ServerId"/>
        <xs:element ref="ParentId"/>
        <xs:element name="DisplayName">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:maxLength value="256"/>
              <xs:minLength value="1"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

6.14   FolderUpdate Response Schema

This section contains the contents of the FolderUpdateResponse.xsd file. The additional file that this schema file requires to operate correctly is listed in the following table.

File name	Defining section
FolderHierarchy.xsd	6.10

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="FolderHierarchy"
    targetNamespace="FolderHierarchy" elementFormDefault="qualified"
    attributeFormDefault="unqualified">
  <xs:include schemaLocation="FolderHierarchy.xsd"/>
  <xs:element name="FolderUpdate">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Status"/>
        <xs:element ref="SyncKey" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

6.15   GAL Namespace Schema

This section contains the contents of the GAL.xsd file. The additional file that this schema file requires to operate correctly is listed in the following table.

File name	Defining specification
RightsManagement.xsd	[MS-ASRM] section 6

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:rm=
    "RightsManagement" xmlns="GAL" targetNamespace="GAL" elementFormDefault=
    "qualified" attributeFormDefault="unqualified">
  <xs:import namespace="RightsManagement" schemaLocation=
      "RightsManagement.xsd"/>
  <xs:group name="TopLevelSchemaProps">
    <xs:sequence>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="DisplayName" type="xs:string"/>
        <xs:element name="Phone" type="xs:string"/>
        <xs:element name="Office" type="xs:string"/>
        <xs:element name="Title" type="xs:string"/>
        <xs:element name="Company" type="xs:string"/>
        <xs:element name="Alias" type="xs:string"/>
        <xs:element name="FirstName" type="xs:string"/>
        <xs:element name="LastName" type="xs:string"/>
        <xs:element name="HomePhone" type="xs:string"/>
        <xs:element name="MobilePhone" type="xs:string"/>
        <xs:element name="EmailAddress" type="xs:string"/>
        <xs:element name="Picture">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Status" type="xs:integer"/>
              <xs:element name="Data" type="xs:string" minOccurs="0"/>
              <xs:element ref="rm:RightsManagementLicense" minOccurs="0"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:sequence>
  </xs:group>
</xs:schema>
```

6.16   GetItemEstimate Request Schema

This section contains the contents of the GetItemEstimate.xsd file. The additional file that this schema file requires to operate correctly is listed in the following table.

File name	Defining section
AirSync.xsd	section 6.1

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:airsync=
    "AirSync" xmlns="GetItemEstimate" targetNamespace="GetItemEstimate"
    elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import namespace="AirSync" schemaLocation="AirSync.xsd"/>
  <xs:element name="GetItemEstimate">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Collections">
          <xs:complexType>
```

```
                <xs:sequence>
                  <xs:element name="Collection" maxOccurs="unbounded">
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element ref="airsync:SyncKey"/>
                        <xs:element name="CollectionId">
                          <xs:simpleType>
                            <xs:restriction base="xs:string">
                              <xs:maxLength value="64"/>
                            </xs:restriction>
                          </xs:simpleType>
                        </xs:element>
                        <xs:element ref="airsync:FilterType" minOccurs="0"/>
                        <xs:element ref="airsync:Options" minOccurs="0"
                          maxOccurs="2"/>
                        <xs:element ref="airsync:ConversationMode" minOccurs="0"/>
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:schema>
```

6.17 GetItemEstimate Response Schema

This section contains the contents of the GetItemEstimateResponse.xsd file.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="GetItemEstimate"
    targetNamespace="GetItemEstimate" elementFormDefault="qualified"
    attributeFormDefault="unqualified">
  <xs:element name="GetItemEstimate">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Status" type="xs:unsignedByte" minOccurs="0"/>
        <xs:element name="Response" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Status" type="xs:unsignedByte"/>
              <xs:element name="Collection" minOccurs="0" maxOccurs="1">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="CollectionId">
                      <xs:simpleType>
                        <xs:restriction base="xs:string">
                          <xs:maxLength value="64"/>
                        </xs:restriction>
                      </xs:simpleType>
                    </xs:element>
                    <xs:element name="Estimate" type="xs:integer"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
```

```
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

6.18 ItemOperations Namespace Schema

This section contains the contents of the ItemOperations.xsd file. The additional files that this schema file requires to operate correctly are listed in the following table.

File name	Defining specification
AirSyncBase.xsd	[MS-ASAIRS] section 6
Calendar.xsd	[MS-ASCAL] section 6
Contacts.xsd	[MS-ASCNTC] section 6.1
Contacts2.xsd	[MS-ASCNTC] section 6.2
DocumentLibrary.xsd	[MS-ASDOC] section 6
Email.xsd	[MS-ASEMAIL] section 6.1
Email2.xsd	[MS-ASEMAIL] section 6.2
RightsManagement.xsd	[MS-ASRM] section 6
Tasks.xsd	[MS-ASTASK] section 6

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:airsyncbase=
    "AirSyncBase" xmlns:calendar="Calendar" xmlns:contacts="Contacts"
    xmlns:contacts2="Contacts2" xmlns:documentlibrary="DocumentLibrary"
    xmlns:email="Email" xmlns:email2="Email2" xmlns:rm="RightsManagement"
    xmlns:tasks="Tasks" xmlns="ItemOperations" targetNamespace="ItemOperations"
    elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import namespace="AirSyncBase" schemaLocation="AirSyncBase.xsd"/>
  <xs:import namespace="Calendar" schemaLocation="Calendar.xsd"/>
  <xs:import namespace="Contacts" schemaLocation="Contacts.xsd"/>
  <xs:import namespace="Contacts2" schemaLocation="Contacts2.xsd"/>
  <xs:import namespace="DocumentLibrary" schemaLocation="DocumentLibrary.xsd"/>
  <xs:import namespace="Email" schemaLocation="Email.xsd"/>
  <xs:import namespace="Email2" schemaLocation="Email2.xsd"/>
  <xs:import namespace="RightsManagement" schemaLocation=
      "RightsManagement.xsd"/>
  <xs:import namespace="Tasks" schemaLocation="Tasks.xsd"/>
  <xs:element name="Store">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:minLength value="1"/>
        <xs:maxLength value="256"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
```

Release: February 10, 2014

```
<xs:element name="Range">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[0-9]{1,9}-[0-9]{1,9}"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="Total" type="xs:integer"/>
<xs:element name="Properties">
  <xs:complexType>
    <xs:choice maxOccurs="unbounded">
      <xs:element ref="Range" minOccurs="0"/>
      <xs:element ref="Total" minOccurs="0"/>
      <xs:element ref="Data" minOccurs="0"/>
      <xs:element ref="Part" minOccurs="0"/>
      <xs:element ref="Version" minOccurs="0"/>
      <xs:element ref="airsyncbase:Body" minOccurs="0"/>
      <xs:element ref="airsyncbase:BodyPart" minOccurs="0"/>
      <xs:element ref="airsyncbase:Attachments" minOccurs="0"/>
      <xs:element ref="airsyncbase:NativeBodyType" minOccurs="0"/>
      <xs:group ref="calendar:AllProps"/>
      <xs:group ref="contacts:AllProps"/>
      <xs:group ref="contacts2:AllProps"/>
      <xs:group ref="documentlibrary:AllProps"/>
      <xs:group ref="email:AllProps"/>
      <xs:group ref="email2:AllProps"/>
      <xs:group ref="tasks:AllProps"/>
      <xs:element ref="rm:RightsManagementLicense" minOccurs="0"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
<xs:element name="Data" type="xs:string"/>
<xs:element name="Status" type="xs:integer"/>
<xs:element name="Version" type="xs:dateTime"/>
<xs:element name="Schema">
  <xs:complexType>
    <xs:choice maxOccurs="unbounded">
      <xs:group ref="email:TopLevelSchemaProps"/>
      <xs:group ref="airsyncbase:TopLevelSchemaProps"/>
      <xs:group ref="calendar:TopLevelSchemaProps"/>
      <xs:group ref="contacts:TopLevelSchemaProps"/>
      <xs:group ref="contacts2:TopLevelSchemaProps"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
<xs:element name="Part" type="xs:integer"/>
<xs:element name="DeleteSubFolders" type="airsyncbase:EmptyTag"/>
<xs:element name="UserName">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="100"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="Password">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="256"/>
    </xs:restriction>
```

```
        </xs:simpleType>
      </xs:element>
      <xs:element name="DstFldId">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:maxLength value="64"/>
            <xs:minLength value="1"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="ConversationId" type="xs:string"/>
      <xs:element name="MoveAlways" type="airsyncbase:EmptyTag"/>
    </xs:schema>
```

6.19 ItemOperations Request Schema

This section contains the contents of the ItemOperationsRequest.xsd file. The additional files that this schema file requires to operate correctly are listed in the following table.

File name	Defining section/specification
AirSync.xsd	6.1
AirSyncBase.xsd	[MS-ASAIRS] section 6
DocumentLibrary.xsd	[MS-ASDOC] section 6
ItemOperations.xsd	6.18
RightsManagement.xsd	[MS-ASRM] section 6
Search.xsd	6.29

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:airsyncbase=
    "AirSyncBase" xmlns:airsync="AirSync" xmlns:documentlibrary=
    "DocumentLibrary" xmlns:rm="RightsManagement" xmlns:search="Search"
    xmlns="ItemOperations" targetNamespace="ItemOperations"
    elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:include schemaLocation="ItemOperations.xsd"/>
  <xs:import namespace="AirSync" schemaLocation="AirSync.xsd"/>
  <xs:import namespace="AirSyncBase" schemaLocation="AirSyncBase.xsd"/>
  <xs:import namespace="DocumentLibrary" schemaLocation="DocumentLibrary.xsd"/>
  <xs:import namespace="RightsManagement" schemaLocation=
      "RightsManagement.xsd"/>
  <xs:import namespace="Search" schemaLocation="Search.xsd"/>
  <xs:element name="ItemOperations">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="EmptyFolderContents">
          <xs:complexType>
            <xs:all>
              <xs:element ref="airsync:CollectionId"/>
              <xs:element name="Options" minOccurs="0">
                <xs:complexType>
                  <xs:all>
                    <xs:element ref="DeleteSubFolders"/>
                  </xs:all>
```

