[MS-ASCMD]:
ActiveSync Command Reference Protocol Specification

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft's Open Specification Promise (available here: http://www.microsoft.com/interop/osp) or the Community Promise (available here: http://www.microsoft.com/interop/cp/default.mspx). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.

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

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

Copyright © 2008 Microsoft Corporation.

Revision Summary

Date	Revision History	Revision Class	Comments
12/03/2008	1.0		Initial Release.
01/15/2009	1.01		Revised and edited technical content.
03/04/2009	1.02		Revised and edited technical content.
04/10/2009	2.0		Updated technical content and applicable product releases.
07/15/2009	3.0	Major	Revised and edited for technical content.
11/04/2009	4.0.0	Major	Updated and revised the technical content.

Copyright © 2008 Microsoft Corporation.

Table of Contents

Copyright © 2008 Microsoft Corporation.

Copyright © 2008 Microsoft Corporation.

Copyright © 2008 Microsoft Corporation.

Copyright © 2008 Microsoft Corporation.

Copyright © 2008 Microsoft Corporation.

Copyright © 2008 Microsoft Corporation.

Copyright © 2008 Microsoft Corporation.

Copyright © 2008 Microsoft Corporation.

Copyright © 2008 Microsoft Corporation.

Copyright © 2008 Microsoft Corporation.

Copyright © 2008 Microsoft Corporation.

Copyright © 2008 Microsoft Corporation.

1 Introduction

This document specifies the ActiveSync protocol commands which are used by a client, typically a mobile device, to synchronize and exchange objects with a server. These objects include e-mail **messages**, SMS messages, **attachments**, **folders**, **contact** information, **meetings**, calendar data, tasks, notes and documents.

1.1 Glossary

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

Active Directory
address book
address list
alias
ambiguous name resolution (ANR)
appointment
ASCII
attachment
Autodiscover server
binary large object (BLOB)
body part
Calendar folder
Calendar object
character set
class
collection
contact
conversation
distribution list
domain
Domain Name System (DNS)
Drafts folder
folder
folder ID (FID)
ghosted
Global Address List (GAL)
GUID
Hypertext Transfer Protocol (HTTP)
Inbox folder
journal
locale
mailbox
meeting
message
message body
message database (MDB)
message ID (MID)
MIME
MIME message
named property
organizer
Out of Office (OOF)
Outbox folder

Copyright © 2008 Microsoft Corporation.

Personal Information Manager (PIM)
plain text
property (1)
read receipt
recipient (2)
recipient information cache
reminder
Root folder
S/MIME
search folder
Secure Sockets Layer (SSL)
Sent Items folder
Short Message Service (SMS)
Simple Mail Transfer Protocol (SMTP)
special folder
stream
store
synchronization
Uniform Resource Locator (URL)
Uniform Resource Identifier (URI)
WAP Binary XML (WBXML)
Wireless Application Protocol (WAP)
XML
XML schema definition (XSD)

The following terms are specific to this document:

certificate authority (CA): A third party that issues public key certificates. Certificates serve to bind public keys to a user identity. Each user and **certificate authority** may decide whether to trust another user or **CA** for a specific purpose, and whether this trust should be transitive.

certificate revocation lists (CRL): A list of certificates that have been revoked by the **certificate authority (CA)** (or certification authority) that issued them (that have not yet expired of their own accord). The list must be cryptographically signed by the **CA** that issues it. Typically, the certificates are identified by serial number. In addition to the serial number for the revoked certificates, the **CRL** also contains the revocation reason for each certificate and the time the certificate was revoked. As specified in [RFC3280], two types of **CRLs** commonly exist in the industry. Base **CRLs** keep a complete list of revoked certificates, while delta **CRLs** maintain only those certificates that have been revoked since the last issuance of a base **CRL**. For more information, see section 7.3 of [X509], [MSFT-CRL], and section 5 of [RFC3280].

fully qualified domain name (FQDN): In a domain naming system, an unambiguous domain name that specifies the node's position in the domain naming service tree hierarchy absolutely.

Universal Naming Convention (UNC): A standard naming format for specifying the location of network resources such as shared files or devices on a network. The format is "\\<servername>\<share>\<filename>", where <servername> is a NetBIOS name, **fully qualified domain name (FQDN)**, or IPv4 address; <share> is a logical share point for accessing <servername>; and <filename> is the name of the file or device.

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.

Copyright © 2008 Microsoft Corporation.

1.2 References

1.2.1 Normative References

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

[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", August 2007, http://go.microsoft.com/fwlink/?linkid=3052&kbid=940881.

[MS-ASAIRS] Microsoft Corporation, "ActiveSync AirSyncBase Namespace Protocol Specification", December 2008.

[MS-ASCAL] Microsoft Corporation, "ActiveSync Calendar Class Protocol Specification", December 2008.

[MS-ASCNTC] Microsoft Corporation, "ActiveSync Contact Class Protocol Specification", December 2008.

[MS-ASCON] Microsoft Corporation, "ActiveSync Conversations Protocol Specification", June 2008.

[MS-ASDOC] Microsoft Corporation, "ActiveSync Document Class Protocol Specification", December 2008.

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

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

[MS-ASHTTP] Microsoft Corporation, "ActiveSync HTTP Protocol Specification", December 2008.

[MS-ASMS] Microsoft Corporation, "ActiveSync Short Message Service Protocol Specification", June 2008.

[MS-ASPROV] Microsoft Corporation, "ActiveSync Provisioning Protocol Specification", December 2008.

[MS-ASTASK] Microsoft Corporation, "ActiveSync Tasks Class Protocol Specification", December 2008.

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

[MS-OXDSCLI] Microsoft Corporation, "Autodiscover Publishing and Lookup Protocol Specification", June 2008.

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

[MSFT-CRL] Microsoft Corporation, "Certificate Revocation and Status Checking", January 2006, http://go.microsoft.com/fwlink/?linkid=90181.

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

Copyright © 2008 Microsoft Corporation.

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

[RFC821] Postel, J., "Simple Mail Transfer Protocol", RFC 821, August 1982, http://www.ietf.org/rfc/rfc821.txt.

[RFC822] Crocker, D.H., "Standard for ARPA Internet Text Messages", RFC 822, August 1982, http://www.ietf.org/rfc/rfc0822.txt.

[RFC3280] Housley, R., Polk, W., Ford, W., and Solo, D., "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, April 2002, http://www.ietf.org/rfc/rfc3280.txt.

[W3C-XML] World Wide Web Consortium, "XML Schema (Second Edition)", October 2004, http://www.w3.org/XML/Schema.

[X509] ITU-T, "Information Technology - Open Systems Interconnection - The Directory: Public-Key and Attribute Certificate Frameworks", Recommendation X.509, August 2005, http://www.itu.int/rec/T-REC-X.509/en.

1.2.2 Informative References

[AUTODISCOVER] Microsoft Corporation, "White Paper: Exchange 2007 Autodiscover Service", November 2007, http://technet.microsoft.com/en-us/library/bb332063.aspx.

1.3 Protocol Overview

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

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

The client then uses the **Provision** command to get and subsequently acknowledge security policy settings from the server.

The next command sent by the client is **FolderSync** to retrieve the folder hierarchy of the user.

This is typically followed by **GetItemEstimate** in order to retrieve the number of changes that need to be downloaded to the client via the first **Sync** request. This is immediately followed by **Sync**, to get a **synchronization** key and then messages from the server. Optionally, **Ping** or hanging **Sync** can then be issued to keep the device up-to-date on any server changes.

The client processes outgoing e-mail using the **SendMail**, **SmartReply**, and **SmartForward** commands. For incoming messages, the client can call the **ItemOperations** command to fetch the message, and then use the **MoveItems** command. **S/MIME** messages are processed with the **ResolveRecipients** and **ValidateCert** commands.

The client calls the **FolderSync**, **FolderCreate**, **FolderUpdate**, and **FolderDelete** commands to update, create, and delete **mailbox** folders on the server.

For meeting requests, the client calls the **MeetingResponse** command.

The client can set and request server parameters with the **Settings** command.

The client uses the **Search** command to find particular items on the server.

Copyright © 2008 Microsoft Corporation.

1.4 Relationship to Other Protocols

The ActiveSync commands specified in this document are sent and received over a **Hypertext Transfer Protocol (HTTP)** connection, as specified in [RFC2616] in an HTTP **POST** method. The information contained in the HTTP **POST** header is specified in [MS-ASHTTP]. The information contained in the HTTP message is sent and received in WBXML format, as specified in [MS-ASWBXML] where the content of the WBXML adheres to the commands specified in this document.

Some of the ActiveSync commands specified in this document are used to synchronize or retrieve more than one class of content. For example, the **Ping** command can be used to monitor changes to the e-mail, note, contact, calendar, or task classes. The elements included in the **Ping** command change depending on which content class is being monitored. Because each of the content classes are used by multiple commands, they are specified in individual documents for each content type. The content class specifications are [MS-ASEMAIL], [MS-ASCNTC], [MS-ASDOC], [MS-ASCAL], [MS-ASNOTE], and [MS-ASTASK].

Another document containing elements and complex types used by multiple commands is [MS-ASAIRS]. [MS-ASAIRS] specifies 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 specifies all of the ActiveSync commands except for the **Provision** command, which is independently specified in [MS-ASPROV].

The **AutoDiscover** command is specified in this document, but more details about AutoDiscover publishing and lookup are available in [MS-OXDSCLI].

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

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

For details about how outbound SMS e-mail messages are sent from mobile devices, see **[MS-ASMS]**.

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.

Copyright © 2008 Microsoft Corporation.

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 **WAP Binary XML (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 Commands

2.2.1.1 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.<1>

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

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 **Options** command to the new server. The **Options** command returns the newly supported protocol versions and commands if they changed due to the **Autodiscover** command.

2.2.1.1.1 Request

2.2.1.1.1.1 Request

The <Request> element contains the **Autodiscover** command request parameters.

Parent elements	Child elements	Data type	Number allowed
<Autodiscover>	<EmailAddress> <AcceptableResponseSchema>	**Container**	1…1 (required)

2.2.1.1.1.2 AcceptableResponseSchema

The <AcceptableResponseSchema> element indicates the schema in which the server MUST send the response.

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
<Request> (request only) <Response> (response only)	None	**String**	1...1 (required)

The schema MUST be "HTTP://schemas.microsoft.com/exchange/autodiscover/mobilesync/responseschema/2006".

2.2.1.1.1.3 EmailAddress

The <EmailAddress> element contains the SMTP e-mail address of the user and is used to identify the user's mailbox in the network.

Parent elements	Child elements	Data type	Number allowed
<Request> (request only) <User> (response only)	None	**String**	1...1 (required)

If the user has multiple addresses, then the primary e-mail address SHOULD be returned in the **Autodiscover** command response. This address can be the same as the e-mail address that was sent in the request. The client device records this address string for use in all additional communication.

2.2.1.1.2 Response

2.2.1.1.2.1 Action

The <Action> element encapsulates the server action type for this request, which can be one of the following: <Redirect>, <Settings>, or <Error>.

Parent elements	Child elements	Data type	Number allowed
None	<Redirect> (response only) <Settings> (response only) <Error> (response only)	**Container**	1...1 (required)

2.2.1.1.2.2 Culture

The <Culture> element specifies the client culture, which is used to localize error messages.

Parent elements	Child elements	Data type	Number allowed
<Response>	None	**String**	0...1 (optional)

The string MUST be of the form "en:us".<2>

2.2.1.1.2.3 DebugData

The <DebugData> element represents more information about the failure that can help the system administrator debug the source of the problem.

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
<Error>	None	**String**	0...1 (optional)

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

2.2.1.1.2.4 DisplayName

The <DisplayName> element contains the user's display name in the directory service.

Parent elements	Child elements	Data type	Number allowed
<User>	None	**String**	0...1 (optional)

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

2.2.1.1.2.5 EmailAddress

The <EmailAddress> element contains the SMTP e-mail address of the user and is used to identify the user's mailbox in the network.

Parent elements	Child elements	Data type	Number allowed
<Request> (request only) <User> (response only)	None	**String**	1...1 (required)

If the user has multiple addresses, then the primary e-mail address is returned in the **Autodiscover** command response. This address can be the same as the e-mail address that was sent in the request. The client device SHOULD record this address and SHOULD use this string for all additional communication.