Release: February 10, 2014

```
            </xs:complexType>
          </xs:element>
        </xs:all>
      </xs:complexType>
    </xs:element>
    <xs:element name="Fetch" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:all>
          <xs:element ref="Store"/>
          <xs:element ref="airsync:ServerId" minOccurs="0"/>
          <xs:element ref="airsync:CollectionId" minOccurs="0"/>
          <xs:element ref="documentlibrary:LinkId" minOccurs="0"/>
          <xs:element ref="search:LongId" minOccurs="0"/>
          <xs:element ref="airsyncbase:FileReference" minOccurs="0"/>
          <xs:element name="Options" minOccurs="0">
            <xs:complexType>
              <xs:choice maxOccurs="unbounded">
                <xs:element ref="Schema" minOccurs="0"
                    maxOccurs="unbounded"/>
                <xs:element ref="Range" minOccurs="0"/>
                <xs:element ref="UserName" minOccurs="0"/>
                <xs:element ref="Password" minOccurs="0"/>
                <xs:element ref="airsync:MIMESupport" minOccurs="0"/>
                <xs:element ref="airsyncbase:BodyPreference" minOccurs="0"
                    maxOccurs="256"/>
                <xs:element ref="airsyncbase:BodyPartPreference"
                    minOccurs="0"/>
                <xs:element ref="rm:RightsManagementSupport"
                    minOccurs="0"/>
              </xs:choice>
            </xs:complexType>
          </xs:element>
          <xs:element ref="rm:RemoveRightsManagementProtection"
              minOccurs="0"/>
        </xs:all>
      </xs:complexType>
    </xs:element>
    <xs:element name="Move">
      <xs:complexType>
        <xs:all>
          <xs:element ref="ConversationId"/>
          <xs:element ref="DstFldId"/>
          <xs:element name="Options" minOccurs="0">
            <xs:complexType>
              <xs:all>
                <xs:element ref="MoveAlways" minOccurs="0"/>
              </xs:all>
            </xs:complexType>
          </xs:element>
        </xs:all>
      </xs:complexType>
    </xs:element>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

6.20   ItemOperations Response Schema

This section contains the contents of the ItemOperationsResponse.xsd file. The additional files that this schema file requires to operate correctly are listed in the following table.

File name	Defining section/specification
AirSync.xsd	6.1
AirSyncBase.xsd	[MS-ASAIRS] section 6
DocumentLibrary.xsd	[MS-ASDOC] section 6
ItemOperations.xsd	6.18
Search.xsd	6.29

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:airsyncbase=
    "AirSyncBase" xmlns:airsync="AirSync" xmlns:documentlibrary=
    "DocumentLibrary" xmlns:search="Search" xmlns="ItemOperations"
    targetNamespace="ItemOperations" elementFormDefault="qualified"
    attributeFormDefault="unqualified">
  <xs:include schemaLocation="ItemOperations.xsd"/>
  <xs:import namespace="AirSync" schemaLocation="AirSync.xsd"/>
  <xs:import namespace="AirSyncBase" schemaLocation="AirSyncBase.xsd"/>
  <xs:import namespace="DocumentLibrary" schemaLocation="DocumentLibrary.xsd"/>
  <xs:import namespace="Search" schemaLocation="Search.xsd"/>
  <xs:element name="ItemOperations">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Status"/>
        <xs:element name="Response" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Move" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:all>
                    <xs:element ref="Status"/>
                    <xs:element ref="ConversationId"/>
                  </xs:all>
                </xs:complexType>
              </xs:element>
              <xs:element name="EmptyFolderContents" minOccurs="0"
                  maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element ref="Status"/>
                    <xs:element ref="airsync:CollectionId"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="Fetch" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element ref="Status"/>
                    <xs:element ref="airsync:CollectionId" minOccurs="0"/>
                    <xs:element ref="airsync:ServerId" minOccurs="0"/>
                    <xs:element ref="search:LongId" minOccurs="0"/>
```

```
                  <xs:element ref="airsync:Class" minOccurs="0"/>
                  <xs:element ref="documentlibrary:LinkId" minOccurs="0"/>
                  <xs:element ref="Properties" minOccurs="0"/>
               </xs:sequence>
            </xs:complexType>
         </xs:element>
      </xs:sequence>
   </xs:complexType>
 </xs:element>
</xs:schema>
```

6.21 MeetingResponse Request Schema

This section contains the contents of the MeetingResponseRequest.xsd file. The additional file that this schema file requires to operate correctly is listed in the following table.

File name	Defining section
Search.xsd	6.29

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:search="Search"
    xmlns="MeetingResponse" targetNamespace="MeetingResponse"
    elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import namespace="Search" schemaLocation="Search.xsd"/>
  <xs:element name="MeetingResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Request" maxOccurs="unbounded">
          <xs:complexType>
            <xs:all>
              <xs:element name="UserResponse">
                <xs:simpleType>
                  <xs:restriction base="xs:unsignedByte">
                    <xs:enumeration value="3"/>
                    <xs:enumeration value="1"/>
                    <xs:enumeration value="2"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
              <xs:element name="CollectionId" minOccurs="0">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <xs:maxLength value="64"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
              <xs:element name="RequestId" minOccurs="0">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <xs:maxLength value="64"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
```

```
                <xs:element ref="search:LongId" minOccurs="0"/>
                <xs:element name="InstanceId" minOccurs="0">
                  <xs:simpleType>
                    <xs:restriction base="xs:string">
                      <xs:minLength value="24"/>
                      <xs:maxLength value="24"/>
                    </xs:restriction>
                  </xs:simpleType>
                </xs:element>
              </xs:all>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:schema>
```

6.22 MeetingResponse Response Schema

This section contains the contents of the MeetingResponseResponse.xsd file.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="MeetingResponse"
    targetNamespace="MeetingResponse" elementFormDefault="qualified"
    attributeFormDefault="unqualified">
  <xs:element name="MeetingResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Result" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="RequestId" minOccurs="0">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <xs:maxLength value="64"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
              <xs:element name="Status" type="xs:unsignedByte"/>
              <xs:element name="CalendarId" minOccurs="0">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <xs:maxLength value="64"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

6.23   MoveItems Request Schema

This section contains the contents of the MoveItemsRequest.xsd file.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="Move"
    targetNamespace="Move" elementFormDefault="qualified"
    attributeFormDefault="unqualified">
  <xs:element name="MoveItems">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Move" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="SrcMsgId">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <xs:maxLength value="64"/>
                    <xs:minLength value="1"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
              <xs:element name="SrcFldId">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <xs:maxLength value="64"/>
                    <xs:minLength value="1"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
              <xs:element name="DstFldId">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <xs:maxLength value="64"/>
                    <xs:minLength value="1"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

6.24   MoveItems Response Schema

This section contains the contents of the MoveItemsResponse.xsd file.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="Move"
    targetNamespace="Move" elementFormDefault="qualified"
    attributeFormDefault="unqualified">
  <xs:element name="MoveItems">
    <xs:complexType>
      <xs:sequence>
```

```
              <xs:element name="Response" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="SrcMsgId">
                      <xs:simpleType>
                        <xs:restriction base="xs:string">
                          <xs:maxLength value="64"/>
                          <xs:minLength value="1"/>
                        </xs:restriction>
                      </xs:simpleType>
                    </xs:element>
                    <xs:element name="Status" type="xs:unsignedByte"/>
                    <xs:element name="DstMsgId" minOccurs="0">
                      <xs:simpleType>
                        <xs:restriction base="xs:string">
                          <xs:maxLength value="64"/>
                          <xs:minLength value="1"/>
                        </xs:restriction>
                      </xs:simpleType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:schema>
```

6.25 Ping Request Schema

This section contains the contents of the PingRequest.xsd file.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="Ping"
    targetNamespace="Ping" elementFormDefault="qualified"
    attributeFormDefault="unqualified">
  <xs:element name="Ping">
    <xs:complexType>
      <xs:all>
        <xs:element name="HeartbeatInterval" minOccurs="0">
          <xs:simpleType>
            <xs:restriction base="xs:integer"/>
          </xs:simpleType>
        </xs:element>
        <xs:element name="Folders" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Folder" minOccurs="1" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:all>
                    <xs:element name="Id">
                      <xs:simpleType>
                        <xs:restriction base="xs:string">
                          <xs:maxLength value="64"/>
                        </xs:restriction>
                      </xs:simpleType>
                    </xs:element>
```

```
                    <xs:element name="Class">
                      <xs:simpleType>
                        <xs:restriction base="xs:string">
                          <xs:enumeration value="Email"/>
                          <xs:enumeration value="Calendar"/>
                          <xs:enumeration value="Contacts"/>
                          <xs:enumeration value="Tasks"/>
                          <xs:enumeration value="Notes"/>
                        </xs:restriction>
                      </xs:simpleType>
                    </xs:element>
                  </xs:all>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:all>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

6.26   Ping Response Schema

This section contains the contents of the PingResponse.xsd file.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="Ping"
    targetNamespace="Ping" elementFormDefault="qualified"
    attributeFormDefault="unqualified">
  <xs:element name="Ping">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Status" type="xs:unsignedByte"/>
        <xs:element name="Folders" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Folder" type="xs:string"
                  maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="MaxFolders" type="xs:integer" minOccurs="0"/>
        <xs:element name="HeartbeatInterval" type="xs:integer" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

6.27   ResolveRecipients Request Schema

This section contains the contents of the ResolveRecipientsRequest.xsd file.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns=
```

```
          "ResolveRecipients" targetNamespace="ResolveRecipients"
      elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xs:element name="ResolveRecipients">
      <xs:complexType>
        <xs:choice maxOccurs="unbounded">
          <xs:element name="To" maxOccurs="1000">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:maxLength value="256"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element name="Options" minOccurs="0">
            <xs:complexType>
              <xs:all>
                <xs:element name="CertificateRetrieval" minOccurs="0">
                  <xs:simpleType>
                    <xs:restriction base="xs:integer">
                      <xs:minInclusive value="1"/>
                      <xs:maxInclusive value="3"/>
                    </xs:restriction>
                  </xs:simpleType>
                </xs:element>
                <xs:element name="MaxCertificates" minOccurs="0">
                  <xs:simpleType>
                    <xs:restriction base="xs:integer">
                      <xs:minInclusive value="0"/>
                      <xs:maxInclusive value="9999"/>
                    </xs:restriction>
                  </xs:simpleType>
                </xs:element>
                <xs:element name="MaxAmbiguousRecipients" minOccurs="0">
                  <xs:simpleType>
                    <xs:restriction base="xs:integer">
                      <xs:minInclusive value="0"/>
                      <xs:maxInclusive value="9999"/>
                    </xs:restriction>
                  </xs:simpleType>
                </xs:element>
                <xs:element name="Availability" minOccurs="0">
                  <xs:complexType>
                    <xs:all>
                      <xs:element name="StartTime" type="xs:string"/>
                      <xs:element name="EndTime" type="xs:string" minOccurs="0"/>
                    </xs:all>
                  </xs:complexType>
                </xs:element>
                <xs:element name="Picture" minOccurs="0">
                  <xs:complexType>
                    <xs:all>
                      <xs:element name="MaxSize" type="xs:unsignedInt"
                          minOccurs="0"/>
                      <xs:element name="MaxPictures" type="xs:unsignedInt"
                          minOccurs="0"/>
                    </xs:all>
                  </xs:complexType>
                </xs:element>
              </xs:all>
            </xs:complexType>
```