2.2.1.1.2.6 Error

The <Error> element contains the error that was encountered while processing the request.

Parent elements	Child elements	Data type	Number allowed
<Action> (response only) <Response> (response only)	<Status> <Message> <DebugData> <ErrorCode>	**Container**	0...1 (optional)

2.2.1.1.2.7 Message

The <Message> element contains the error string localized using the <Culture> specified in the <Response> element, enabling the client to display error status to the end-user.

Parent elements	Child elements	Data type	Number allowed
Error	None	**String**	0...1 (optional)

Copyright © 2008 Microsoft Corporation.

2.2.1.1.2.8 Name

The <Name> element specifies a URL if the **Type** element is set to *MobileSync*.

Parent elements	Child elements	Data type	Number allowed
<Server> (response only)	None	**String**	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> value is NULL.

2.2.1.1.2.9 Redirect

The <Redirect> element specifies the SMTP address of the requested user.

Parent elements	Child elements	Data type	Number allowed
<Action>	None	**String**	0...1 (optional)

The **Redirect** element is an optional child of the <Action> element in the **Autodiscover** response message. The client device uses the **domain** part of the address to send a new **Autodiscover** command request.

2.2.1.1.2.10 Response

The <Response> element contains the **Autodiscover** command response parameters.

Parent elements	Child elements	Data type	Number allowed
<Autodiscover>	<User> <Culture> <Action> <Error>	**Container**	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 node.

2.2.1.1.2.11 Server

The <Server> element encapsulates settings that apply to a particular server in the **Autodiscover** command response.

Parent elements	Child elements	Data type	Number allowed
<Settings> (response only)	<Type> <URL> <Name> <ServerData>	**Container**	1...N (required)

2.2.1.1.2.12 ServerData

The <ServerData> element contains the template name for the client certificate. <3>

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
<Server> (response only)	None	**String**	0...1 (optional)

The <ServerData> element is a string that is present only when the <Type> element value is set to **CertEnroll**.

2.2.1.1.2.13 Settings

The <Settings> element contains the settings for the specified user or schema.

Parent elements	Child elements	Data type	Number allowed
<Action> (response only)	Settings that are specific to the requested service	**Container**	0...1 (optional)

2.2.1.1.2.14 Status

The <Status> element provides a status code that corresponds to the error.

Parent elements	Child elements	Data type	Number allowed
<Error>	None	**Integer**	0...1 (optional)

The following table specifies valid values for the <Status> element in the context of the <Settings> element.

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.

2.2.1.1.2.15 Type

The <Type> element specifies the server type.

Parent elements	Child elements	Data type	Number allowed
<Server> (response only)	None	**String**	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 can be accessed by clients. <4>
- *CertEnroll*. Indicates that the URL that is returned by the URL element can be accessed by clients that have a valid certificate over a **Secure Sockets Layer (SSL)**. <5>

Copyright © 2008 Microsoft Corporation.

If the server supports both *MobileSync* and **CertEnroll**, the response buffer includes multiple <Server> elements that contain a URL value for each <Type> value.

2.2.1.1.2.16 Url

The <URL> element contains a URL string that conveys the protocol, port, resource location, and other information.

Parent elements	Child elements	Data type	Number allowed
<Server> (response only)	None	**String**	0...1 (optional)

The URL element is a child of the <Server> element. The value is a URL string that conveys the protocol, port, resource location, and other information. <6>

2.2.1.1.2.17 User

The <User> element encapsulates information about the user to whom this response element relates.

Parent elements	Child elements	Data type	Number allowed
<Response>	<DisplayName> <EmailAddress>	**Container**	1...1 (required)

2.2.1.1.2.18 ErrorCode

The <ErrorCode> element contains an error number that indicates the cause of the error.

Parent elements	Child elements	Data type	Number allowed
Error	None	**Integer**	0...1 (optional)

If the provider cannot be found, or if the <AcceptableResponseSchema> cannot be matched, then the ErrorCode 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> that was specified.

2.2.1.2 FolderCreate

The **FolderCreate** command creates a new folder as a child of the specified parent folder. A parent ID of 0 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.

2.2.1.2.1 Request

The server that is implementing [MS-ASCMD] enforces the following **XML schema definition (XSD)** when processing protocol requests.

Requests that do not adhere to the schema result in the return of a status 10 to the client.

Copyright © 2008 Microsoft Corporation.

```
<?xml version="1.0" ?>
<xs:schema xmlns:tns="FolderHierarchy:" attributeFormDefault="unqualified"
elementFormDefault="qualified" targetNamespace="FolderHierarchy:"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="FolderCreate">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="SyncKey">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:maxLength value="64"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="ParentId">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:maxLength value="64"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <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>
```

2.2.1.2.1.1 FolderCreate

The <FolderCreate> element is the top-level element in the XML document. It identifies the body of the HTTP Post as containing a **FolderCreate** command.

Parent elements	Child elements	Data type	Number allowed
None	<SyncKey> <ParentId> (request only) <DisplayName> (request only) <Type> (request only) <ServerId> (response only) <Status> (response only)	**Container**	1 (required)

2.2.1.2.1.2 SyncKey

The <SyncKey> element specified in the **FolderCreate** command request represents the synchronization state of a **collection**. After a successful **FolderCreate** command, the server sends a synchronization key to the client in a response. 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 checks the

Copyright © 2008 Microsoft Corporation.

value of the key to make sure the value of the <SyncKey> provided in the request matches a <SyncKey> value on the server. The server MUST provide a <Status> value of 9 if the <SyncKey> values do not match.

Parent elements	Child elements	Data type	Number allowed
<FolderCreate>	None	**String** (Up to 64 characters)	Request: 1 (required) Response: 0...1

The client MUST store the synchronization key as an opaque string of up to 64 characters.

The <SyncKey> element is returned if the **FolderCreate** command request was successful and the element is not returned if the **FolderCreate** command request fails.

2.2.1.2.1.3 ParentId

The <ParentId> element specifies the server ID of the parent folder and is used in **FolderCreate** command requests only. The server ID of the parent folder is obtained from the <ServerId> element of a previous **FolderSync** command. A parent ID of 0 signifies the mailbox root folder.

Parent elements	Child elements	Data type	Number allowed
<FolderCreate> (request only)	None	**String** (Up to 64 characters)	1 (required)

2.2.1.2.1.4 DisplayName

The <DisplayName> element specifies the name of the folder that is shown to the user.

Parent elements	Child elements	Data type	Number allowed
<FolderCreate> (request only)	None	**String** (Between 1 and 256 characters)	1 (required)

2.2.1.2.1.5 Type

The <Type> element specifies the type of the folder to be created.

Parent elements	Child elements	Data type	Number allowed
<FolderCreate> (request only)	None	**Integer**	1 (required)

The folder type values are listed in the following table. Folder types 2–11 and 19 are reserved for default folder types.

Type	Definition
1	User-created folder (generic)
12	User-created mail folder
13	User-created **Calendar folder**
14	User-created Contacts folder

Copyright © 2008 Microsoft Corporation.

Type	Definition
15	User-created Tasks folder
16	User-created Journal folder
17	User-created Notes folder

2.2.1.2.2 Response

The following code shows the XSD for the **FolderCreate** command response.

```
<?xml version="1.0" ?>
<xs:schema xmlns:tns="FolderHierarchy:" attributeFormDefault="unqualified"
elementFormDefault="qualified" targetNamespace="FolderHierarchy:"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="FolderCreate">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Status" type="xs:unsignedByte" />
                <xs:element minOccurs="0" name="SyncKey" type="xs:string" />
                <xs:element minOccurs="0" name="ServerId" type="xs:string" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

2.2.1.2.2.1 FolderCreate

The <FolderCreate> element is the top-level element in the XML document. It identifies the body of the HTTP **POST** response as containing a **FolderCreate** command.

Parent elements	Child elements	Data type	Number allowed
None	<SyncKey> <ParentId> (request only) <DisplayName> (request only) <Type> (request only) <ServerId> (response only) <Status> (response only)	**Container**	1 (required)

2.2.1.2.2.2 ServerId

The <ServerId> element 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. The <ServerId> can also be used in the <ServerId> element of future **FolderDelete** and **FolderUpdate** command requests. The client MUST store the <ServerId> for each folder and MUST be able to locate a folder on the client given a <ServerId>.

Parent elements	Child elements	Data type	Number allowed
<FolderCreate> (response only)	None	**String** (Up to 64 characters)	0...1 (optional)

Copyright © 2008 Microsoft Corporation.

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.

2.2.1.2.2.3 Status

The <Status> element indicates in the **FolderCreate** command response the success or failure of a **FolderCreate** command request. If the command failed, the <Status> element contains a code indicating the type of failure. The values are summarized in the following table.

Parent elements	Child elements	Data type	Number allowed
<FolderCreate> (response only)	None	**Integer**	1 (required)

The following table shows valid values for the element.

Value	Meaning
1	Success.
2	A folder with that name already exists.
3	The specified folder is a special system folder, like the **Inbox**, **Outbox**, Contacts, Recipient information, or **Drafts folders**, and cannot be created by the client.
5	The specified parent folder was not found.
6	An error on the server.
8	The request timed out.
9	Synchronization key mismatch or invalid synchronization key.
10	Incorrectly formatted request.
11	An unknown error occurred.

2.2.1.2.2.4 SyncKey

The <SyncKey> element specified in the **FolderCreate** command response represents the synchronization state of a collection.

Parent elements	Child elements	Data type	Number allowed
<FolderCreate>	None	**String** (Up to 64 characters)	Request: 1 (required) Response: 0...1

After a successful **FolderCreate** command, the server MUST send a synchronization key to the client in a response. If the **FolderCreate** 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> provided in the request matches a <SyncKey> value on the server. The server MUST provide a Status value of 9 if the <SyncKey> values do not match.

The client MUST store the synchronization key as an opaque string of up to 64 characters.

Copyright © 2008 Microsoft Corporation.

2.2.1.3 FolderDelete

The **FolderDelete** command deletes a folder from the server. The ServerId of the folder is passed to the server in the **FolderDelete** command request, 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> value of 3.

2.2.1.3.1 Request

The following code shows the XSD for the **FolderDelete** command request.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
        xmlns:tns="FolderHierarchy:"
        attributeFormDefault="unqualified"
        elementFormDefault="qualified"
        targetNamespace="FolderHierarchy:"
        xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:element name="FolderDelete">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="SyncKey">
                    <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:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

2.2.1.3.1.1 FolderDelete

The <FolderDelete> element is the top-level element in the XML document. It identifies the body of the HTTP Post as containing a **FolderDelete** command.

Parent elements	Child elements	Data type	Number allowed
None	<SyncKey> <ServerId> (request only) <Status> (response only)	**Container**	1 (required)

Copyright © 2008 Microsoft Corporation.

2.2.1.3.1.2 SyncKey

The <SyncKey> element represents the synchronization state of a folder hierarchy.

Parent elements	Child elements	Data type	Number allowed
<FolderDelete>	None	**String** (Up to 64 characters)	Request: 1 (required) Response: 0...1

After a successful **FolderDelete** command request, the server MUST send a synchronization key to the client in the response. If the **FolderDelete** command request is unsuccessful, 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> provided in the request matches a <SyncKey> value on the server. The server MUST provide a <Status> value of 9 if the <SyncKey> values do not match.

The client MUST store the synchronization key as an opaque string of up to 64 characters.

2.2.1.3.1.3 ServerId

The <ServerId> element 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.

Parent elements	Child elements	Data type	Number allowed
<FolderDelete> (request only)	None	**String** (Up to 64 characters)	1 (required)

The server ID of the folder to be deleted is returned to the client in the <ServerId> element of a previous **FolderSync** or **FolderCreate** command. The client MUST store the server ID for each object and MUST be able to locate an object given a server ID.

The client MUST store the synchronization key as an opaque string of up to 64 characters.

2.2.1.3.2 Response

The following code shows the XSD for the **FolderDelete** command response.

```
<?xml version="1.0" ?>
<xs:schema xmlns:tns="FolderHierarchy:" attributeFormDefault="unqualified"
elementFormDefault="qualified" targetNamespace="FolderHierarchy:"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="FolderDelete">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Status" type="xs:unsignedByte" />
                <xs:element minOccurs="0" name="SyncKey" type="xs:string" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```


Copyright © 2008 Microsoft Corporation.