```
        </xs:element>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

6.28 ResolveRecipients Response Schema

This section contains the contents of the ResolveRecipientsResponse.xsd file.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns=
    "ResolveRecipients" targetNamespace="ResolveRecipients"
    elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="ResolveRecipients">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Status" type="xs:unsignedByte"/>
        <xs:element name="Response" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="To">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <xs:maxLength value="256"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
              <xs:element name="Status" type="xs:string"/>
              <xs:element name="RecipientCount" type="xs:integer"/>
              <xs:element name="Recipient" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Type" type="xs:unsignedByte"/>
                    <xs:element name="DisplayName" type="xs:string"/>
                    <xs:element name="EmailAddress" type="xs:string"/>
                    <xs:element name="Availability" minOccurs="0">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="Status" type="xs:string"/>
                          <xs:element name="MergedFreeBusy" type="xs:string"
                              minOccurs="0"/>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="Certificates" minOccurs="0">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="Status" type="xs:unsignedByte"/>
                          <xs:element name="CertificateCount" type="xs:integer"
                              minOccurs="0"/>
                          <xs:element name="RecipientCount" type="xs:integer"/>
                          <xs:element name="Certificate" type="xs:string"
                              minOccurs="0" maxOccurs="unbounded"/>
                          <xs:element name="MiniCertificate" type="xs:string"
                              minOccurs="0"/>
                        </xs:sequence>
```

```
                                      </xs:complexType>
                                    </xs:element>
                                    <xs:element name="Picture" minOccurs="0"
                                        maxOccurs="unbounded">
                                      <xs:complexType>
                                        <xs:all>
                                          <xs:element name="Status" type="xs:string"/>
                                          <xs:element name="Data" type="xs:string"
                                              minOccurs="0"/>
                                        </xs:all>
                                      </xs:complexType>
                                    </xs:element>
                                  </xs:sequence>
                                </xs:complexType>
                              </xs:element>
                            </xs:sequence>
                          </xs:complexType>
                        </xs:element>
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                </xs:schema>
```

6.29   Search Namespace Schema

This section contains the contents of the Search.xsd file. The additional files that this schema file requires to operate correctly are listed in the following table.

File name	Defining section/specification
AirSync.xsd	6.1
AirSyncBase.xsd	[MS-ASAIRS] section 6
DocumentLibrary.xsd	[MS-ASDOC] section 6
Email.xsd	[MS-ASEMAIL] section 6.1
RightsManagement.xsd	[MS-ASRM] section 6

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:airsyncbase=
    "AirSyncBase" xmlns:airsync="AirSync" xmlns:documentlibrary=
    "DocumentLibrary" xmlns:email="Email" xmlns:rm="RightsManagement"
    xmlns="Search" targetNamespace="Search" elementFormDefault="qualified"
    attributeFormDefault="unqualified">
  <xs:import namespace="AirSync" schemaLocation="AirSync.xsd"/>
  <xs:import namespace="AirSyncBase" schemaLocation="AirSyncBase.xsd"/>
  <xs:import namespace="DocumentLibrary" schemaLocation="DocumentLibrary.xsd"/>
  <xs:import namespace="Email" schemaLocation="Email.xsd"/>
  <xs:import namespace="RightsManagement"
      schemaLocation="RightsManagement.xsd"/>
  <xs:complexType name="queryType" mixed="true">
    <xs:sequence>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="And" type="queryType" minOccurs="0"/>
        <xs:element name="Or" type="queryType" minOccurs="0"/>
        <xs:element name="FreeText" type="xs:string" minOccurs="0"/>
```

```xml
                <xs:element ref="airsync:Class" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref="airsync:CollectionId" minOccurs="0"
                    maxOccurs="unbounded"/>
                <xs:element name="ConversationId" type="xs:string" minOccurs="0"/>
                <xs:element name="EqualTo" minOccurs="0">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element ref="documentlibrary:LinkId"/>
                      <xs:element ref="Value"/>
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
                <xs:element name="GreaterThan" minOccurs="0">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:group ref="email:ComparisonProps"/>
                      <xs:element ref="Value"/>
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
                <xs:element name="LessThan" minOccurs="0">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:group ref="email:ComparisonProps"/>
                      <xs:element ref="Value"/>
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
              </xs:choice>
            </xs:sequence>
          </xs:complexType>
          <xs:element name="Name">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:minLength value="1"/>
                <xs:maxLength value="256"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
          <xs:element name="Query" type="queryType"/>
          <xs:element name="Options">
            <xs:complexType>
              <xs:choice maxOccurs="unbounded">
                <xs:element ref="airsync:MIMESupport" minOccurs="0"/>
                <xs:element ref="airsyncbase:BodyPreference" minOccurs="0"
                    maxOccurs="unbounded"/>
                <xs:element ref="airsyncbase:BodyPartPreference" minOccurs="0"/>
                <xs:element ref="rm:RightsManagementSupport" minOccurs="0"/>
                <xs:element ref="Range"/>
                <xs:element name="UserName">
                  <xs:simpleType>
                    <xs:restriction base="xs:string">
                      <xs:maxLength value="100"/>
                    </xs:restriction>
                  </xs:simpleType>
                </xs:element>
                <xs:element name="Password">
                  <xs:simpleType>
                    <xs:restriction base="xs:string">
```

```
                <xs:maxLength value="100"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="DeepTraversal" type="airsyncbase:EmptyTag"/>
    <xs:element name="RebuildResults" type="airsyncbase:EmptyTag"/>
    <xs:element name="Picture" minOccurs="0">
        <xs:complexType>
            <xs:all>
                <xs:element name="MaxSize" type="xs:unsignedInt" minOccurs="0"/>
                <xs:element name="MaxPictures" type="xs:unsignedInt"
                    minOccurs="0"/>
            </xs:all>
        </xs:complexType>
    </xs:element>
    </xs:choice>
    </xs:complexType>
</xs:element>
<xs:element name="Status" type="xs:integer"/>
<xs:element name="Total" type="xs:integer"/>
<xs:element name="Value">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:maxLength value="1024"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="LongId">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:minLength value="1"/>
            <xs:maxLength value="256"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="Range">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:pattern value="[0-9]{1,3}-[0-9]{1,3}"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
</xs:schema>
```

6.30   Search Request Schema

This section contains the contents of the SearchRequest.xsd file. The additional file that this schema file requires to operate correctly is listed in the following table.

File name	Defining section
Search.xsd	[6.29](#)

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="Search"
    targetNamespace="Search" elementFormDefault="qualified"
    attributeFormDefault="unqualified">
```