2.2.1.3.2.1 FolderDelete

The <FolderDelete> element is the top-level element in the XML document. It identifies the body of the HTTP **POST** response as containing a **FolderDelete** command.

Parent elements	Child elements	Data type	Number allowed
None	<SyncKey> <ServerId> (request only) <Status> (response only)	**Container**	1 (required)

2.2.1.3.2.2 SyncKey

The <SyncKey> element is used by the server to mark the synchronization state of a folder hierarchy.

Parent elements	Child elements	Data type	Number allowed
<FolderDelete>	None	**String** (Up to 64 characters)	Request: 1 (required) Response: 0...1

After a successful **FolderDelete** command, the server MUST send a synchronization key to the client in a response. If the **FolderDelete** 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> provided in the request matches a <SyncKey> value on the server. The server MUST provide a <Status> value of 9if the <SyncKey> values do not match.

2.2.1.3.2.3 Status

The <Status> element indicates the success or failure of the **FolderDelete** command request. If the command failed, the <Status> element in the server response contains a code indicating the type of failure.

Parent elements	Child elements	Data type	Number allowed
<FolderDelete> (response)	None	**Integer**	1 (required)

The following table lists the valid values for this element.

Value	Meaning
1	Success.
3	The specified folder is a special system folder, such as the Inbox, Outbox, Contacts, Recipient information, or Drafts folders, and cannot be deleted by the client.
4	The specified folder does not exist.
6	An error occurred on the server.
8	The request timed out.

Copyright © 2008 Microsoft Corporation.

Value	Meaning
9	Synchronization key mismatch or invalid synchronization key.
10	Incorrectly formatted request.
11	An unknown error occurred.

2.2.1.4 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. An initial **FolderSync** command with a synchronization key of 0 (value of 0 in <SyncKey> element) is required in order to obtain the list of folders and the synchronization key associated with that list. The synchronization key MUST be returned in the <SyncKey> element of the response. This synchronization key MUST be used in subsequent **FolderSync** commands to obtain folder hierarchy changes.

Unlike a **Sync** request, there is no <GetChanges> element submitted in the **FolderSync** request 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.

2.2.1.4.1 Request

The following code shows the XSD for the **FolderSync** command request.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
        xmlns:tns="FolderHierarchy:"
        attributeFormDefault="unqualified"
        elementFormDefault="qualified"
        targetNamespace="FolderHierarchy:"
        xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:element name="FolderSync">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="SyncKey">
                 <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:maxLength value="64"/>
                    </xs:restriction>
                 </xs:simpleType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

2.2.1.4.1.1 FolderSync

The <FolderSync> element is the top-level element in the XML **stream**. It indicates that the body of the HTTP POST contains a **FolderSync** command.

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
None	<SyncKey> <Status> (response only) <Changes> (response only)	**Container**	1 (required)

2.2.1.4.1.2 SyncKey

The <SyncKey> element is used by the server to track the current state of the client.

Parent elements	Child elements	Data type	Number allowed
<FolderSync>	None	**String** (Up to 64 characters)	1 (required)

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> provided in the request matches a <SyncKey> value on the server. The server MUST provide a <Status> value of 9 if the <SyncKey> values do not match.

The client MUST store the synchronization key as an opaque string of up to 64 characters.

If a synchronization error occurs, and the **FolderSync** response has status code 9 (see section 2.2.2.3), then the client MUST restart the synchronization process with a synchronization key of 0. The client data can then be merged with the data returned by the server, or the client data can be completely deleted and replaced with the data from the server.

2.2.1.4.2 Response

The following code shows the XSD for the **FolderSync** command response.

```
<?xml version="1.0" ?>
<xs:schema xmlns:tns="FolderHierarchy:" attributeFormDefault="unqualified"
elementFormDefault="qualified"
targetNamespace="FolderHierarchy:" xmlns:xs="http://www.w3.org/2001/XMLSchema">
     <xs:element name="FolderSync">
          <xs:complexType>
               <xs:sequence>
                    <xs:element minOccurs="1" maxOccurs="1" name="Status"
type="xs:unsignedByte" />
                    <xs:element minOccurs="0" maxOccurs="1" name="SyncKey" type="xs:string"
/>
                    <xs:element minOccurs="0" maxOccurs="1" name="Changes">
                         <xs:complexType>
                              <xs:sequence>
                                   <xs:element name="Count" minOccurs="0" maxOccurs="1"
type="xs:unsignedByte" />
                                   <xs:element minOccurs="0" maxOccurs="unbounded"
name="Update">
                                        <xs:complexType>
                                             <xs:sequence>
                                                  <xs:element minOccurs="1" maxOccurs="1"
name="ServerId" type="xs:string" />
                                                  <xs:element minOccurs="1" maxOccurs="1"
name="ParentId" type="xs:string" />
```


Copyright © 2008 Microsoft Corporation.

```
                                                    <xs:element minOccurs="1" maxOccurs="1"
name="DisplayName" type="xs:string" />
                                                    <xs:element minOccurs="1" maxOccurs="1"
name="Type" type="xs:unsignedByte" />
                                                </xs:sequence>
                                            </xs:complexType>
                                        </xs:element>
                                        <xs:element minOccurs="0" maxOccurs="unbounded"
name="Delete">
                                            <xs:complexType>
                                                <xs:sequence>
                                                    <xs:element minOccurs="1" maxOccurs="1"
name="ServerId" type="xs:string" />
                                                </xs:sequence>
                                            </xs:complexType>
                                        </xs:element>
                                        <xs:element minOccurs="0" maxOccurs="unbounded"
name="Add">
                                            <xs:complexType>
                                                <xs:sequence>
                                                    <xs:element minOccurs="1" maxOccurs="1"
name="ServerId" type="xs:string" />
                                                    <xs:element minOccurs="1" maxOccurs="1"
name="ParentId" type="xs:string" />
                                                    <xs:element minOccurs="1" maxOccurs="1"
name="DisplayName" type="xs:string" />
                                                    <xs:element minOccurs="1" maxOccurs="1"
name="Type" type="xs:unsignedByte" />
                                                </xs:sequence>
                                            </xs:complexType>
                                        </xs:element>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
</xs:schema>
```

2.2.1.4.2.1 FolderSync

The <FolderSync> element is the top-level element in the XML stream. It indicates that the body of the HTTP **POST** response contains a **FolderSync** command.

Parent elements	Child elements	Data type	Number allowed
None	<SyncKey> <Status>(response only) <Changes> (response only)	**Container**	1 (required)

2.2.1.4.2.2 Status

The <Status> element indicates the success or failure of a **FolderSync** command request.

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
<FolderSync> (response only)	None	**Integer** (See values in the following table)	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 valid values for this element.

Value	Meaning
1	Success.
6	An error occurred on the server.
8	The request timed out.
9	Synchronization key mismatch or invalid synchronization key.
10	Incorrectly formatted request.
11	An unknown error occurred.

2.2.1.4.2.3 SyncKey

The <SyncKey> element is used by the server to track the current state of the client.

Parent elements	Child elements	Data type	Number allowed
<FolderSync>	None	**String** (Up to 64 characters)	0…1 (optional)

After a 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> provided in the request matches a <SyncKey> value on the server. The server MUST provide a <Status> value of 9 if the <SyncKey> values do not match.

The client MUST store the synchronization key as an opaque string of up to 64 characters.

If a synchronization error occurs, where the **FolderSync** response has status code 9 (see section 2.2.2.3), the client MUST restart the synchronization process with a synchronization key of 0. The client data can then be merged with the data returned by the server, or the client data can be completely deleted and replaced with the data from the server.

2.2.1.4.2.4 Changes

The <Changes> element is a container for changes to the folder hierarchy. It is used in the **FolderSync** command response to update the client with folder additions, deletions, and updates on the server.

Copyright © 2008 Microsoft Corporation.

The server SHOULD maintain the same set of folder data being returned across synchronization key 0, in terms of <ServerId> and <DisplayName> mapping. However, if an error occurs, the server can return a totally different set.

Parent elements	Child elements	Data type	Number allowed
<FolderSync> (response only)	<Count> <Add> <Delete> <Update>	**Container**	0…1 (optional)

2.2.1.4.2.5 Count

The <Count> element is used in the **FolderSync** command response to list 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> of 0 is returned.

Parent elements	Child elements	Data type	Number allowed
<Changes> (response only)	None	**Unsigned Integer**	0…1 (optional)

2.2.1.4.2.6 Delete

The <Delete> element is used in the **FolderSync** command response to specify that a folder on the server was deleted since the last folder synchronization.

Parent elements	Child elements	Data type	Number allowed
<Changes> (response only)	<ServerId>	**Container**	0...N (optional)

2.2.1.4.2.7 Add

The <Add> element is used in a **FolderSync** command response to create 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.

Parent elements	Child elements	Data type	Number allowed
<Changes> (response only)	<ServerId> <ParentId> <DisplayName> <Type>	**Container**	0...N (optional)

2.2.1.4.2.8 ServerId

The <ServerId> element specifies the server-unique identifier for a folder on the server.

Parent elements	Child elements	Data type	Number allowed
<Add> (response only)	None	**String** (Up to 64 characters)	0…N (optional)

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
<Delete> (response only) <Update> (response only)			

The <ServerId> element is used to identify folders that have been added, deleted, or updated on the server in the **FolderSync** command response.

The client MUST store the server ID as an opaque string of up to 64 characters.

Each <Add> or <Update> element included in a **FolderSync** response MUST contain one <ServerId> element.

2.2.1.4.2.9 ParentId

The <ParentId> element specifies the server ID of the parent of the folder on the server that has been added or updated.

Parent elements	Child elements	Data type	Number allowed
<Add> (response only) <Update> (response only)	None	**String** (Up to 64 characters)	0…N (optional)

The client MUST store the parent ID as an opaque string of up to 64 characters.

Each <Add> or <Update> element included in a **FolderSync** response MUST contain one <ParentId> element.

2.2.1.4.2.10 DisplayName

The <DisplayName> element specifies the name of the folder that is shown to the user.

Parent elements	Child elements	Data type	Number allowed
<Add> (response only) <Update> (response only)	None	**String**	0…N (optional)

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

2.2.1.4.2.11 Type

The <Type> element specifies the type of the folder that was added or updated (renamed or moved) on the server.

Parent elements	Child elements	Data type	Number allowed
<Add> (response only) <Update> (response only)	None	**Integer**	0…N (optional)c

Each <Add> or <Update> element included in a **FolderSync** response MUST contain one <Type> element.

Copyright © 2008 Microsoft Corporation.

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
16	User-created journal folder
17	User-created Notes folder
18	Unknown folder type
19	Recipient information cache

2.2.1.4.2.12 Update

The <Update> element is used in a **FolderSync** command response to identify a folder on the server that has been updated (renamed or moved).

Parent elements	Child elements	Data type	Number allowed
<Changes> (response only)	<ServerId> <ParentId> <DisplayName> <Type>	**Container**	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.

Copyright © 2008 Microsoft Corporation.

2.2.1.5 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> value of 3.