```
<xs:include schemaLocation="Search.xsd"/>
<xs:element name="Search">
  <xs:complexType>
    <xs:choice maxOccurs="unbounded">
      <xs:element name="Store">
        <xs:complexType>
          <xs:all>
            <xs:element ref="Name"/>
            <xs:element name="Query" type="queryType"/>
            <xs:element ref="Options" minOccurs="0"/>
          </xs:all>
        </xs:complexType>
      </xs:element>
    </xs:choice>
  </xs:complexType>
</xs:element>
</xs:schema>
```

6.31   Search Response Schema

This section contains the contents of the SearchResponse.xsd file. The additional files that this schema file requires to operate correctly are listed in the following table.

File name	Defining section/specification
AirSync.xsd	6.1
AirSyncBase.xsd	[MS-ASAIRS] section 6
Calendar.xsd	[MS-ASCAL] section 6
Contacts.xsd	[MS-ASCNTC] section 6.1
Contacts2.xsd	[MS-ASCNTC] section 6.2
DocumentLibrary.xsd	[MS-ASDOC] section 6
Email.xsd	[MS-ASEMAIL] section 6.1
Email2.xsd	[MS-ASEMAIL] section 6.2
GAL.xsd	6.15
Notes.xsd	[MS-ASNOTE] section 6
Search.xsd	6.29
Tasks.xsd	[MS-ASTASK] section 6

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:airsyncbase=
    "AirSyncBase" xmlns:airsync="AirSync" xmlns:calendar="Calendar"
    xmlns:contacts="Contacts" xmlns:contacts2="Contacts2"
    xmlns:documentlibrary="DocumentLibrary" xmlns:email="Email"
    xmlns:email2="Email2" xmlns:gal="GAL" xmlns:notes="Notes"
    xmlns:tasks="Tasks" xmlns="Search" targetNamespace="Search"
    elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:include schemaLocation="Search.xsd"/>
```

```
<xs:import namespace="AirSync" schemaLocation="AirSync.xsd"/>
<xs:import namespace="AirSyncBase" schemaLocation="AirSyncBase.xsd"/>
<xs:import namespace="Calendar" schemaLocation="Calendar.xsd"/>
<xs:import namespace="Contacts" schemaLocation="Contacts.xsd"/>
<xs:import namespace="Contacts2" schemaLocation="Contacts2.xsd"/>
<xs:import namespace="DocumentLibrary" schemaLocation="DocumentLibrary.xsd"/>
<xs:import namespace="Email" schemaLocation="Email.xsd"/>
<xs:import namespace="Email2" schemaLocation="Email2.xsd"/>
<xs:import namespace="GAL" schemaLocation="GAL.xsd"/>
<xs:import namespace="Notes" schemaLocation="Notes.xsd"/>
<xs:import namespace="Tasks" schemaLocation="Tasks.xsd"/>
<xs:element name="Search">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Status" minOccurs="0"/>
      <xs:element name="Response" minOccurs="0">
        <xs:complexType>
          <xs:all>
            <xs:element name="Store">
              <xs:complexType>
                <xs:sequence>
                  <xs:element ref="Status"/>
                  <xs:element name="Result" nillable="true" minOccurs="1"
                      maxOccurs="unbounded">
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element ref="airsync:Class"/>
                        <xs:element ref="LongId" minOccurs="0"/>
                        <xs:element ref="airsync:CollectionId"/>
                        <xs:element name="Properties">
                          <xs:complexType>
                            <xs:group ref="SearchProperties"/>
                          </xs:complexType>
                        </xs:element>
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                  <xs:element ref="Range" minOccurs="0"/>
                  <xs:element ref="Total" minOccurs="0"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:all>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:group name="SearchProperties">
  <xs:sequence>
    <xs:choice maxOccurs="unbounded">
      <xs:element ref="airsyncbase:Attachments"/>
      <xs:element ref="airsyncbase:Body"/>
      <xs:element ref="airsyncbase:BodyPart"/>
      <xs:element ref="airsyncbase:NativeBodyType"/>
      <xs:element ref="calendar:Timezone"/>
      <xs:element ref="calendar:AllDayEvent"/>
      <xs:element ref="calendar:BusyStatus"/>
      <xs:element ref="calendar:OrganizerName"/>
```

```
<xs:element ref="calendar:OrganizerEmail"/>
<xs:element ref="calendar:DtStamp"/>
<xs:element ref="calendar:EndTime"/>
<xs:element ref="calendar:Location"/>
<xs:element ref="calendar:Reminder"/>
<xs:element ref="calendar:Sensitivity"/>
<xs:element ref="calendar:Subject"/>
<xs:element ref="calendar:StartTime"/>
<xs:element ref="calendar:UID"/>
<xs:element ref="calendar:MeetingStatus"/>
<xs:element ref="calendar:Attendees"/>
<xs:element ref="calendar:Recurrence"/>
<xs:element ref="calendar:Exceptions"/>
<xs:element ref="calendar:ResponseRequested"/>
<xs:element ref="calendar:AppointmentReplyTime"/>
<xs:element ref="calendar:ResponseType"/>
<xs:element ref="calendar:DisallowNewTimeProposal"/>
<xs:element ref="contacts:Anniversary"/>
<xs:element ref="contacts:AssistantName"/>
<xs:element ref="contacts:AssistantPhoneNumber"/>
<xs:element ref="contacts:Birthday"/>
<xs:element ref="contacts:Business2PhoneNumber"/>
<xs:element ref="contacts:BusinessAddressCity"/>
<xs:element ref="contacts:BusinessPhoneNumber"/>
<xs:element ref="contacts:WebPage"/>
<xs:element ref="contacts:BusinessAddressCountry"/>
<xs:element ref="contacts:Department"/>
<xs:element ref="contacts:Email1Address"/>
<xs:element ref="contacts:Email2Address"/>
<xs:element ref="contacts:Email3Address"/>
<xs:element ref="contacts:BusinessFaxNumber"/>
<xs:element ref="contacts:FileAs"/>
<xs:element ref="contacts:Alias"/>
<xs:element ref="contacts:WeightedRank"/>
<xs:element ref="contacts:FirstName"/>
<xs:element ref="contacts:MiddleName"/>
<xs:element ref="contacts:HomeAddressCity"/>
<xs:element ref="contacts:HomeAddressCountry"/>
<xs:element ref="contacts:HomeFaxNumber"/>
<xs:element ref="contacts:HomePhoneNumber"/>
<xs:element ref="contacts:Home2PhoneNumber"/>
<xs:element ref="contacts:HomeAddressPostalCode"/>
<xs:element ref="contacts:HomeAddressState"/>
<xs:element ref="contacts:HomeAddressStreet"/>
<xs:element ref="contacts:MobilePhoneNumber"/>
<xs:element ref="contacts:Suffix"/>
<xs:element ref="contacts:CompanyName"/>
<xs:element ref="contacts:OtherAddressCity"/>
<xs:element ref="contacts:OtherAddressCountry"/>
<xs:element ref="contacts:CarPhoneNumber"/>
<xs:element ref="contacts:OtherAddressPostalCode"/>
<xs:element ref="contacts:OtherAddressState"/>
<xs:element ref="contacts:OtherAddressStreet"/>
<xs:element ref="contacts:PagerNumber"/>
<xs:element ref="contacts:Title"/>
<xs:element ref="contacts:BusinessAddressPostalCode"/>
<xs:element ref="contacts:LastName"/>
<xs:element ref="contacts:Spouse"/>
<xs:element ref="contacts:BusinessAddressState"/>
```

```
<xs:element ref="contacts:BusinessAddressStreet"/>
<xs:element ref="contacts:JobTitle"/>
<xs:element ref="contacts:YomiFirstName"/>
<xs:element ref="contacts:YomiLastName"/>
<xs:element ref="contacts:YomiCompanyName"/>
<xs:element ref="contacts:OfficeLocation"/>
<xs:element ref="contacts:RadioPhoneNumber"/>
<xs:element ref="contacts:Picture"/>
<xs:element ref="contacts:Categories"/>
<xs:element ref="contacts:Children"/>
<xs:element ref="contacts2:CustomerId"/>
<xs:element ref="contacts2:GovernmentId"/>
<xs:element ref="contacts2:IMAddress"/>
<xs:element ref="contacts2:IMAddress2"/>
<xs:element ref="contacts2:IMAddress3"/>
<xs:element ref="contacts2:ManagerName"/>
<xs:element ref="contacts2:CompanyMainPhone"/>
<xs:element ref="contacts2:AccountName"/>
<xs:element ref="contacts2:NickName"/>
<xs:element ref="contacts2:MMS"/>
<xs:element ref="documentlibrary:LinkId"/>
<xs:element ref="documentlibrary:DisplayName"/>
<xs:element ref="documentlibrary:IsFolder"/>
<xs:element ref="documentlibrary:CreationDate"/>
<xs:element ref="documentlibrary:LastModifiedDate"/>
<xs:element ref="documentlibrary:IsHidden"/>
<xs:element ref="documentlibrary:ContentLength"/>
<xs:element ref="documentlibrary:ContentType"/>
<xs:element ref="email:To"/>
<xs:element ref="email:Cc"/>
<xs:element ref="email:From"/>
<xs:element ref="email:Subject"/>
<xs:element ref="email:ReplyTo"/>
<xs:element ref="email:DateReceived"/>
<xs:element ref="email:DisplayTo"/>
<xs:element ref="email:ThreadTopic"/>
<xs:element ref="email:Importance"/>
<xs:element ref="email:Read"/>
<xs:element ref="email:MessageClass"/>
<xs:element ref="email:MeetingRequest"/>
<xs:element ref="email:InternetCPID"/>
<xs:element ref="email:Flag"/>
<xs:element ref="email:ContentClass"/>
<xs:element ref="email2:UmCallerID"/>
<xs:element ref="email2:UmUserNotes"/>
<xs:element ref="email2:ConversationId"/>
<xs:element ref="email2:ConversationIndex"/>
<xs:element ref="email2:Sender"/>
<xs:element ref="email2:LastVerbExecuted"/>
<xs:element ref="email2:LastVerbExecutionTime"/>
<xs:element ref="email2:ReceivedAsBcc"/>
<xs:element ref="email:Categories"/>
<xs:element ref="notes:Subject"/>
<xs:element ref="notes:MessageClass"/>
<xs:element ref="notes:LastModifiedDate"/>
<xs:element ref="notes:Categories"/>
<xs:element ref="tasks:Subject"/>
<xs:element ref="tasks:Importance"/>
<xs:element ref="tasks:UtcStartDate"/>
```

```
            <xs:element ref="tasks:StartDate"/>
            <xs:element ref="tasks:UtcDueDate"/>
            <xs:element ref="tasks:DueDate"/>
            <xs:element ref="tasks:Categories"/>
            <xs:element ref="tasks:Recurrence"/>
            <xs:element ref="tasks:Complete"/>
            <xs:element ref="tasks:DateCompleted"/>
            <xs:element ref="tasks:Sensitivity"/>
            <xs:element ref="tasks:ReminderTime"/>
            <xs:element ref="tasks:ReminderSet"/>
            <xs:group ref="gal:TopLevelSchemaProps"/>
         </xs:choice>
      </xs:sequence>
   </xs:group>
</xs:schema>
```

6.32   SendMail Request Schema

This section contains the contents of the SendMailRequest.xsd file. The additional files that this schema file requires to operate correctly are listed in the following table.

File name	Defining section/specification
ComposeMail.xsd	6.5
RightsManagement.xsd	[MS-ASRM] section 6

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:rm=
    "RightsManagement" xmlns="ComposeMail" targetNamespace="ComposeMail"
    elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:include schemaLocation="ComposeMail.xsd"/>
  <xs:import namespace="RightsManagement" schemaLocation=
      "RightsManagement.xsd"/>
  <xs:element name="SendMail">
    <xs:complexType>
      <xs:all>
        <xs:element ref="ClientId"/>
        <xs:element ref="AccountId" minOccurs="0"/>
        <xs:element ref="SaveInSentItems" minOccurs="0"/>
        <xs:element ref="Mime"/>
        <xs:element ref="rm:TemplateID" minOccurs="0"/>
      </xs:all>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

6.33   SendMail Response Schema

This section contains the contents of the SendMailResponse.xsd file. The additional file that this schema file requires to operate correctly is listed in the following table.

File name	Defining section
ComposeMail.xsd	6.5

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="ComposeMail"
    targetNamespace="ComposeMail" elementFormDefault="qualified"
    attributeFormDefault="unqualified">
  <xs:include schemaLocation="ComposeMail.xsd"/>
  <xs:element name="SendMail">
    <xs:complexType>
      <xs:all>
        <xs:element ref="Status" minOccurs="0"/>
      </xs:all>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

6.34 Settings Namespace Schema

This section contains the contents of the Settings.xsd file. The additional file that this schema file requires to operate correctly is listed in the following table.

File name	Defining specification
AirSyncBase.xsd	[MS-ASAIRS] section 6

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:airsyncbase=
    "AirSyncBase" xmlns="Settings" targetNamespace="Settings"
    elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import namespace="AirSyncBase" schemaLocation="AirSyncBase.xsd"/>
  <xs:simpleType name="DeviceInformationStringType">
    <xs:restriction base="xs:string">
      <xs:maxLength value="1024"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="NonEmptyStringType">
    <xs:restriction base="xs:string">
      <xs:minLength value="1"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="Status" type="xs:integer"/>
  <xs:element name="OofState">
    <xs:simpleType>
      <xs:restriction base="xs:integer">
        <xs:enumeration value="0"/>
        <xs:enumeration value="1"/>
        <xs:enumeration value="2"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="StartTime" type="xs:dateTime"/>
  <xs:element name="EndTime" type="xs:dateTime"/>
  <xs:element name="OofMessage">
    <xs:complexType>
      <xs:all>
        <xs:element name="AppliesToInternal" type="airsyncbase:EmptyTag"
            minOccurs="0"/>
        <xs:element name="AppliesToExternalKnown" type="airsyncbase:EmptyTag"
            minOccurs="0"/>
```

Release: February 10, 2014