2.2.1.5.1 Request

The following code shows the XSD for the **FolderUpdate** command request.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
        xmlns:tns="FolderHierarchy:"
        attributeFormDefault="unqualified"
        elementFormDefault="qualified"
        targetNamespace="FolderHierarchy:"
        xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:element name="FolderUpdate">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="SyncKey">
                    <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="ParentId">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:maxLength value="64"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <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>
```


Copyright © 2008 Microsoft Corporation.

2.2.1.5.1.1 FolderUpdate

The <FolderUpdate> element is the top-level element in the XML stream. It indicates that the body of the HTTP POST contains a **FolderUpdate** command.

Parent elements	Child elements	Data type	Number allowed
None	<SyncKey> <ServerId> (request only) <ParentId> (request only) <DisplayName> (request only) <Status> (response only)	**Container**	1 (required)

2.2.1.5.1.2 SyncKey

The <SyncKey> element is used by the server to track the current state of the client.

Parent elements	Child elements	Data type	Number allowed
<FolderUpdate>	None	**String** (Up to 64 characters)	1 (required)

After a successful **FolderUpdate** command, the server MUST send a new synchronization key to the client. If the **FolderUpdate** command was not successful, the server MUST NOT return a <SyncKey> element.

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> provided in the request matches a <SyncKey> value on the server. The server MUST provide a <Status> value of 9 if the <SyncKey> values do not match.

The client MUST store the synchronization key as an opaque string of up to 64 characters.

2.2.1.5.1.3 ServerId

The <ServerId> element identifies the folder on the server to be renamed or moved.

Parent elements	Child elements	Data type	Number allowed
<FolderUpdate> (request only)	None	**String** (Up to 64 characters)	1 (required)

The server ID is obtained from the <ServerId> element of a previous **FolderSync** or **FolderUpdate** command. 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 object and MUST be able to locate an object given a <ServerId>.

The client MUST store the <ServerId> as an opaque string of up to 64 characters.

2.2.1.5.1.4 ParentId

The <ParentId> element specifies the server ID of the parent of the folder to be renamed or the destination folder of the folder to be moved.

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
<FolderUpdate> (request only)	None	**String** (Up to 64 characters)	1 (required)

The <ParentId> is obtained from the <ServerId> element of a previous **FolderSync** or **FolderCreate** command. The client MUST store the <ParentId> as an opaque string of up to 64 characters.

A <ParentId> of 0 signifies the mailbox root folder.

2.2.1.5.1.5 DisplayName

The <DisplayName> element specifies the name of the folder that is shown to the user.

Parent elements	Child elements	Data type	Number allowed
<FolderUpdate> (request only)	None	**String**	1 (required)

2.2.1.5.2 Response

The following code shows the XSD for the **FolderUpdate** command response.

```
<?xml version="1.0" ?>
<xs:schema xmlns:tns="FolderHierarchy:" attributeFormDefault="unqualified"
elementFormDefault="qualified"
targetNamespace="FolderHierarchy:" xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="FolderUpdate">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Status" type="xs:unsignedByte" />
                <xs:element minOccurs="0" name="SyncKey" type="xs:string" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

2.2.1.5.2.1 FolderUpdate

The **FolderUpdate** element is the top-level element in the XML stream. It indicates that the body of the HTTP **POST** response contains a **FolderUpdate** command.

Parent elements	Child elements	Data type	Number allowed
None	<SyncKey> <ServerId> (request only) <ParentId> (request only) <DisplayName> (request only) <Status> (response only)	**Container**	1 (required)

2.2.1.5.2.2 Status

The <Status> element indicates the success or failure of a **FolderUpdate** command request.

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
<FolderUpdate> (response only)	None	**Integer** (See values in the following table)	1 (required)

If the command fails, the <Status> element contains a code that indicates the type of failure.

The following table lists the valid values for this element.

Value	Meaning
1	Success.
2	A folder with that name already exists or the specified folder is a special folder, such as the Inbox, Outbox, Contacts, **Recipient** information, or Drafts folders, and cannot be renamed.
3	The specified folder is a special system folder, such as the Inbox, Outbox, Contacts, Recipient information, or Drafts folders, and cannot be updated by the client.
4	The specified folder does not exist.
5	The specified parent folder was not found.
6	An error occurred on the server.
8	The request timed out.
9	Synchronization key mismatch or invalid synchronization key.
10	Incorrectly formatted request.
11	An unknown error occurred.

2.2.1.5.2.3 SyncKey

The <SyncKey> element is used by the server to track the current state of the client.

Parent elements	Child elements	Data type	Number allowed
<FolderUpdate>	None	**String** (Up to 64 characters)	1 (required)

After a successful **FolderUpdate** command, the server MUST send a new synchronization key to the client. If the **FolderUpdate** command was not successful, the server MUST NOT return a <SyncKey> element.

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> provided in the request matches a <SyncKey> value on the server. The server MUST provide a <Status> value of 9if the <SyncKey> values do not match.

The client MUST store the synchronization key as an opaque string of up to 64 characters.

2.2.1.6 GetAttachment

The **GetAttachment** command retrieves an e-mail attachment from the server.<7>

Copyright © 2008 Microsoft Corporation.

Attachments are not automatically included with e-mail messages in a synchronization; they are explicitly retrieved 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 *AttachmentName* parameter MUST be base-64 encoded, as specified in [MS-ASHTTP] section 2.2.1.1.1.

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.

2.2.1.6.1 Request

No XML body is included in the **GetAttachment** command request.

2.2.1.6.2 Response

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.

No XML body is included in the **GetAttachment** command response.

2.2.1.7 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.

2.2.1.7.1 Request

The following code defines the XSD for the **GetItemEstimate** command request.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
        xmlns:tns="GetItemEstimate:"
        attributeFormDefault="unqualified"
        elementFormDefault="qualified"
        targetNamespace="GetItemEstimate:"
        xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:import namespace="AirSync:" />
    <xs:element name="GetItemEstimate">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Collections">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element maxOccurs="300" name="Collection">
                                <xs:complexType>
                                    <xs:sequence>
                                      <xs:element xmlns:q2="AirSync:" ref="q2:SyncKey" />
                                      <xs:element minOccurs="1" name="CollectionId">
```


Copyright © 2008 Microsoft Corporation.

```
                                                <xs:simpleType>
                                                    <xs:restriction base="xs:string">
                                                        <xs:maxLength value="64"/>
                                                    </xs:restriction>
                                                </xs:simpleType>
                                            </xs:element>
                                            <xs:element xmlns:q3="AirSync:"
ref="q3:ConversationMode" minOccurs="0" maxOccurs="1" />
                                            <xs:element xmlns:q5="AirSync:" ref="q5:Options"
minOccurs="0" maxOccurs="2">
                                              <xs:complexType>
                                                <xs:element xmlns:q6="AirSync:" ref="q6:Class"
minOccurs="0" maxOccurs="1" type="xs:string" />
                                                <xs:element xmlns:q7="AirSync:"
ref="q7:FilterType" minOccurs="0" maxOccurs="1" type="xs:integer" />
                                                <xs:element xmlns:q8="AirSync:" ref="q8:MaxItems"
minOccurs="0" maxOccurs="1" type="xs:integer" />
                                            <xs:complexType>
</xs:element>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

2.2.1.7.1.1 GetItemEstimate

The required element <GetItemEstimate> is the top-level element in the XML stream. It indicates that the body of the HTTP **POST** contains a **GetItemEstimate** command.

Parent elements	Child elements	Data type	Number allowed
None	<Collections> (request only) <Response> (response only)	**Container**	1 (required)

2.2.1.7.1.2 Collections

The <Collections> element serves as a container for one to 300 <Collection> elements.

Parent elements	Child elements	Data type	Number allowed
<GetItemEstimate> (request only)	<Collection>	**Container**	1 (required)

2.2.1.7.1.3 Collection

The <Collection> element wraps elements that apply to a particular collection. A maximum of 300 <Collection> elements can be included in a single <Collections> element.

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
<Collections> (request only) <Response> (response only)	<SyncKey> (request only) <CollectionId> <ConversationMode> (request only) <Options> (request only) <8> <Estimate> (response only)	**Container**	1...300 (required)

2.2.1.7.1.4 SyncKey

The required element <SyncKey> <9> represents the current state of a collection. The value of the element is examined by the server to determine the state of the synchronization process.

The <SyncKey> used within the <GetItemEstimate> requests is the same as the one returned within the **Sync** responses. The server does not update the <SyncKey> on **GetItemEstimate** requests. For more details about the <SyncKey> element, see section 2.2.1.19.1.11.The server checks the value of the key to verify that the value of the <SyncKey> provided in the request matches a <SyncKey> value on the server. The server MUST provide a Status value of 4if the <SyncKey> values do not match those expected within the **Sync** requests.

Parent elements	Child elements	Data type	Number allowed
<Collection> (request only)	None	**String** (Up to 64 characters)	1 (required)

2.2.1.7.1.5 CollectionId

The <CollectionId> element specifies the server ID of the collection from which the item estimate is being obtained.

Parent elements	Child elements	Data type	Number allowed
<Collection>	None	**String** (Up to 64 characters)	1...1 (required)

The <CollectionId> is obtained from the <ServerId> element of a previous **FolderSync** or **FolderCreate** command. The <CollectionId> element is used in both **GetItemEstimate** command requests and responses.

2.2.1.7.1.6 ConversationMode

The optional element <ConversationMode><10> specifies whether to include items that are included within the **conversation** modality within the results of the **GetItemEstimate** response. A single conversation MAY span multiple collections and therefore <ConversationMode> is a child of the <Collection> element, rather than the <Options> element.

Parent elements	Child elements	Data type	Number allowed
<Collection> (request only)	None	**Boolean**	0...1 (optional)

Specifying <ConversationMode> for collections that do not store e-mail results in an invalid XML error, status code 103.

Copyright © 2008 Microsoft Corporation.

2.2.1.7.1.7 Options

The <Options><11> element is a container that encloses elements that filter the results of the **GetItemEstimate** command.

Parent elements	Child elements	Data type	Number allowed
<Collection> (request only)	<Class> <FilterType> <MaxItems>	**Container**	0...2 (optional)

This element is optional; however, when it is present, it MUST include at least one child element. The <Options> element appears only in requests to the server from the client. If the <Options> element is not included in a request, then the **GetItemEstimate** command will enumerate all of the items within the collection, without any filter (up to a maximum of 512 items).

2.2.1.7.1.8 Class

The <Class> element<12> assigns the filters within the <Options> container to a given **class**.

Parent elements	Child elements	Data type	Number allowed
<Options> (request only)	None	**String**	0...1 (optional)

Options for the same class within the same collection MUST NOT be 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 class "SMS". To also sync e-mail messages at the same time, include another <Options> node with class "Email".

The valid <Class> element values are:

- *Tasks*
- *Email*
- *Calendar*
- *Contacts*
- *Document*
- *SMS*

2.2.1.7.1.9 FilterType

The <FilterType> element <13> specifies an optional time window in the **GetItemEstimate** command request for the objects sent from the server to the client.

Parent elements	Child elements	Data type	Number allowed
<Collection> <14> <Options>	None	**Integer**	0...1 (optional)

Copyright © 2008 Microsoft Corporation.

The <FilterType> applies to e-mail, calendar, task and calendar collections. If a filter type is specified, then the server sends an estimate of the items within the filter specifications.

If the <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. If the <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 <FilterType> value.

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

Value	Meaning	Applies to E-mail	Applies to calendar	Applies to tasks	Applies to contacts
0	No filter	Yes	Yes	No, <Status> value 110	Yes
1	1 day ago	Yes	No, <Status> value 110	No, <Status> value 110	Yes
2	3 days ago	Yes	No, <Status> value 110	No, <Status> value 110	Yes
3	1 week ago	Yes	No, <Status> value 110	No, <Status> value 110	Yes
4	2 weeks ago	Yes	Yes	No, <Status> value 110	Yes
5	1 month ago	Yes	Yes	No, <Status> value 110	Yes
6	3 months ago	No, <Status> value 110	Yes	No, <Status> value 110	Yes
7	6 months ago	No, <Status> value 110	Yes	No, <Status> value 110	Yes
8	Incomplete tasks	No, <Status> value 110	No, <Status> value 110	Yes	Yes

Specifying a <FilterType> of 9 or above for when the <CollectionId> identifies any e-mail, contact, calendar or task collection results in a <Status> value of 103.

2.2.1.7.1.10 MaxItems

The <MaxItems> element<15> specifies the maximum number of items to include in the response. This element can only be included in a request when the <CollectionId> is set to *RI* to specify a recipient information store; otherwise, the server will respond with a status 2 error. The value of <MaxItems> does not specify the limit of estimates available; rather, it only specifies the number of items, as a complete replacement would be double the number of items in the **store** (n deletes plus n additions).

Parent elements	Child elements	Data type	Number allowed
<Options>	None	**Integer**	0...1 (optional)

Copyright © 2008 Microsoft Corporation.

Including <MaxItems> when the <CollectionId> is set to anything other than *RI* results in an invalid XML error, status code 103.