```xml
            <xs:element name="AppliesToExternalUnknown" type="airsyncbase:EmptyTag"
                minOccurs="0"/>
            <xs:element name="Enabled" type="NonEmptyStringType" minOccurs="0"/>
            <xs:element name="ReplyMessage" type="xs:string" minOccurs="0"/>
            <xs:element ref="BodyType" minOccurs="0"/>
        </xs:all>
    </xs:complexType>
</xs:element>
<xs:element name="BodyType" type="NonEmptyStringType"/>
<xs:element name="Password">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="255"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="Model" type="DeviceInformationStringType"/>
<xs:element name="IMEI" type="DeviceInformationStringType"/>
<xs:element name="FriendlyName" type="DeviceInformationStringType"/>
<xs:element name="OS" type="DeviceInformationStringType"/>
<xs:element name="OSLanguage" type="DeviceInformationStringType"/>
<xs:element name="PhoneNumber" type="DeviceInformationStringType"/>
<xs:element name="UserAgent" type="DeviceInformationStringType"/>
<xs:element name="EnableOutboundSMS">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="1"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="MobileOperator" type="DeviceInformationStringType"/>
<xs:element name="Accounts">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Account" maxOccurs="unbounded">
        <xs:complexType>
          <xs:all>
            <xs:element name="AccountId" type="xs:string" minOccurs="0"/>
            <xs:element name="AccountName" type="xs:string" minOccurs="0"/>
            <xs:element name="UserDisplayName" type="xs:string"
                minOccurs="0"/>
            <xs:element name="SendDisabled" type="xs:boolean" minOccurs="0"/>
            <xs:element name="EmailAddresses" minOccurs="0">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="SMTPAddress" type="xs:string"
                      minOccurs="0" maxOccurs="unbounded"/>
                  <xs:element name="PrimarySmtpAddress" type="xs:string"
                      minOccurs="0"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:all>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
        </xs:schema>
```

6.35   Settings Request Schema

This section contains the contents of the SettingsRequest.xsd file. The additional files that this schema file requires to operate correctly are listed in the following table.

File name	Defining section/specification
AirSyncBase.xsd	[MS-ASAIRS] section 6
Settings.xsd	6.34

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:airsyncbase=
    "AirSyncBase" xmlns="Settings" targetNamespace="Settings"
    elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:include schemaLocation="Settings.xsd"/>
  <xs:import namespace="AirSyncBase" schemaLocation="AirSyncBase.xsd"/>
  <xs:element name="DeviceInformation">
    <xs:complexType>
      <xs:all>
        <xs:element name="Set">
          <xs:complexType>
            <xs:all>
              <xs:element ref="Model" minOccurs="0"/>
              <xs:element ref="IMEI" minOccurs="0"/>
              <xs:element ref="FriendlyName" minOccurs="0"/>
              <xs:element ref="OS" minOccurs="0"/>
              <xs:element ref="OSLanguage" minOccurs="0"/>
              <xs:element ref="PhoneNumber" minOccurs="0"/>
              <xs:element ref="UserAgent" minOccurs="0"/>
              <xs:element ref="EnableOutboundSMS" minOccurs="0"/>
              <xs:element ref="MobileOperator" minOccurs="0"/>
            </xs:all>
          </xs:complexType>
        </xs:element>
      </xs:all>
    </xs:complexType>
  </xs:element>
  <xs:element name="Settings">
    <xs:complexType>
      <xs:all>
        <xs:element name="RightsManagementInformation" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Get" type="airsyncbase:EmptyTag"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="Oof" minOccurs="0">
          <xs:complexType>
            <xs:choice>
              <xs:element name="Get" minOccurs="0">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element ref="BodyType"/>
```

```
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="Set">
            <xs:complexType>
              <xs:sequence>
                <xs:element ref="OofState" minOccurs="0"/>
                <xs:element ref="StartTime" minOccurs="0"/>
                <xs:element ref="EndTime" minOccurs="0"/>
                <xs:element ref="OofMessage" minOccurs="0" maxOccurs="3"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
    <xs:element name="DevicePassword" minOccurs="0">
      <xs:complexType>
        <xs:choice>
          <xs:element name="Set">
            <xs:complexType>
              <xs:sequence>
                <xs:element ref="Password"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
    <xs:element ref="DeviceInformation" minOccurs="0"/>
    <xs:element name="UserInformation" minOccurs="0">
      <xs:complexType>
        <xs:choice>
          <xs:element name="Get" type="airsyncbase:EmptyTag"/>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:all>
</xs:complexType>
</xs:element>
</xs:schema>
```

6.36 Settings Response Schema

This section contains the contents of the SettingsResponse.xsd file. The additional files that this
schema file requires to operate correctly are listed in the following table.

File name	Defining section/specification
RightsManagement.xsd	[MS-ASRM] section 6
Settings.xsd	6.34

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:rm=
    "RightsManagement" xmlns="Settings" targetNamespace="Settings"
    elementFormDefault="qualified" attributeFormDefault="unqualified">
```

```
<xs:include schemaLocation="Settings.xsd"/>
<xs:import namespace="RightsManagement" schemaLocation=
    "RightsManagement.xsd"/>
<xs:element name="DeviceInformation">
  <xs:complexType>
    <xs:all>
      <xs:element ref="Status" minOccurs="0"/>
    </xs:all>
  </xs:complexType>
</xs:element>
<xs:element name="Settings">
  <xs:complexType>
    <xs:all>
      <xs:element ref="Status" minOccurs="0"/>
      <xs:element name="Oof" minOccurs="0">
        <xs:complexType>
          <xs:all>
            <xs:element ref="Status" minOccurs="0"/>
            <xs:element name="Get" minOccurs="0">
              <xs:complexType>
                <xs:sequence>
                  <xs:element ref="OofState" minOccurs="0"/>
                  <xs:element ref="StartTime"/>
                  <xs:element ref="EndTime"/>
                  <xs:element ref="OofMessage" minOccurs="0" maxOccurs="3"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:all>
        </xs:complexType>
      </xs:element>
      <xs:element ref="DeviceInformation" minOccurs="0"/>
      <xs:element name="DevicePassword" minOccurs="0">
        <xs:complexType>
          <xs:all>
            <xs:element ref="Status" minOccurs="0"/>
          </xs:all>
        </xs:complexType>
      </xs:element>
      <xs:element name="UserInformation" minOccurs="0">
        <xs:complexType>
          <xs:all>
            <xs:element ref="Status" minOccurs="0"/>
            <xs:element name="Get" minOccurs="0">
              <xs:complexType>
                <xs:all>
                  <xs:element ref="Accounts" minOccurs="0"/>
                </xs:all>
              </xs:complexType>
            </xs:element>
          </xs:all>
        </xs:complexType>
      </xs:element>
      <xs:element name="RightsManagementInformation" minOccurs="0">
        <xs:complexType>
          <xs:all>
            <xs:element ref="Status" minOccurs="0"/>
            <xs:element name="Get">
              <xs:complexType>
```

```
            <xs:all>
              <xs:element ref="rm:RightsManagementTemplates"/>
            </xs:all>
          </xs:complexType>
        </xs:element>
      </xs:all>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

6.37 SmartForward Request Schema

This section contains the contents of the SmartForwardRequest.xsd file. The additional files that this
schema file requires to operate correctly are listed in the following table.

File name	Defining section/specification
ComposeMail.xsd	6.5
RightsManagement.xsd	[MS-ASRM] section 6

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:rm=
    "RightsManagement" xmlns="ComposeMail" targetNamespace="ComposeMail"
    elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:include schemaLocation="ComposeMail.xsd"/>
  <xs:import namespace="RightsManagement" schemaLocation=
      "RightsManagement.xsd"/>
  <xs:element name="SmartForward">
    <xs:complexType>
      <xs:all>
        <xs:element ref="ClientId"/>
        <xs:element ref="Source"/>
        <xs:element ref="AccountId" minOccurs="0"/>
        <xs:element ref="SaveInSentItems" minOccurs="0"/>
        <xs:element ref="ReplaceMime" minOccurs="0"/>
        <xs:element ref="Mime"/>
        <xs:element ref="rm:TemplateID" minOccurs="0"/>
      </xs:all>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

6.38 SmartForward Response Schema

This section contains the contents of the SmartForwardResponse.xsd file. The additional file that this
schema file requires to operate correctly is listed in the following table.

File name	Defining section
ComposeMail.xsd	6.5

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="ComposeMail"
    targetNamespace="ComposeMail" elementFormDefault="qualified"
    attributeFormDefault="unqualified">
  <xs:include schemaLocation="ComposeMail.xsd"/>
  <xs:element name="SmartForward">
    <xs:complexType>
      <xs:all>
        <xs:element ref="Status" minOccurs="0"/>
      </xs:all>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

6.39 SmartReply Request Schema

This section contains the contents of the SmartReplyRequest.xsd file. The additional files that this schema file requires to operate correctly are listed in the following table.

File name	Defining section/specification
ComposeMail.xsd	6.5
RightsManagement.xsd	[MS-ASRM] section 6

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:rm=
    "RightsManagement" xmlns="ComposeMail" targetNamespace="ComposeMail"
    elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:include schemaLocation="ComposeMail.xsd"/>
  <xs:import namespace="RightsManagement" schemaLocation=
      "RightsManagement.xsd"/>
  <xs:element name="SmartReply">
    <xs:complexType>
      <xs:all>
        <xs:element ref="ClientId"/>
        <xs:element ref="Source"/>
        <xs:element ref="AccountId" minOccurs="0"/>
        <xs:element ref="SaveInSentItems" minOccurs="0"/>
        <xs:element ref="ReplaceMime" minOccurs="0"/>
        <xs:element ref="Mime"/>
        <xs:element ref="rm:TemplateID" minOccurs="0"/>
      </xs:all>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

6.40 SmartReply Response Schema

This section contains the contents of the SmartReplyResponse.xsd file. The additional file that this schema file requires to operate correctly is listed in the following table.

File name	Defining section
ComposeMail.xsd	6.5

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="ComposeMail"
    targetNamespace="ComposeMail" elementFormDefault="qualified"
    attributeFormDefault="unqualified">
  <xs:include schemaLocation="ComposeMail.xsd"/>
  <xs:element name="SmartReply">
    <xs:complexType>
      <xs:all>
        <xs:element ref="Status" minOccurs="0"/>
      </xs:all>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

6.41   Sync Request Schema

This section contains the contents of the SyncRequest.xsd file. The additional files that this schema file requires to operate correctly are listed in the following table.

File name	Defining section/specification
AirSync.xsd	6.1
AirSyncBase.xsd	[MS-ASAIRS] section 6
Calendar.xsd	[MS-ASCAL] section 6
Contacts.xsd	[MS-ASCNTC] section 6.1
Contacts2.xsd	[MS-ASCNTC] section 6.2
Email.xsd	[MS-ASEMAIL] section 6.1
Notes.xsd	[MS-ASNOTE] section 6
Tasks.xsd	[MS-ASTASK] section 6

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:airsyncbase=
    "AirSyncBase" xmlns:calendar="Calendar" xmlns:contacts="Contacts"
    xmlns:contacts2="Contacts2" xmlns:email="Email" xmlns:notes="Notes"
    xmlns:tasks="Tasks" xmlns="AirSync" targetNamespace="AirSync"
    elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:include schemaLocation="AirSync.xsd"/>
  <xs:import namespace="AirSyncBase" schemaLocation="AirSyncBase.xsd"/>
  <xs:import namespace="Calendar" schemaLocation="Calendar.xsd"/>
  <xs:import namespace="Contacts" schemaLocation="Contacts.xsd"/>
  <xs:import namespace="Contacts2" schemaLocation="Contacts2.xsd"/>
  <xs:import namespace="Email" schemaLocation="Email.xsd"/>
  <xs:import namespace="Notes" schemaLocation="Notes.xsd"/>
  <xs:import namespace="Tasks" schemaLocation="Tasks.xsd"/>
  <xs:element name="Sync">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Collections" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Collection" maxOccurs="unbounded">
                <xs:complexType>
```

```
<xs:sequence>
  <xs:element ref="SyncKey"/>
  <xs:element ref="CollectionId"/>
  <xs:element ref="Supported" minOccurs="0"/>
  <xs:element ref="DeletesAsMoves" minOccurs="0"/>
  <xs:element ref="GetChanges" minOccurs="0"/>
  <xs:element ref="WindowSize" minOccurs="0"/>
  <xs:element name="ConversationMode" type="xs:boolean"
      minOccurs="0"/>
  <xs:element ref="Options" minOccurs="0" maxOccurs="2"/>
  <xs:element name="Commands" minOccurs="0">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="Change" minOccurs="0"
            maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element ref="ServerId"/>
              <xs:element name="ApplicationData">
                <xs:complexType>
                  <xs:sequence>
                    <xs:choice maxOccurs="unbounded">
                      <xs:group ref="ItemProperties"/>
                      <xs:element ref="email:Categories"/>
                    </xs:choice>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="Delete" minOccurs="0"
            maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element ref="ServerId"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="Add" minOccurs="0"
            maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element ref="Class" minOccurs="0"/>
              <xs:element ref="ClientId"/>
              <xs:element name="ApplicationData">
                <xs:complexType>
                  <xs:sequence>
                    <xs:choice maxOccurs="unbounded">
                      <xs:group ref="ItemProperties"/>
                      <xs:element ref="email:To"/>
                      <xs:element ref="email:From"/>
                      <xs:element ref="email:DateReceived"/>
                      <xs:element ref="email:InternetCPID"/>
                      <xs:element ref="email:Importance"/>
                    </xs:choice>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
```

```
                                    </xs:sequence>
                                  </xs:complexType>
                                </xs:element>
                                <xs:element name="Fetch" minOccurs="0"
                                    maxOccurs="unbounded">
                                  <xs:complexType>
                                    <xs:sequence>
                                      <xs:element ref="ServerId"/>
                                    </xs:sequence>
                                  </xs:complexType>
                                </xs:element>
                              </xs:choice>
                            </xs:complexType>
                          </xs:element>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element ref="Wait" minOccurs="0"/>
              <xs:element ref="HeartbeatInterval" minOccurs="0"/>
              <xs:element ref="WindowSize" minOccurs="0"/>
              <xs:element ref="Partial" minOccurs="0"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:group name="ItemProperties">
          <xs:sequence>
            <xs:choice maxOccurs="unbounded">
              <xs:element ref="airsyncbase:NativeBodyType"/>
              <xs:element ref="airsyncbase:Body"/>
              <xs:element ref="calendar:OrganizerName"/>
              <xs:element ref="calendar:OrganizerEmail"/>
              <xs:element ref="calendar:Exceptions"/>
              <xs:element ref="calendar:Attendees"/>
              <xs:element ref="calendar:DisallowNewTimeProposal"/>
              <xs:element ref="calendar:ResponseRequested"/>
              <xs:element ref="calendar:Timezone"/>
              <xs:element ref="calendar:AllDayEvent"/>
              <xs:element ref="calendar:BusyStatus"/>
              <xs:element ref="calendar:Categories"/>
              <xs:element ref="calendar:DtStamp"/>
              <xs:element ref="calendar:EndTime"/>
              <xs:element ref="calendar:Location"/>
              <xs:element ref="calendar:MeetingStatus"/>
              <xs:element ref="calendar:Reminder"/>
              <xs:element ref="calendar:Sensitivity"/>
              <xs:element ref="calendar:Subject"/>
              <xs:element ref="calendar:StartTime"/>
              <xs:element ref="calendar:UID"/>
              <xs:element ref="calendar:Recurrence"/>
              <xs:element ref="contacts:Anniversary"/>
              <xs:element ref="contacts:AssistantName"/>
              <xs:element ref="contacts:AssistantPhoneNumber"/>
              <xs:element ref="contacts:Birthday"/>
              <xs:element ref="contacts:Business2PhoneNumber"/>
              <xs:element ref="contacts:BusinessAddressCity"/>
              <xs:element ref="contacts:BusinessAddressCountry"/>
```

```
<xs:element ref="contacts:BusinessAddressPostalCode"/>
<xs:element ref="contacts:BusinessAddressState"/>
<xs:element ref="contacts:BusinessAddressStreet"/>
<xs:element ref="contacts:BusinessFaxNumber"/>
<xs:element ref="contacts:BusinessPhoneNumber"/>
<xs:element ref="contacts:CarPhoneNumber"/>
<xs:element ref="contacts:Categories"/>
<xs:element ref="contacts:Children"/>
<xs:element ref="contacts:CompanyName"/>
<xs:element ref="contacts:Department"/>
<xs:element ref="contacts:Email1Address"/>
<xs:element ref="contacts:Email2Address"/>
<xs:element ref="contacts:Email3Address"/>
<xs:element ref="contacts:FileAs"/>
<xs:element ref="contacts:FirstName"/>
<xs:element ref="contacts:MiddleName"/>
<xs:element ref="contacts:Home2PhoneNumber"/>
<xs:element ref="contacts:HomeAddressCity"/>
<xs:element ref="contacts:HomeAddressCountry"/>
<xs:element ref="contacts:HomeAddressPostalCode"/>
<xs:element ref="contacts:HomeAddressState"/>
<xs:element ref="contacts:HomeAddressStreet"/>
<xs:element ref="contacts:HomeFaxNumber"/>
<xs:element ref="contacts:HomePhoneNumber"/>
<xs:element ref="contacts:JobTitle"/>
<xs:element ref="contacts:LastName"/>
<xs:element ref="contacts:MobilePhoneNumber"/>
<xs:element ref="contacts:OfficeLocation"/>
<xs:element ref="contacts:OtherAddressCity"/>
<xs:element ref="contacts:OtherAddressCountry"/>
<xs:element ref="contacts:OtherAddressPostalCode"/>
<xs:element ref="contacts:OtherAddressState"/>
<xs:element ref="contacts:OtherAddressStreet"/>
<xs:element ref="contacts:PagerNumber"/>
<xs:element ref="contacts:RadioPhoneNumber"/>
<xs:element ref="contacts:Spouse"/>
<xs:element ref="contacts:Suffix"/>
<xs:element ref="contacts:Title"/>
<xs:element ref="contacts:WebPage"/>
<xs:element ref="contacts:YomiCompanyName"/>
<xs:element ref="contacts:YomiFirstName"/>
<xs:element ref="contacts:YomiLastName"/>
<xs:element ref="contacts:Picture"/>
<xs:element ref="contacts2:CustomerId"/>
<xs:element ref="contacts2:GovernmentId"/>
<xs:element ref="contacts2:IMAddress"/>
<xs:element ref="contacts2:IMAddress2"/>
<xs:element ref="contacts2:IMAddress3"/>
<xs:element ref="contacts2:ManagerName"/>
<xs:element ref="contacts2:CompanyMainPhone"/>
<xs:element ref="contacts2:AccountName"/>
<xs:element ref="contacts2:NickName"/>
<xs:element ref="contacts2:MMS"/>
<xs:element ref="email:Flag"/>
<xs:element ref="email:Read"/>
<xs:element ref="notes:Subject"/>
<xs:element ref="notes:MessageClass"/>
<xs:element ref="notes:LastModifiedDate"/>
<xs:element ref="notes:Categories"/>
```

```
                <xs:element ref="tasks:Complete"/>
                <xs:element ref="tasks:Subject"/>
                <xs:element ref="tasks:Categories"/>
                <xs:element ref="tasks:DateCompleted"/>
                <xs:element ref="tasks:DueDate"/>
                <xs:element ref="tasks:UtcDueDate"/>
                <xs:element ref="tasks:Importance"/>
                <xs:element ref="tasks:Recurrence"/>
                <xs:element ref="tasks:ReminderSet"/>
                <xs:element ref="tasks:ReminderTime"/>
                <xs:element ref="tasks:Sensitivity"/>
                <xs:element ref="tasks:StartDate"/>
                <xs:element ref="tasks:UtcStartDate"/>
            </xs:choice>
        </xs:sequence>
    </xs:group>
</xs:schema>
```

6.42   Sync Response Schema

This section contains the contents of the SyncResponse.xsd file. The additional files that this schema file requires to operate correctly are listed in the following table.

File name	Defining section/specification
AirSync.xsd	6.1
AirSyncBase.xsd	[MS-ASAIRS] section 6
Calendar.xsd	[MS-ASCAL] section 6
Contacts.xsd	[MS-ASCNTC] section 6.1
Contacts2.xsd	[MS-ASCNTC] section 6.2
DocumentLibrary.xsd	[MS-ASDOC] section 6
Email.xsd	[MS-ASEMAIL] section 6.1
Email2.xsd	[MS-ASEMAIL] section 6.2
RightsManagement.xsd	[MS-ASRM] section 6
Tasks.xsd	[MS-ASTASK] section 6

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:airsyncbase=
    "AirSyncBase" xmlns:calendar="Calendar" xmlns:contacts="Contacts"
    xmlns:contacts2="Contacts2" xmlns:documentlibrary="DocumentLibrary"
    xmlns:email="Email" xmlns:email2="Email2" xmlns:rm="RightsManagement"
    xmlns:tasks="Tasks" xmlns="AirSync" targetNamespace="AirSync"
    elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:include schemaLocation="AirSync.xsd"/>
  <xs:import namespace="AirSyncBase" schemaLocation="AirSyncBase.xsd"/>
  <xs:import namespace="Calendar" schemaLocation="Calendar.xsd"/>
  <xs:import namespace="Contacts" schemaLocation="Contacts.xsd"/>
  <xs:import namespace="Contacts2" schemaLocation="Contacts2.xsd"/>
  <xs:import namespace="DocumentLibrary" schemaLocation="DocumentLibrary.xsd"/>
```