2.2.1.7.2 Response

The following code shows the XSD for the **GetItemEstimate** command response.

```
<?xml version="1.0" ?>
<xs:schema xmlns:tns="GetItemEstimate:" attributeFormDefault="unqualified"
elementFormDefault="qualified"
targetNamespace="GetItemEstimate:" xmlns:xs="http://www.w3.org/2001/XMLSchema">
     <xs:element name="GetItemEstimate">
          <xs:complexType>
               <xs:sequence>
                    <xs:element name="Response">
                         <xs:complexType>
                              <xs:sequence>
                                   <xs:element name="Status" type="xs:unsignedByte" />
                                   <xs:element minOccurs="0" name="Collection">
                                        <xs:complexType>
                                             <xs:sequence>

                                                  <xs:element name="CollectionId"
type="xs:string" />

                                                  <xs:element name="Estimate"
type="xs:unsignedByte" />
                                             </xs:sequence>
                                        </xs:complexType>
                                   </xs:element>
                              </xs:sequence>
                         </xs:complexType>
                    </xs:element>
               </xs:sequence>
          </xs:complexType>
     </xs:element>
</xs:schema>
```

2.2.1.7.2.1 GetItemEstimate

The <GetItemEstimate> element is the top-level element in the XML stream. It indicates that the body of the HTTP **POST** response contains a **GetItemEstimate** command.

2.2.1.7.2.2 Response

The <Response> element wraps 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.

Parent elements	Child elements	Data type	Number allowed
<GetItemEstimate> (response only)	<Collection> <Status>	**Container**	1 (required)

Copyright © 2008 Microsoft Corporation.

2.2.1.7.2.3 Status

The <Status> element indicates the success or failure of a **GetItemEstimate** command request.

Parent elements	Child elements	Data type	Number allowed
<Response> (response only)	None	**Integer**	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 returned <Collection> elements. If one <Collection> fails, a failure status is returned for all collections.

The following table lists the valid values for the element.

Value	Meaning
1	Success.
2	A collection was invalid or one of the specified collection IDs was invalid.
3	Synchronization state has not been primed yet. The **Sync** command MUST be performed first.
4	The specified synchronization key was invalid

2.2.1.7.2.4 Collection

The <Collection> element wraps elements that apply to a particular collection. A maximum of 300 <Collection> elements can be included in a single <Collections> element.

Parent elements	Child elements	Data type	Number allowed
<Collections> (request only) <Response> (response only)	<SyncKey> (request only) <CollectionId> <FilterType> (request only) <Estimate> (response only)	**Container**	0...n (optional)

2.2.1.7.2.5 CollectionId

The <CollectionId> element specifies the server ID of the collection from which the item estimate is being obtained.

The collection ID is obtained from the <ServerId> element of a previous **FolderSync** or **FolderCreate** command. The <CollectionId> element is used in both **GetItemEstimate** command requests and responses.

2.2.1.7.2.6 Estimate

The <Estimate> element specifies the estimated number of items in the collection or folder that have to be synchronized.

Parent elements	Child elements	Data type	Number allowed
<Collection> (response only)	None	**Integer**	1 (required)

Copyright © 2008 Microsoft Corporation.

2.2.1.8 ItemOperations

The **ItemOperations** command acts as a container for the <Fetch>, <EmptyFolderContents>, and <Move> commands to provide batched online operations of these commands against the server.

Operations that are contained within the **ItemOperations** node 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 or **Universal Naming Convention (UNC)** document metadata, search results, and items and attachments.

The <EmptyFolderContents> operation enables the client to empty a folder of all its items. Clients use <EmptyFolderContents> specifically 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.

2.2.1.8.1 Delivery of Content Requested by Fetch

Because the **ItemOperations** response 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** command:

- Inline
- Multipart

Inline

The inline method of delivering binary content is including Base64-encoded data inside the WBXML. The inline approach generally requires the client to read the whole response into memory in order to parse it, thereby consuming a large amount of memory.

Multipart

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 RAM, because a file is read in pieces, one piece at a time.

The multipart approach enables the client to parse the small WBXML part, obtain references to the binary parts, and handle the binary parts as neccessary, without reading the entire response into memory.

Multipart Request

If the client wants to have the document or documents returned in multipart format, the only difference between this request and the inline content request is the addition of the following HTTP header:

```
MS-ASAcceptMultiPart: T
```


Copyright © 2008 Microsoft Corporation.

If this header is not present, then the server uses the default of FALSE, and returns content inline. If the header is set to TRUE, then the server returns the document contents by using the multipart format.

The following is a sample request for the test.txt document in a UNC share:

```
POST /Microsoft-Server-
ActiveSync?Cmd=ItemOperations&User=administrator&DeviceId=v140Device&DeviceType=PocketPC
Content-Type: application/vnd.ms-sync
MS-ASProtocolVersion: 14.0
MS-ASAcceptMultiPart: T
<ItemOperations>
 <Fetch>
  <Store>DocumentLibrary</Store>
  <LinkId>\\feod31\public\test.txt</LinkId>
 </Fetch>
</ItemOperations>
```

Multipart Response

At a high level, the multipart response consists of several key elements:

- HTTP headers that specify the content type (HTTP 'Content-Type' header) of the multipart response: application/vnd.ms-sync.multipart.
- Metadata consisting of a list of [integer, integer] tuples that specify the start and count of bytes, respectively, of each **body part**. The following is the format of the metadata:

```
'Number of Parts :' <number of body parts, including WBXML>
'Part' <part #> ':' <range>
```

Range specifies a [start, count] value that indicates the start and count of bytes for each body part. There is always at least one tuple, pointing to the WBXML response.

- The WBXML response, which contains status and application data for all requested items. The WBXML response is always the first part in the response. Items composed of binary content have a <Part> element that indicates the index (base 0) of the body part that corresponds to that item in the multipart structure. This index is used by the client to find the appropriate [start, count] entry in the metadata.
- Binary application data, which includes one or more binary data parts, the start and end byte of each of which is indicated in the WBXMLEX-Ranges header.

The following figure shows the elements of the multipart response.

Copyright © 2008 Microsoft Corporation.

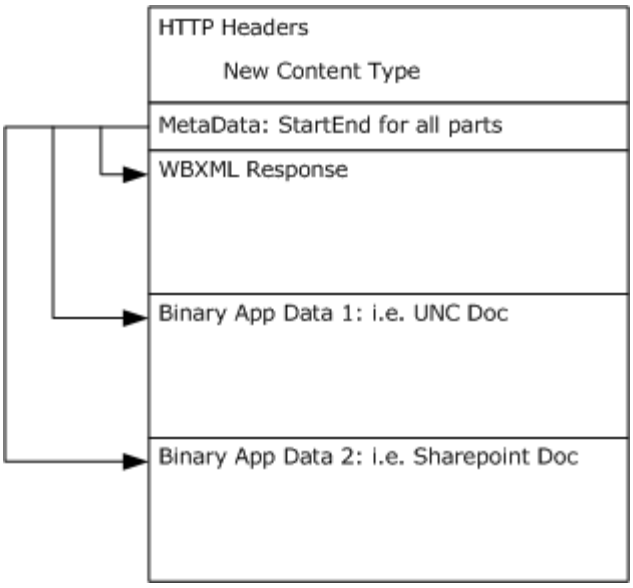

Figure 1: ItemOperations command multipart response

2.2.1.8.2 Request

The server that is implementing [MS-ASCMD] enforces the following XSD when it processes protocol requests.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
        id="ItemOperations"
        targetNamespace="ItemOperations:"
        xmlns:search="Search:"
        xmlns:calendar="Calendar:"
        xmlns:contacts2="Contacts2:"
        xmlns:contacts="Contacts:"
        xmlns:email="Email:"
        xmlns:mstns="ItemOperations:"
        xmlns:airsyncbase="AirSyncBase:"
        xmlns:documentLibrary="DocumentLibrary:"
        xmlns:airsync="AirSync:"
        xmlns:xs="http://www.w3.org/2001/XMLSchema"
        attributeFormDefault="qualified"
        elementFormDefault="qualified">

    <xs:import namespace="DocumentLibrary:"/>
    <xs:import namespace="AirSync:"/>
    <xs:import namespace="AirSyncBase:"/>
    <xs:import namespace="Email:"/>
    <xs:element name="ItemOperations">
        <xs:complexType>
            <xs:choice maxOccurs="unbounded">
                <xs:element name="EmptyFolderContents">
                    <xs:complexType>
                        <xs:all>
                            <xs:element ref="airsync:CollectionId" minOccurs="1"
maxOccurs="1"/>
```


Copyright © 2008 Microsoft Corporation.

```
                            <xs:element name="Options" minOccurs="0" maxOccurs="1">
                                <xs:complexType>
                                    <xs:all>
                                        <xs:element name="DeleteSubFolders"/>
                                    </xs:all>
                                </xs:complexType>
                            </xs:element>
                        </xs:all>
                    </xs:complexType>
                </xs:element>
                <xs:element name="Fetch">
                    <xs:complexType>
                        <xs:all>
                            <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>
                            <xs:element ref="airsync:ServerId" minOccurs="0" maxOccurs="1"/>
                            <xs:element ref="airsync:CollectionId" minOccurs="0"
maxOccurs="1"/>
                            <xs:element ref="documentLibrary:LinkId" minOccurs="0"
maxOccurs="1"/>
                            <xs:element ref="search:LongId" minOccurs="0" maxOccurs="1"/>
                            <xs:element ref="airsyncbase:FileReference" minOccurs="0"
maxOccurs="1"/>
                            <xs:element name="Options" minOccurs="0" maxOccurs="1">
                                <xs:complexType>
                                    <xs:choice maxOccurs="unbounded">
                                        <xs:element minOccurs="0" maxOccurs="1"
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="Range" minOccurs="0" maxOccurs="1">
                                            <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 minOccurs="0" maxOccurs="1"
name="UserName">
                                            <xs:simpleType>
```


Copyright © 2008 Microsoft Corporation.

```
                                                <xs:restriction base="xs:string">
                                                    <xs:maxLength value="100" />
                                                </xs:restriction>
                                            </xs:simpleType>
                                        </xs:element>
                                        <xs:element minOccurs="0" maxOccurs="1"
name="Password">
                                            <xs:simpleType>
                                                <xs:restriction base="xs:string">
                                                    <xs:maxLength value="100" />
                                                </xs:restriction>
                                            </xs:simpleType>
                                        </xs:element>
                                        <xs:element ref="airsync:MIMESupport" minOccurs="0"
maxOccurs="1" />
                                        <xs:element ref="airsyncbase:BodyPreference"
minOccurs="0" maxOccurs="256" />
                                    </xs:choice>
                                </xs:complexType>
                            </xs:element>
                        </xs:all>
                    </xs:complexType>
                </xs:element>
                <xs:element name="Move">
                    <xs:complexType>
                        <xs:all>
                            <xs:element name="ConversationId" type="xs:string" minOccurs="1"
maxOccurs="1"/>
                            <xs:element name="DstFldId" minOccurs="1" maxOccurs="1">
                                <xs:simpleType>
                                    <xs:restriction base="xs:string">
                                        <xs:maxLength value="64"/>
                                        <xs:minLength value="1"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                            <xs:element name="Options" minOccurs="0" maxOccurs="1">
                                <xs:complexType>
                                    <xs:all>
                                        <xs:element name="MoveAlways" minOccurs="0"
maxOccurs="1"/>
                                    </xs:all>
                                </xs:complexType>
                            </xs:element>
                        </xs:all>
                    </xs:complexType>
                </xs:element>
            </xs:choice>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

2.2.1.8.2.1 ItemOperations

The <ItemOperations> element is the top-level element in the XML document. The element identifies the body of the HTTP **POST** as containing an **ItemOperations** command.

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
None	<Fetch> (request only) <EmptyFolderContents> (request only) <Move> (request only) <Response> (response only) <Status> (response only)	**Container**	1…1 (required)

2.2.1.8.2.2 Fetch

The <Fetch> element retrieves an item from the server.

Parent elements	Child elements	Data type	Number allowed
<ItemOperations> (request) <Response> (response)	<Store> (request only) <LinkId> (optional) <LongId> (optional) <CollectionId> (optional) <ServerId> (optional) <Options> <Status> (response only) <Class> (response only) <Properties> (response only) <FileReference> (request only) <LongId> (request only)	**Container**	0…N (optional)

The <Fetch> response <Status> element uses the same values as the parent **ItemOperations** response <Status> element. For more details, see section 2.2.1.8.3.12.

The <Fetch> operation is intended to be used on Microsoft Windows® SharePoint® Services technology or UNC document metadata, search results, and items and attachments.

There are two methods to deliver content that is requested by the **Fetch** operation, Inline and Multiple, as specified in section 2.2.1.8.1. The client can choose the delivery method that is most efficient for its implementation.