Release: February 10, 2014

```
<xs:import namespace="Email" schemaLocation="Email.xsd"/>
<xs:import namespace="Email2" schemaLocation="Email2.xsd"/>
<xs:import namespace="RightsManagement" schemaLocation=
    "RightsManagement.xsd"/>
<xs:import namespace="Tasks" schemaLocation="Tasks.xsd"/>
<xs:element name="Sync">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Status" minOccurs="0"/>
      <xs:choice>
        <xs:element ref="Limit" minOccurs="0"/>
        <xs:element name="Collections" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Collection" minOccurs="0"
                  maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>
                    <xs:choice maxOccurs="unbounded">
                      <xs:element ref="SyncKey"/>
                      <xs:element ref="CollectionId"/>
                      <xs:element ref="Status"/>
                      <xs:element name="Commands" minOccurs="0">
                        <xs:complexType>
                          <xs:sequence>
                            <xs:element name="Delete" minOccurs="0"
                                maxOccurs="unbounded">
                              <xs:complexType>
                                <xs:sequence>
                                  <xs:element ref="Class" minOccurs="0"/>
                                  <xs:element ref="ServerId"/>
                                </xs:sequence>
                              </xs:complexType>
                            </xs:element>
                            <xs:element name="SoftDelete" minOccurs="0"
                                maxOccurs="unbounded">
                              <xs:complexType>
                                <xs:sequence>
                                  <xs:element ref="ServerId"/>
                                </xs:sequence>
                              </xs:complexType>
                            </xs:element>
                            <xs:element name="Change" minOccurs="0"
                                maxOccurs="unbounded">
                              <xs:complexType>
                                <xs:sequence>
                                  <xs:element ref="Class" minOccurs="0"/>
                                  <xs:element ref="ServerId"/>
                                  <xs:element name="ApplicationData">
                                    <xs:complexType>
                                      <xs:choice maxOccurs="unbounded">
                                        <xs:element ref="airsyncbase:Body"/>
                                        <xs:element
                                            ref="airsyncbase:BodyPart"/>
                                        <xs:element
                                            ref="airsyncbase:NativeBodyType"/>
                                        <xs:group ref="calendar:AllProps"/>
                                        <xs:group ref="contacts:AllProps"/>
                                        <xs:group ref="contacts2:AllProps"/>
```

```xml
                        <xs:group
                            ref="documentlibrary:AllProps"/>
                        <xs:group ref="email:AllProps"/>
                        <xs:group ref="email2:AllProps"/>
                        <xs:group ref="tasks:AllProps"/>
                        <xs:element
                            ref="rm:RightsManagementLicense"/>
                    </xs:choice>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="Add" minOccurs="0"
    maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="ServerId"/>
      <xs:element name="ApplicationData">
        <xs:complexType>
          <xs:sequence>
            <xs:choice maxOccurs="unbounded">
              <xs:element
                  ref="airsyncbase:Body"/>
              <xs:element
                  ref="airsyncbase:BodyPart"/>
              <xs:element
                  ref="airsyncbase:NativeBodyType"/>
              <xs:group ref="calendar:AllProps"/>
              <xs:group ref="contacts:AllProps"/>
              <xs:group
                  ref="contacts2:AllProps"/>
              <xs:group
                  ref="documentlibrary:AllProps"/>
              <xs:group ref="email:AllProps"/>
              <xs:group ref="email2:AllProps"/>
              <xs:group ref="tasks:AllProps"/>
              <xs:element
                  ref="rm:RightsManagementLicense"/>
            </xs:choice>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Responses" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Change" minOccurs="0"
          maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element ref="Class" minOccurs="0"/>
            <xs:element ref="ServerId"/>
            <xs:element ref="Status"/>
```

```
                              </xs:sequence>
                          </xs:complexType>
                      </xs:element>
                      <xs:element name="Add" minOccurs="0"
                          maxOccurs="unbounded">
                        <xs:complexType>
                          <xs:sequence>
                            <xs:element ref="Class" minOccurs="0"/>
                            <xs:element ref="ClientId"/>
                            <xs:element ref="ServerId" minOccurs="0"/>
                            <xs:element ref="Status"/>
                          </xs:sequence>
                        </xs:complexType>
                      </xs:element>
                      <xs:element name="Fetch" minOccurs="0">
                        <xs:complexType>
                          <xs:sequence>
                            <xs:element ref="ServerId" minOccurs="0"/>
                            <xs:element ref="Status"/>
                            <xs:element name="ApplicationData">
                              <xs:complexType>
                                <xs:sequence>
                                  <xs:choice maxOccurs="unbounded">
                                    <xs:element
                                        ref="airsyncbase:Attachments"/>
                                    <xs:element
                                        ref="airsyncbase:Body"/>
                                    <xs:element
                                        ref="airsyncbase:NativeBodyType"/>
                                    <xs:group ref="calendar:AllProps"/>
                                    <xs:group ref="contacts:AllProps"/>
                                    <xs:group
                                        ref="contacts2:AllProps"/>
                                    <xs:group
                                        ref="documentlibrary:AllProps"/>
                                    <xs:group ref="email:AllProps"/>
                                    <xs:group ref="email2:AllProps"/>
                                    <xs:group ref="tasks:AllProps"/>
                                    <xs:element
                                        ref="rm:RightsManagementLicense"/>
                                  </xs:choice>
                                </xs:sequence>
                              </xs:complexType>
                            </xs:element>
                          </xs:sequence>
                        </xs:complexType>
                      </xs:element>
                      <xs:element ref="MoreAvailable" minOccurs="0"/>
                    </xs:choice>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
```

```
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:schema>
```

6.43 ValidateCert Request Schema

This section contains the contents of the ValidateCertRequest.xsd file.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="ValidateCert"
    targetNamespace="ValidateCert" elementFormDefault="qualified"
    attributeFormDefault="unqualified">
  <xs:element name="ValidateCert">
    <xs:complexType>
      <xs:all minOccurs="0">
        <xs:element name="CertificateChain" minOccurs="0">
          <xs:complexType>
            <xs:choice maxOccurs="unbounded">
              <xs:element name="Certificate" maxOccurs="unbounded">
                <xs:simpleType>
                  <xs:restriction base="xs:base64Binary">
                    <xs:minLength value="4"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
            </xs:choice>
          </xs:complexType>
        </xs:element>
        <xs:element name="Certificates">
          <xs:complexType>
            <xs:choice maxOccurs="unbounded">
              <xs:element name="Certificate" maxOccurs="unbounded">
                <xs:simpleType>
                  <xs:restriction base="xs:base64Binary">
                    <xs:minLength value="4"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
            </xs:choice>
          </xs:complexType>
        </xs:element>
        <xs:element name="CheckCrl" minOccurs="0">
          <xs:simpleType>
            <xs:restriction base="xs:integer">
              <xs:minInclusive value="0"/>
              <xs:maxInclusive value="1"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
      </xs:all>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

6.44   ValidateCert Response Schema

This section contains the contents of the ValidateCertResponse.xsd file.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="ValidateCert"
    targetNamespace="ValidateCert" elementFormDefault="qualified"
    attributeFormDefault="unqualified">
  <xs:element name="ValidateCert">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Status" type="xs:unsignedByte"/>
        <xs:element name="Certificate" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Status" type="xs:unsignedByte" minOccurs="0"
                  maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Release: February 10, 2014

7 Appendix B: 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 2007 Service Pack 1 (SP1)

- Microsoft Exchange Server 2010

- Microsoft Exchange Server 2013

- Windows 8.1

- Windows Communication Apps

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

<1> Section 2.2.2.1: When sending an **Autodiscover** command request to Exchange 2007, the Content-Type header accepts the following values: "text/html" or "text/xml".

<2> Section 2.2.2.6: The **GetAttachment** command is not supported when the MS-ASProtocolVersion header is set to 14.0 or 14.1 in the **GetAttachment** command request. Use the **Fetch** element of the **ItemOperations** command instead. For more information about the MS-ASProtocolVersion header, see [MS-ASHTTP] section 2.2.1.1.2.4.

<3> Section 2.2.2.9: In Exchange 2007, the **MeetingResponse** command is used to accept, tentatively accept, or decline a meeting request in the user's Inbox folder only. The Calendar folder cannot be used to modify meeting requests in Exchange 2007.

<4> Section 2.2.2.13: Retrieval of free/busy information using the **Availability** element in the **ResolveRecipients** command is not supported when the MS-ASProtocolVersion header is set to 12.1.

<5> Section 2.2.2.16: Sending the **DeviceInformation** parameters immediately after the client has been provisioned, and before the **FolderSync** command is not recommended for Exchange 2007.

<6> Section 2.2.3.2: The **Account** element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<7> Section 2.2.3.3.1: The **AccountId** element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0. Exchange 2007 returns **Status** value 103 if the **AccountId** element is included in a **SendMail** command request, a **SmartForward** command request, or a **SmartReply** command request.

<8> Section 2.2.3.3.2: The **AccountName** element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<9> Section 2.2.3.4: The **AccountName** element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<10> Section 2.2.3.5: The **Accounts** element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<11> Section 2.2.3.16: The **Availability** element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<12> Section 2.2.3.27.1: The **airsync:Class** element is not supported as a child element of the **airsync:Options** element when the MS-ASProtocolVersion header is set to 12.1. If the MS-ASProtocolVersion is set to 12.1, the **airsync:Class** element is a child element of the **Collection** element.

<13> Section 2.2.3.27.3: The value "Notes" is not supported when the MS-ASProtocolVersion header is set to 12.1.

<14> Section 2.2.3.27.4: The following classes are supported for mailbox searches when the MS-ASProtocolVersion header is set to 12.1: Email, Calendar, Contacts, Tasks. The SMS and Notes classes are only available if the MS-ASProtocolVersion header is set to 14.0 or 14.1.

<15> Section 2.2.3.27.5: The **Delete** and **Class** elements are not returned in the **Sync** response for an SMS deletion when the MS-ASProtocolVersion header is set to 14.0.

<16> Section 2.2.3.27.5: The **Class** element is not supported as a child element of the **Add** element when the MS-ASProtocolVersion header is set to 12.1. When the MS-ASProtocolVersion is set to 12.1, the **Class** element is a child element of the **Collection** element.

<17> Section 2.2.3.27.5: The **Class** element is not supported as a child element of the **Options** element when the MS-ASProtocolVersion header is set to 12.1. When the MS-ASProtocolVersion is set to 12.1, the **Class** element is a child element of the **Collection** element.

<18> Section 2.2.3.27.5: The **Class** element is not supported in a **Sync** command response when the MS-ASProtocolVersion header value is set to 12.1.

<19> Section 2.2.3.27.5: The **Class** element is not supported in a **Sync** command response when the MS-ASProtocolVersion header value is set to 12.1.

<20> Section 2.2.3.27.5: The **Class** element is not supported in a **Sync** command response when the MS-ASProtocolVersion header value is set to 12.1.

<21> Section 2.2.3.29.1: The **FilterType** element is a supported child element of the **Collection** element in requests when the MS-ASProtocolVersion header is set to 12.1.

<22> Section 2.2.3.35.1: The **ConversationId** element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<23> Section 2.2.3.35.2: The **ConversationId** element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<24> Section 2.2.3.36.1: The **airsync:ConversationMode** element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<25> Section 2.2.3.36.2: The **ConversationMode** element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<26> Section 2.2.3.38: In Exchange 2007, the **Culture** element always returns "en:en", regardless of the culture that is sent by the client.

Release: February 10, 2014

<27> Section 2.2.3.39.2: The **Data** element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<28> Section 2.2.3.39.3: The **gal:Data** element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<29> Section 2.2.3.42.2: The **Delete** and **Class** elements are not returned in the **Sync** response for an SMS deletion when the MS-ASProtocolVersion header is set to 14.0.

<30> Section 2.2.3.49.1: The **DstFldId** element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<31> Section 2.2.3.49.2: The **DstFldId** element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<32> Section 2.2.3.54: The **Account** element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<33> Section 2.2.3.54: The **EmailAddresses** element is only supported as a child element of the **Get** element when the MS-ASProtocolVersion header is set to 12.1 or 14.0.

<34> Section 2.2.3.54: The **PrimarySmtpAddress** element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<35> Section 2.2.3.57: The **EnableOutboundSMS** element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<36> Section 2.2.3.58.1: The **EndTime** element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<37> Section 2.2.3.58.1: Exchange 2010 and Exchange 2013 use 42 days; Exchange 2007 SP1 uses 62 days.

<38> Section 2.2.3.64.1: The **Collection** element is a supported parent element of the **FilterType** element when the MS-ASProtocolVersion header is set to 12.1.

<39> Section 2.2.3.75: The **EmailAddresses** element is only supported as a child element of the **Get** element when the MS-ASProtocolVersion header is set to 12.1 or 14.0.

<40> Section 2.2.3.83.1: The **InstanceId** element is not supported when the MS-ASProtocolVersion header is set to 12.1 or 14.0. A **Status** element value of 2 is returned if the **InstanceId** element is included in requests in which the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<41> Section 2.2.3.93.1: The **airsync:MaxItems** element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<42> Section 2.2.3.93.2: The **MaxItems** element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<43> Section 2.2.3.94: The **MaxPictures** element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<44> Section 2.2.3.94.1: The **MaxPictures** element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<45> Section 2.2.3.94.2: The **MaxPictures** element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<46> Section 2.2.3.95: The **MaxSize** element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<47> Section 2.2.3.95.1: The **MaxSize** element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<48> Section 2.2.3.95.2: The **MaxSize** element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<49> Section 2.2.3.97: The **MergedFreeBusy** element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<50> Section 2.2.3.103: The **MobileOperator** element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<51> Section 2.2.3.106: In Exchange 2007, the server sends **Sync** response messages containing the **MoreAvailable** element and between zero (0) and **WindowSize** schema changes when it encounters elements external to the protocol.

<52> Section 2.2.3.107.1: The **Move** element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<53> Section 2.2.3.108: The **MoveAlways** element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<54> Section 2.2.3.113: Exchange 2007, Exchange 2010, and Exchange 2013 require that the reply message for unknown external and known external audiences be the same.

<55> Section 2.2.3.115.1: The **airsync:Options** element is not supported in a **GetItemEstimate** request when the MS-ASProtocolVersion header is set to 12.1.

<56> Section 2.2.3.115.2: The **rm:RightsManagementSupport** element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<57> Section 2.2.3.115.3: The **Picture** element is not supported as a child element when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<58> Section 2.2.3.115.4: The **rm:RightsManagementSupport** element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<59> Section 2.2.3.115.4: The **rm:RightsManagementSupport** element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<60> Section 2.2.3.115.5: The **rm:RightsManagementSupport** element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<61> Section 2.2.3.117: Windows Communication Apps use the value "WINDOWS" for the **Os** element.

<62> Section 2.2.3.124: When the **EnableOutboundSMS** element is set to 1 and the MS-ASProtocolVersion header is set to 14.0, the **PhoneNumber** element is required to have a value. Under these conditions, if the **PhoneNumber** element does not have a value, a **Status** element value of 5 is returned by the server.

<63> Section 2.2.3.125: The **Picture** element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<64> Section 2.2.3.125.1: The **Picture** element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<65> Section 2.2.3.125.2: The **Picture** element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<66> Section 2.2.3.125.2: The **gal:Picture** element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<67> Section 2.2.3.127: The **PrimarySmtpAddress** element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<68> Section 2.2.3.143: The **RightsManagementInformation** element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<69> Section 2.2.3.147: The **SendDisabled** element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<70> Section 2.2.3.152: Sending the **DeviceInformation** parameters immediately after the client has been provisioned, and before the **FolderSync** command is not recommended for Exchange 2007.

<71> Section 2.2.3.161.1: The **StartTime** element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<72> Section 2.2.3.162.7: Exchange 2007, Exchange 2010 SP1, and Exchange 2013 do not return a value of 155 in the **Status** element.

<73> Section 2.2.3.162.11: The **Availability** element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<74> Section 2.2.3.162.11: The **Picture** element is not supported when the MS-ASProtocolVersion header is set to 12.1 or 14.0.

<75> Section 2.2.3.162.12: The **gal:Picture** element is not supported when the MS-ASProtocolVersion header is set to 12.1 or 14.0.

<76> Section 2.2.3.164: The **ResponseRequested** element is not supported as a child element of the **Supported** element when the MS-ASProtocolVersion header value is set to 12.1.

<77> Section 2.2.3.164: The **DisallowNewTimeProposal** element is not supported as a child of the **Supported** element when the MS-ASProtocolVersion header value is set to 12.1.

<78> Section 2.2.3.166.3: When the MS-ASProtocolVersion header value is set to 12.1, the **airsync:SyncKey** element is placed after the **FilterType** element in a **GetItemEstimate** command request.

<79> Section 2.2.3.168: Some fields that are indexed by using ANR in **Active Directory** by default in Exchange 2007 are as follows: Name, Alias, Email, Office. The ANR property set that can be indexed is definable by the administrator and can be extended to include other fields.

<80> Section 2.2.3.170.3: The value 19 for the **Type** element in **FolderSync** command response is not supported when the MS-ASProtocolVersion header, as described in [MS-ASHTTP] section 2.2.1.1.2.4, is set to 12.1.

<81> Section 2.2.3.175: The **UserDisplayName** element is not supported when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<82> Section 2.2.4: In Exchange 2007, this was an HTTP 400 response.

<83> Section 2.2.4: In Exchange 2007, this was an HTTP 400 response, or 500 for the **SendMail** command.

<84> Section 2.2.4: In Exchange 2007, this was an HTTP 500 response.

<85> Section 2.2.4: In Exchange 2007, this was an HTTP 503 response.

<86> Section 2.2.4: In Exchange 2007, this was an HTTP 403 response.

<87> Section 2.2.4: In Exchange 2007, this was an HTTP 507 response.

<88> Section 2.2.4: In Exchange 2007, this was an HTTP 500 response, or 403 for the **Provision** command.

<89> Section 2.2.4: In Exchange 2007, this was an HTTP 501 response.

<90> Section 2.2.4: In Exchange 2007, this was an HTTP 400 response, or 505 for version 1.0 devices.

<91> Section 2.2.4: In Exchange 2007, this was an HTTP 449 response, or 403 if there was no policy key header.

<92> Section 2.2.4: In Exchange 2007, this was an HTTP 449 response.

<93> Section 2.2.4: In Exchange 2007, this was an HTTP 400 or 501 response.

<94> Section 2.2.4: **Status** value 160 is not returned when the MS-ASProtocolVersion header value is set to 12.1.

<95> Section 2.2.4: **Status** value 161 is not returned when the MS-ASProtocolVersion header value is set to 12.1.

<96> Section 2.2.4: **Status** value 162 is not returned when the MS-ASProtocolVersion header value is set to 12.1.

<97> Section 2.2.4: **Status** value 163 is not returned when the MS-ASProtocolVersion header value is set to 12.1.

<98> Section 2.2.4: **Status** value 164 is not returned when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<99> Section 2.2.4: **Status** value 165 is not returned when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<100> Section 2.2.4: **Status** value 166 is not returned when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<101> Section 2.2.4: **Status** value 167 is not returned when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<102> Section 2.2.4: **Status** value 168 is not returned when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<103> Section 2.2.4: **Status** value 169 is not returned when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<104> Section 2.2.4: **Status** value 170 is not returned when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<105> Section 2.2.4: **Status** value 171 is not returned when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<106> Section 2.2.4: **Status** value 172 is not returned when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<107> Section 2.2.4: **Status** value 173 is not returned when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<108> Section 2.2.4: **Status** value 174 is not returned when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<109> Section 2.2.4: **Status** value 175 is not returned when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<110> Section 2.2.4: **Status** value 176 is not returned when the MS-ASProtocolVersion header value is set to 12.1 or 14.0.

<111> Section 2.2.4: **Status** value 177 is not returned when the MS-ASProtocolVersion header value is set to 12.1. In addition, Exchange 2007, Exchange 2010 SP2, and Exchange 2013 do not return status value 177 when the MS-ASProtocolVersion header value is set to 14.0.

<112> Section 3.1.5.1: Windows Communication Apps do not issue a **Provision** command as the first command.

<113> Section 3.1.5.2: Windows Communication Apps send the **settings:DeviceInformation** element after the **FolderSync** command.

<114> Section 3.1.5.2: Sending the **settings:DeviceInformation** parameters immediately after the client has been provisioned and before the **FolderSync** command is not recommended for Exchange 2007.

<115> Section 3.1.5.3: Windows Communication Apps do not restart the synchronization process with a synchronization key of 0 when the **FolderSync** response contains a **Status** element value of 9.

<116> Section 3.1.5.5: Exchange 2007 SP1 and Exchange 2010 do not return substitute meeting invitation email messages.

<117> Section 3.1.5.5: Exchange 2007 SP1 and Exchange 2010 return meeting requests with the **email2:MeetingMessageType** element ([MS-ASEMAIL] section 2.2.2.39) set to 6.

<118> Section 3.1.5.8: Exchange 2007 SP1 and Exchange 2010 do not limit the number of elements in command requests. Update Rollup 6 for Exchange 2010 SP2 and Exchange 2013 do limit the number of elements in command requests.

<119> Section 3.1.5.8: Exchange 2007 SP1 and Exchange 2010 do not limit the number of elements in command requests. Update Rollup 6 for Exchange 2010 SP2 and Exchange 2013 use the specified limit values by default but can be configured to use different values.

Release: February 10, 2014

8 Change Tracking

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

358 / 360

9 Index

Security

T

V