In the request, the client specifies the location and a byte range for the item. The location is indicated by either a link ID (<LinkId> element) if the target item is identified by a **Uniform Resource Identifier (URI)**, or a file reference (<FileReference> element) if the client is retrieving an e-mail attachment.<16>

The <Fetch> command supports several options, including:

- Byte ranges—The range of bytes for an item that is contained in a given <Fetch> command response. The specified range facilitates a checkpoint to improve the reliability of large data downloads. This option is supported for document library items and attachments; it is not supported for other item types.
- Body preference—Per-class settings on preferred body format. It is supported only for e-mail, contact, calendar, and task items; it is not supported for document library items or attachments.

Copyright © 2008 Microsoft Corporation.

- Schema—Per-class settings on format for search results. It is supported only for e-mail, contact, calendar, and task items; it is not supported for document library items or attachments.

The 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.

2.2.1.8.2.3 EmptyFolderContents

The <EmptyFolderContents> element identifies the body of the request or response as containing the operation that deletes the contents of a folder.

Parent elements	Child elements	Data type	Number allowed
<ItemOperations> (request only) <Response> (response only)	<CollectionId> <Options> (request only) <Status> (response only)	**Container**	0...1 (optional)

The <EmptyFolderContents> response <Status> element uses the same values as the parent **ItemOperations** response <Status> element. For more details, see section 2.2.1.8.3.12.

The <EmptyFolderContents> element enables the client to empty a folder of all its items. The element supports a single option, which is whether to delete subfolders contained in the folder (the default is not to delete subfolders).

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 (generally indicated by the return of an HTTP 507 status code from the server).

2.2.1.8.2.4 CollectionId

The <CollectionId> element enables a client to specify the folder to be emptied or the item to be fetched.

Parent elements	Child elements	Data type	Number allowed
<EmptyFolderContents> <Fetch> (request only)	None	**String**	0...1

2.2.1.8.2.5 Options

The <Options> element contains the options for its parent element. The child elements of <Options>, therefore, depend on its parent element and the store/item type that is being acted upon.

Parent elements	Child elements, <EmptyFolderContents> parent	Child element, <Fetch> parent	Child elements, <Move> parent	Data type	Number allowed
<EmptyFolderContents>	<DeleteSubFolders>	<Range> <Schema>	<MoveAlways>	**Container**	0...1 (option

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements, <EmptyFolderContents> parent	Child element, <Fetch> parent	Child elements, <Move> parent	Data type	Number allowed
<Fetch>		<UserName> <Password> <airsyncbase:BodyPreference> <airsync:MIMESupport>			al)

The following options are supported for <Fetch>:

- Byte ranges
  - Facilitates a checkpoint to improve the reliability of large data downloads.
  - [MS-ASCMD] supports ranges for document library items and attachments; it does not support ranges for other item types—that is, **Personal Information Manager (PIM)** items, such as e-mail, contact, calendar, or task items.
  - For attachments, the range applies to the file content.
  - For document library items, this applies to the file content.
- Body preference
  - Per-class settings on preferred body format.
  - [MS-ASCMD] supports body preferences for PIM items only; it does not support body preferences for document library items or attachments.
- Schema
  - Per-class settings on format for search results.
  - [MS-ASCMD] supports schemas for PIM items only; it does not support schemas for document library items or attachments.
  - Supports all top-level **property** nodes.

If the <FileReference> element is present, then <Range> is the only valid child element of <Options>.

If you specify an option that is invalid for the parent command, the server returns a protocol error.

2.2.1.8.2.6 DeleteSubFolders

The <DeleteSubFolders> element is a flag that indicates whether to delete the subfolders of the specified folder.

Parent elements	Child elements	Data type	Number allowed
<Options>	None	**Flag**	1...1 (required)

Copyright © 2008 Microsoft Corporation.

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

2.2.1.8.2.7 Store

The <Store> element specifies the name of the store to which the parent operation applies.

Parent elements	Child elements	Data type	Number allowed
<Fetch> (request only)	None	**String**	1...1 (required)

The following values are valid for the store element:

- Document Library (SharePoint and UNC links)
- Mailbox (items and attachments)

2.2.1.8.2.8 MIMESupport

The <MIMESupport> element is included in the <Options> element of a client <Fetch> command request to enable **MIME** support for e-mail items that are sent from the server to the client. For an example, see section 4.5.2.

Parent elements	Child elements	Data type	Number allowed
<Options> (request only)	None	**Unsigned Byte**	0...1 (optional)

The following table lists the valid values for this element.

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 <Fetch> request MUST include the following elements in the <Options> element:

- The <MIMESupport> element to indicate to the server to return MIME for S/MIME-only messages, all messages, or no messages.
- The <BodyPreference> element with its child element, <Type> 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 <Body> element, which is a child of the <Properties> element. The <Body> element is a complex element and MUST contain the following child nodes in an S/MIME <Fetch> response:

- The <Type> element with a value of 4 to inform the device that the data is a MIME BLOB.
- The <EstimatedDataSize> element to specify the rough total size of the data.
- The <Data> element that contains the full MIME BLOB.

Copyright © 2008 Microsoft Corporation.

For more details about the <Body> element or the <BodyPreference> element, see [MS-ASAIRS] section 2.2.1.3 or 2.2.1.4, respectively.

2.2.1.8.2.9 LinkId

The <LinkId> element specifies a URI that is assigned by the server to certain resources, such as Windows SharePoint Services or UNC documents.

Parent elements	Child elements	Data type	Number allowed
<Fetch>	None	**URI**	0...1 (optional)

The client MUST store the <LinkID> that is retrieved by the **Sync** or **Search** command if the client will make calls by using the <LinkID> in the future. In an **ItemOperations** request, the <LinkId> element can be used by the <Fetch> command to refer to the location of an item.

2.2.1.8.2.10 LongId

The <LongId> element specifies a unique identifier that was assigned by the server to each result returned by a previous **Search** response.

Parent elements	Child elements	Data type	Number allowed
<Fetch>	None	**String** (up to 256 characters)	0...1 (optional)

2.2.1.8.2.11 ServerId

The <ServerId> element 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.

Parent elements	Child elements	Data type	Number allowed
<Fetch>	None	**String**	0...1 (optional)

The client MUST store the <ServerId> for any item that is retrieved by means of the **Sync** or **Search** command. In an **ItemOperations** request, the <ServerId> element can be used by the <Fetch> command to refer to the location of the item in question.

2.2.1.8.2.12 FileReference

The <FileReference> element specifies a unique identifier that is assigned by the server to each attachment to a given item.

Parent elements	Child elements	Data type	Number allowed
<Fetch> (request only)	None	**String**	0...1 (optional)

The client MUST store the file reference for any item that is retrieved by means of the **Sync** or **Search** command. In an **ItemOperations** request, only one <FileReference> identifier can exist per <Fetch> node. Violation of this constraint results in an error from the server. The client can, however, retrieve multiple attachments by using one <Fetch> node per attachment.

If the <FileReference> element is present, then <Range> is the only valid child element of <Options>.

Copyright © 2008 Microsoft Corporation.

2.2.1.8.2.13 Schema

The <Schema> element specifies the schema of the item to be fetched.

Parent elements	Child elements	Data type	Number allowed
<Options> (request only)	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 3.1.5.1and [MS-ASTASK] section 3.1.5.1.	**Container**	0...1 (optional)

The <Schema> element is supported within options for PIM <Fetch> requests. It is not supported when the client is retrieving items from a document library or retrieving an attachment.

If <Schema> is not specified, the server allows all properties to be retrieved.

2.2.1.8.2.14 Range

In an <ItemOperations> request, the <Range> element specifies the range of bytes that the client can receive in response to the <Fetch> operation for a document library item. In an <ItemOperations> response, the <Range> element specifies the actual range of bytes for an item that is contained in a given <Fetch> operation.

Parent elements	Child elements	Data type	Number allowed
<Options> (request) <Properties> (response)	None	String in the format m-n, where m<n, and m is the minimum value and n is the maximum value.	0...1 (optional)

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 <Range> is omitted in the <Fetch> request, the whole item is fetched.

If the <FileReference> element is present, then <Range> is the only valid child element of <Options>.

2.2.1.8.2.15 UserName

The <UserName> element specifies the username of the account leveraged to fetch the desired item. The <Password> element contains the corresponding account password.

Parent elements	Child elements	Data type	Number allowed
<Options> (request only)	None	**String** (up to 100 characters)	0...1 (optional)

2.2.1.8.2.16 Password

The <Password> element specifies the password for the given <UserName>.

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
<Options> (request only)	None	**String** (up to 100 characters)	0...1 (optional)

2.2.1.8.2.17 Move

The <Move> element identifies the body of the request or response as containing the operation that moves a given conversation.<17>

Parent elements	Child elements	Data type	Number allowed
<ItemOperations> (request only) <Response> (response only)	<ConversationId> (request only) <DstFldId> (request only) <Options> (request only) <Status> (response only)	**Container**	0...n (optional)

The <Move> response <Status> element uses the same values as the parent **ItemOperations** response <Status> element. For more details, see section 2.2.1.8.3.12.

2.2.1.8.2.18 ConversationId

The <ConversationId> element specifies the conversation to be moved.<18>

Parent elements	Child elements	Data type	Number allowed
<Move> (request only)	None	**String**	1...1 (required)

2.2.1.8.2.19 DstFldId

The <DstFldId> element specifies the destination folder where the conversation is to be moved.<19>

Parent elements	Child elements	Data type	Number allowed
<Move> (request only)	None	**String**	1...1 (required)

2.2.1.8.2.20 MoveAlways

The <MoveAlways> element is a flag that indicates whether to always move the specified conversation, including all future e-mails in the conversation, to the <DstFldId> folder.<20>

Parent elements	Child elements	Data type	Number allowed
<Options> (request only)	None	**Boolean**	0...1 (optional)

2.2.1.8.3 Response

The following code shows the XSD for the **ItemOperations** command response.

```
<?xml version="1.0" ?>
<xs:schema xmlns:tns="ItemOperations:"
attributeFormDefault="unqualified" elementFormDefault="qualified"
```


Copyright © 2008 Microsoft Corporation.

```
targetNamespace="ItemOperations:" xmlns:DocumentLibrary ="DocumentLibrary:"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:import namespace="DocumentLibrary:" />
  <xs:element name="ItemOperations">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Status" type="xs:integer " minOccurs="1" maxOccurs="1" />
        <xs:element name="Response" minOccurs="0" maxOccurs="1">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Move" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                    <xs:element name="Status" type="xs:integer" minOccurs="1" maxOccurs="1"
/>
                </xs:complexType>
              </xs:element>
              <xs:element name="EmptyFolderContents" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Status" type="xs:integer" minOccurs="1" maxOccurs="1"
/>
                    <xs:element ref="AirSync:CollectionId" minOccurs="1" maxOccurs="1" />
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
               <xs:element name="Fetch" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Status" type="xs:integer" minOccurs="1" maxOccurs="1"/>
                    <xs:element ref="AirSync:CollectionId" minOccurs="1" maxOccurs="1" />
                    <xs:element ref="AirSync:ServerId type="xs:string" minOccurs="0"
maxOccurs="1"/>
                    <xs:element ref="AirSync:Class type="xs:string" minOccurs="0"
maxOccurs="1"/>
                    <xs:element ref="DocumentLibrary:LinkId" minOccurs="0" maxOccurs="1"/>
                    <xs:element name="Properties" minOccurs="0" maxOccurs="1">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="Range" type="xs:string" minOccurs="0"
maxOccurs="1"/>
                          <xs:element name="Total" type="xs:integer" minOccurs="0"
maxOccurs="1" />
                          <xs:element name="Data" type="xs:string" minOccurs="0"
maxOccurs="1" />
                          <xs:element name="Version" type="xs:datetime" minOccurs="0"
maxOccurs="1"/>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
 </xs:schema>
```


Copyright © 2008 Microsoft Corporation.

2.2.1.8.3.1 ItemOperations

The <ItemOperations> element is the top-level element in the XML document. The element identifies the body of the HTTP **POST** response as containing an **ItemOperations** command.

Parent elements	Child elements	Data type	Number allowed
None	<Fetch> (request only) <EmptyFolderContents> (request only) <Move> (request only) <Response> (response only) <Status> (response only)	**Container**	1…1 (required)

2.2.1.8.3.2 Response

The <Response> element is a container for the operation responses.

Parent elements	Child elements	Data type	Number allowed
<ItemOperations> (response only)	<EmptyFolderContents> <Fetch> <Move>	**Container**	0…1 (optional)

2.2.1.8.3.3 EmptyFolderContents

The <EmptyFolderContents> element identifies the body of the request or response as containing the operation that deletes the contents of a folder.

Parent elements	Child elements	Data type	Number allowed
<ItemOperations> (request only) <Response> (response only)	<CollectionId> <Options> (request only) <Status> (response only)	<Container>	0…N (optional)

2.2.1.8.3.4 CollectionId

The <CollectionId> element enables a client to specify the folder to be emptied.

Parent elements	Child elements	Data type	Number allowed
<EmptyFolderContents>	None	**String**	1…1

2.2.1.8.3.5 ServerId

The <ServerId> element 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.

Parent elements	Child elements	Data type	Number allowed
<Fetch>	None	**String**	0…1 (optional)

Copyright © 2008 Microsoft Corporation.

The client MUST store the <ServerId> for any item that is retrieved by means of the **Sync** or **Search** command. In an **ItemOperations** request, the <ServerId> element can be used by the <Fetch> command to refer to the location of the item in question.

2.2.1.8.3.6 Fetch

The <Fetch> element retrieves an item from the server.

Parent elements	Child elements	Data type	Number allowed
<ItemOperations> (request) <Response> (response)	<Store> (request only) <LinkId> (optional) <LongId> (optional) <CollectionId> (optional) <ServerId> (optional) <Options> <Status> (response only) <Class> (response only) <Properties> (response only) <FileReference> (request only)	**Container**	0...N (optional)

The **Fetch** operation is intended to be used on Microsoft 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, the [MS-ASCMD] 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** command:

- Inline—The binary content is Base64-encoded 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 RAM.

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 (**LinkId** element) if the target item is identified by a URI, or a file reference (**FileReference** element) if the client is retrieving an e-mail attachment.<21>

The **Fetch** command supports several options, including:

- Byte ranges—The range of bytes for an item that is contained in a given **Fetch** command response. The specified range facilitates a checkpoint to improve the reliability of large data **downloads**. This option is supported for **document** library items and attachments; it is not supported for other item types.
- Body preference—Per-class settings on preferred body format. It is supported only for e-mail, contact, calendar, and task items; it is not supported for document library items or attachments.

Copyright © 2008 Microsoft Corporation.

- Schema—Per-class settings on format for search results. It is supported only for e-mail, contact, calendar, and task items; it is not supported for document library items or attachments.

The 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.

2.2.1.8.3.7 LinkId

The <LinkId> element specifies a URI that is assigned by the server to certain resources, such as Windows SharePoint Services or UNC documents.

Parent elements	Child elements	Data type	Number allowed
<Fetch>	None	**URI**	0...1 (optional)

The client MUST store the <LinkID> that is retrieved by the **Sync** or **Search** command if the client will make calls by using the <LinkID> in the future. In an **ItemOperations** request, the <LinkId> element can be used by the <Fetch> command to refer to the location of an item.

2.2.1.8.3.8 LongId

The <LongId> element specifies a unique identifier that was assigned by the server to each result returned by a previous **Search** response.

Parent elements	Child elements	Data type	Number allowed
<Fetch>	None	**String** (up to 256 characters)	0...1 (optional)

2.2.1.8.3.9 Class

In a response, the <Class> element indicates the class of the content of the fetched item.

Parent elements	Child elements	Data type	Number allowed
<Fetch> (response only)	None	**String**	0...1 (optional)

The following are valid values for the <Class> element in a request or response.

- *E-mail*
- *Contacts*
- *Calendar*
- *Tasks*

2.2.1.8.3.10 Properties

The <Properties> element contains a list of the schema properties for the item that the client wants to have returned in the <Fetch> response.

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
<Schema> (request only) <Fetch> (response only)	The schema properties of the item being fetched (request only) <Data> (response only) <Part> (response only) <Version> (response only) <Total> (response only) <airsyncbase:Body> (response only) 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].	**Container**	0...1 (optional)

If an unsupported property is specified by the client, the server returns an error. If <Properties> is not specified, the server uses the synchronized schema for that item class for <Fetch> results.

2.2.1.8.3.11 Range

In an **ItemOperations** request, the <Range> element specifies the range of bytes that the client can receive in response to the <Fetch> operation for a document library item. In an **ItemOperations** response, the <Range> element specifies the actual range of bytes for an item that is contained in a given <Fetch> operation.

Parent elements	Child elements	Data type	Number allowed
<Options> (request) <Properties> (response)	None	**String** 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).	0...1 (optional)

The server provides a best effort at fulfilling the request. Therefore, the client cannot assume that the byte-range specified in the request will exactly match the byte-range returned in the response. The byte-range that is specified by the server in the response is the authoritative value.

If <Range> is omitted in the <Fetch> request, the entire item is fetched.

2.2.1.8.3.12 Status

The <Status> element contains a code that indicates the success or failure of the **ItemOperations** command and the operations within the **ItemOperations** command.

Parent elements	Child elements	Data type	Number allowed
<EmptyFolderContents> <Fetch> <Move> <ItemOperations>	None	**Integer**	0...1

The following table lists the different status codes.

Copyright © 2008 Microsoft Corporation.

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.
7	Document library - Failed to connect to the server.
8	Document library - The byte-range is invalid or too large.
9	Document library - The store is unknown or unsupported.
10	Document library - The file is empty.
11	Document library - The requested data size is too large.
12	Document library - 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	Policy-related - Server blocked access.
17	Empty folder contents - Partial success; the command completed partially.
18	Credentials required.
155	Protocol error. The <Options> element does not contain a <MoveAlways> element.
156	Action not supported. The destination folder cannot be Drafts, Junk E-mail, or Outbox.

The status is specified for the **ItemOperations** response, and for each fetch operation, empty-folder-contents operation, or move operation within the **ItemOperations** command.

2.2.1.8.3.13 Data

The <Data> element is part of the response for the <Fetch> command and contains the item content for inline content responses.

Parent elements	Child elements	Data type	Number allowed
<Properties> (response only)	None	**String**	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 in the fetch response. The total size of the item (in bytes) is indicated by the <Total> element.

Copyright © 2008 Microsoft Corporation.

2.2.1.8.3.14 Part

The <Part> element specifies an integer index into the metadata of the multipart response.

Parent elements	Child elements	Data type	Number allowed
<Properties> (response only)	None	**Integer**	0...1 (optional)

The <Part> element is present only in a multipart **ItemOperations** response.

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.

2.2.1.8.3.15 Version

The <Version> element is a date/time stamp that indicates the time at which a document item was last modified.

Parent elements	Child elements	Data type	Number allowed
<Properties> (response only)	None	**DateTime**	0...1 (optional)

The <Version> element is present only in the response and only when **ItemOperations** is used to access a Windows SharePoint Services or UNC resource.

2.2.1.8.3.16 Total

The <Total> element indicates the total size of an item on the server, in bytes.

Parent elements	Child elements	Data type	Number allowed
<Properties> (response only)	None	**Integer**	0...1 (optional)

2.2.1.8.3.17 Move

The <Move> element identifies the body of the request or response as containing the operation that moves a conversation to a folder.

Parent elements	Child elements	Data type	Number allowed
<ItemOperations> (request only) <Response> (response only)	<ConversationId> (request only) <Options> (request only) <Status> (response only)	**Container**	0...N (optional)

2.2.1.9 MeetingResponse

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

The **MeetingResponse** command can only be used when the <CollectionId> element is being used to synchronize the folder that contains the meeting request item.

Copyright © 2008 Microsoft Corporation.

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

2.2.1.9.1 Request

The following code shows the XSD for the **MeetingResponse** command request.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
        xmlns:tns="MeetingResponse:"
        attributeFormDefault="unqualified"
        elementFormDefault="qualified"
        targetNamespace="MeetingResponse:"
        xmlns:search="Search:"
        xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:element name="MeetingResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element maxOccurs="unbounded" name="Request">
                    <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 minOccurs="0" name="CollectionId">
                                <xs:simpleType>
                                    <xs:restriction base="xs:string">
                                        <xs:maxLength value="64"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                            <xs:element minOccurs="0" name="RequestId">
                                <xs:simpleType>
                                    <xs:restriction base="xs:string">
                                        <xs:maxLength value="64"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                            <xs:element ref="search:LongId" minOccurs="0" maxOccurs="1"/>
                        </xs:all>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
```


Copyright © 2008 Microsoft Corporation.

2.2.1.9.1.1 MeetingResponse

The <MeetingResponse> element is the top-level element in the XML document. The element identifies the body of the HTTP **POST** as containing a **MeetingResponse** command.

Parent elements	Child elements	Data type	Number allowed
None	<Request> (request only)	**Container**	1…1 (required)

2.2.1.9.1.2 CollectionId

The <CollectionId> element specifies the folder that contains the meeting request.

Parent elements	Child elements	Data type	Number allowed
<Request> (request only)	None	**String** (Up to 64 characters)	0…1 (Required, or optional if <Search:LongId> is included, as specified in 2.2.1.14.2.1)

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

The <CollectionId> is obtained from the <ServerId> element of a previous **FolderSync** or **FolderCreate** command.

2.2.1.9.1.3 Request

The <Request> element is a container for elements in a **MeetingResponse** command request. 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.

Parent elements	Child elements	Data type	Number allowed
<MeetingResponse> (request only)	<UserResponse> <CollectionId> <RequestId> <Search:LongId> as specified in 2.2.1.14.2.1	**Container**	1…n (required)

2.2.1.9.1.4 RequestId

The <RequestId> element specifies the server ID of the meeting request message item.

Parent elements	Child elements	Data type	Number allowed
<Request> (request only) <Result> (response only)	None	**String** (Up to 64 characters)	0…1 (Required, or optional if <Search:LongId> is included, as specified in 2.2.1.14.2.1)

Copyright © 2008 Microsoft Corporation.

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.

2.2.1.9.1.5 UserResponse

The <UserResponse> element indicates in the **MeetingResponse** command request whether the meeting is being accepted, tentatively accepted, or declined.

Parent elements	Child elements	Data type	Number allowed
<Request> (request only)	None	**Integer**	1 (required)

The following table shows valid values for the element.

Value	Meaning
1	Accepted
2	Tentatively accepted
3	Declined

2.2.1.9.2 Response

The following code shows the XSD for the **MeetingResponse** command response.

```
<?xml version="1.0" ?>
<xs:schema xmlns:tns="MeetingResponse:" attributeFormDefault="unqualified"
elementFormDefault="qualified"
     targetNamespace="MeetingResponse:" xmlns:xs="http://www.w3.org/2001/XMLSchema">
     <xs:element name="MeetingResponse">
          <xs:complexType>
               <xs:sequence>
                    <xs:element maxOccurs="unbounded" name="Result">
                         <xs:complexType>
                              <xs:sequence>
                                   <xs:element name="RequestId" type="xs:string" />
                                   <xs:element name="Status" type="xs:unsignedByte" />
                                   <xs:element name="CalendarId" type="xs:string"
minOccurs="0" />
                              </xs:sequence>
                         </xs:complexType>
                    </xs:element>
               </xs:sequence>
          </xs:complexType>
     </xs:element>
</xs:schema>
```

2.2.1.9.2.1 MeetingResponse

The <MeetingResponse> element is the top-level element in the XML document. The element identifies the body of the HTTP **POST** as containing a **MeetingResponse** command.

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
None	<Result> (response only)	**Container**	1…1 (required)

2.2.1.9.2.2 CalendarId

The <CalendarId> element specifies the server ID of the calendar item.

Parent elements	Child elements	Data type	Number allowed
<Result> (response only)	None	**String** (Up to 64 characters)	0…1

The following table shows valid values for the element.

Value	Meaning
1	Success

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>.

2.2.1.9.2.3 RequestId

The <RequestId> element specifies the server ID of the meeting request message item.

Parent elements	Child elements	Data type	Number allowed
<Request> (request only) <Result> (response only)	None	**String** (Up to 64 characters)	1 (required)

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.

2.2.1.9.2.4 Result

The <Result> element is a container for elements that are sent to the client in a **MeetingResponse** command response.

Parent elements	Child elements	Data type	Number allowed
<MeetingResponse> (response only)	<RequestId> <Status> <CalendarId>	**Container**	1…n (required)

Copyright © 2008 Microsoft Corporation.

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.

2.2.1.9.2.5 Status

The <Status> element indicates the success or failure of the **MeetingResponse** command request.

Parent elements	Child elements	Data type	Number allowed
<Result> (response only)	None	**Integer**	1...n (required)

The following table shows valid values for the element.

Value	Meaning
1	Success.
2	Invalid meeting request.
3	An error occurred on the mailbox.
4	Error occurred on the server.

2.2.1.10 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. 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> and <Add> 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.

2.2.1.10.1 Request

The following code shows the XSD for the **MoveItems** command request.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
        xmlns:tns="Move:"
        attributeFormDefault="unqualified"
        elementFormDefault="qualified"
        targetNamespace="Move:"
        xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:element name="Moves">
        <xs:complexType>
            <xs:sequence>
                <xs:element maxOccurs="unbounded" name="Move">
                    <xs:complexType>
```


Copyright © 2008 Microsoft Corporation.

```
                        <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>
```

2.2.1.10.1.1 DstFldId

The <DstFldId> element specifies the server ID of the destination folder (that is, the folder to which the items are moved).

Parent elements	Child elements	Data type	Number allowed
<Move> (request only)	None	**String** (Up to 64 characters)	1 (required)

The server ID of the destination folder is obtained from the <ServerId> element of a previous **FolderSync** or **FolderCreate** command.

2.2.1.10.1.2 Moves

The <Moves> element is the top-level element in the XML document. It identifies the body of the HTTP **POST** as containing a **MoveItems** command.

Parent elements	Child elements	Data type	Number allowed
None	<Move> (request only) <Response> (response only)	**Container**	1 (required)

Copyright © 2008 Microsoft Corporation.

2.2.1.10.1.3 Move

The <Move> element is a container for elements that describe details of the items to be moved.

Parent elements	Child elements	Data type	Number allowed
<Moves> (request only)	<SrcMsgId> <SrcFldId> <DstFldId>	**Container**	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.

2.2.1.10.1.4 SrcFldId

The <SrcFldId> element specifies the server ID of the source folder (that is, the folder that contains the items to be moved).

Parent elements	Child elements	Data type	Number allowed
<Move>	None	**String** (Up to 64 characters)	1 (required)

The server ID of the source folder is obtained from the <ServerId> element of a previous **FolderSync** or **FolderCreate** command.

2.2.1.10.2 Response

The following code shows the XSD for the **MoveItems** command response.

```
<?xml version="1.0" ?>
<xs:schema xmlns:tns="Move:" attributeFormDefault="unqualified"
elementFormDefault="qualified" targetNamespace="Move:"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
     <xs:element name="Moves">
          <xs:complexType>
               <xs:sequence>
                    <xs:element maxOccurs="unbounded" name="Response">
                         <xs:complexType>
                              <xs:sequence>
                                   <xs:element name="SrcMsgId" type="xs:string" />
                                   <xs:element name="Status" type="xs:unsignedByte" />
                                   <xs:element minOccurs="0" name="DstMsgId" type="xs:string"
/>
                              </xs:sequence>
                         </xs:complexType>
                    </xs:element>
               </xs:sequence>
          </xs:complexType>
     </xs:element>
</xs:schema>
```

2.2.1.10.2.1 DstMsgId

The <DstMsgId> element specifies the new server ID of the item after the item is moved to the destination folder.

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
<Response> (response only)	None	**String** (Up to 64 characters)	0…1 (optional)

2.2.1.10.2.2 Moves

The <Moves> element is the top-level element in the XML document. It identifies the body of the HTTP **POST** as containing a **MoveItems** command.

Parent elements	Child elements	Data type	Number allowed
None	<Move> (request only) <Response> (response only)	**Container**	1 (required)

2.2.1.10.2.3 Response

The <Response> element serves as a container for elements that describe the moved items.

Parent elements	Child elements	Data type	Number allowed
<Moves> (response only)	<SrcMsgId> <Status> <DstMsgId>	**Container**	1 (required)

2.2.1.10.2.4 SrcMsgId

The <SrcMsgId> element specifies the server ID of the item to be moved.

Parent elements	Child elements	Data type	Number allowed
<Move> (request only) <Response> (response only)	None	**String** (Up to 64 characters)	1 (required)

2.2.1.10.2.5 Status

The <Status> element indicates the success or failure of an item moved. If the command failed, <Status> contains a code indicating the type of failure.

Parent elements	Child elements	Data type	Number allowed
<Response> (response only)	None	**Integer**	1 (required)

The following table shows valid values for the element.

Value	Meaning
1	Invalid source collection ID.
2	Invalid destination collection ID.
3	Success.

Copyright © 2008 Microsoft Corporation.

Value	Meaning
4	Source and destination collection IDs are the same.
5	A failure occurred during the **MoveItem** command.
7	Source or destination item was locked.

2.2.1.11 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.

The server does not immediately issue a response to the client's **Ping** request. Instead, the server waits until one of two events occur: either the time-out that is specified by the client elapses, or changes occur in one of the folders that the client specifies. The response that the server issues indicates which of these events has happened so that the client can react appropriately.

The server uses the last <SyncKey> returned to the client when determining to report in the **Ping** response. Therefore the client MUST have received the response to its last **Sync** request and successfully applied the changes sent by the server, prior to issuing a **Ping** request.

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, the client runs 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. This loop is called the heartbeat. The length of time that the server waits before issuing a response is called the heartbeat interval.

To reduce the amount of data sent in a **Ping** command request, the server caches the heartbeat interval and folder list. The client can omit the heartbeat interval, the folder list, 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.

If the **Ping** element is specified in an XML request body, either the <HeartbeatInterval> element or the <Folders> element or both MUST be specified.

2.2.1.11.1 Request

A **Ping** command can be sent with no body, in which case the cached version is used. This XSD is applied only to requests with a body.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
        xmlns:tns="Ping:"
        attributeFormDefault="unqualified"
        elementFormDefault="qualified"
```


Copyright © 2008 Microsoft Corporation.

```
        targetNamespace="Ping:"
        xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:element name="Ping">
        <xs:complexType>
            <xs:all>
                <xs:element name="HeartbeatInterval" minOccurs="0" maxOccurs="1">
                    <xs:simpleType>
                        <xs:restriction base="xs:integer"/>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="Folders" minOccurs="0" maxOccurs="1">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="Folder" minOccurs="1" maxOccurs="unbounded">
                                <xs:complexType>
                                    <xs:all>
                                        <xs:element name="Id" minOccurs="1" maxOccurs="1">
                                            <xs:simpleType>
                                                <xs:restriction base="xs:string">
                                                    <xs:maxLength value="64"/>
                                                </xs:restriction>
                                            </xs:simpleType>
                                        </xs:element>
                                        <xs:element name="Class" minOccurs="1" maxOccurs="1">
                                            <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>
```

2.2.1.11.1.1 Class

The <Class> element specifies the content class of the folder to be monitored. The possible content classes are *Email*, *Calendar*, *Contacts*, *Tasks*, and *Notes*<22>.

Parent elements	Child elements	Data type	Number allowed
<Folder> (request only)	None	**String**	1 (required)

Copyright © 2008 Microsoft Corporation.

2.2.1.11.1.2 Folder

The <Folder> element contains the <Id> and <Class> elements in the **Ping** command request, which identifies the folder and folder type to be monitored by the client. The <Folder> element is also returned by the server with the **Status** element, where the element identifies the folder that is being described by the returned status code.

Parent elements	Child elements	Data type	Number allowed
<Folders>	<Id> (request only) <Class> (request only) None (response only)	**Container** (request only) **String** (response only)	1...N (optional)

2.2.1.11.1.3 Folders

The <Folders> element serves as a container for the <Folder> element.

Parent elements	Child elements	Data type	Number allowed
<Ping>	<Folder>	**Container**	0...1 (optional)

2.2.1.11.1.4 HeartbeatInterval

The <HeartbeatInterval> element specifies the length of time, in seconds, that the server SHOULD wait before notifying the client of changes in a folder on the server. 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.

Parent elements	Child elements	Data type	Number allowed
<Ping>	None	**Integer**	Request- 1 (Required in first request only) Response- 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.

2.2.1.11.1.5 Id

The <Id> element specifies the server ID of the folder to be monitored.

Parent elements	Child elements	Data type	Number allowed
<Folder> (request only)	None	**String** (Up to 64 characters)	1 (required)

The server ID of the folder is obtained from the <ServerId> element of a previous **FolderSync** or **FolderCreate** command.

2.2.1.11.1.6 Ping

The <Ping> element is the top-level element in the XML document. It identifies the body of the HTTP **POST** as containing a **Ping** command.

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
None	<Folders> <HeartbeatInterval> <MaxFolders> (response only) <Status> (response only)	**Container**	1 (required)

The <Ping> element can also include one or more explicit namespace attributes.

2.2.1.11.2 Response

The following code shows the XSD for the **Ping** command response.

```
<?xml version="1.0" ?>
<xs:schema xmlns:tns="Ping:" attributeFormDefault="unqualified"
elementFormDefault="qualified"
targetNamespace="Ping:" xmlns:xs="http://www.w3.org/2001/XMLSchema">
     <xs:element name="Ping">
          <xs:complexType>
               <xs:choice>
                    <xs:element minOccurs="1" name="Status" type="xs:unsignedByte" />
                    <xs:element minOccurs="0" maxOccurs="1" name="Folders">
                         <xs:complexType>
                              <xs:sequence>
                                 <xs:element minOccurs="1"  maxOccurs="unbounded"
name="Folder" type=" xs:string "/>
                              </xs:sequence>
                         </xs:complexType>
                    </xs:element>
                    <xs:element minOccurs="0" name="MaxFolders" type="xs:integer" />
                    <xs:element minOccurs="0" name="HeartbeatInterval" type="xs:integer" />
               </xs:choice>
          </xs:complexType>
     </xs:element>
</xs:schema>
```

2.2.1.11.2.1 Folder

The <Folder> element contains the <Id> and <Class> elements in the **Ping** command request, which identifies the folder and folder type to be monitored by the client. The <Folder> element is also returned by the server with the <Status> element, where the element identifies the folder that is being described by the returned status code.

Parent elements	Child elements	Data type	Number allowed
<Folders>	<Id> (request only) <Class> (request only) None (response only)	**Container** (request only) **String** (response only)	1...N (optional)

2.2.1.11.2.2 Folders

The <Folders> element serves as a container for the <Folder> element.

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
<Ping>	<Folder>	**Container**	0...1 (optional)

2.2.1.11.2.3 HeartbeatInterval

The <HeartbeatInterval> element specifies the length of time, in seconds, that the server SHOULD wait before notifying the client of changes in a folder on the server. 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.

Parent elements	Child elements	Data type	Number allowed
<Ping>	None	**Integer**	Request- 1 (Required in first request only) Response- 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.

2.2.1.11.2.4 MaxFolders

The <MaxFolders> element specifies the maximum number of folders that can be monitored.

Parent elements	Child elements	Data type	Number allowed
<Ping> (response only)	None	**Integer**	0...1 (optional)

The <MaxFolders> element is returned in a response with a status code of 6.

2.2.1.11.2.5 Ping

The <Ping> element is the top-level element in the XML document. It identifies the body of the HTTP **POST** as containing a **Ping** command.

Parent elements	Child elements	Data type	Number allowed
None	<Folders> <HeartbeatInterval> <MaxFolders> (response only) <Status> (response only)	**Container**	1 (required)

The <Ping> element can also include one or more explicit namespace attributes.

2.2.1.11.2.6 Status

The <Status> element indicates the success or failure of the **Ping** command request. 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.

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
<Ping> (response only)	None	**Integer**	1 (required)

The following table lists valid values for the element.

Value	Meaning
1	The heartbeat interval expired before any changes occurred in the folders that are being monitored. The client reissues the **Ping** command request.
2	Changes occurred in at least one of the folders that were being monitored. The response includes the folders in which these changes have occurred.
3	The client **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.
4	There is a general error in the **Ping** request that was issued by the client, which can be caused by poorly formatted WBXML.
5	The heartbeat interval that was specified by the client is outside the range that was set by the server administrator. If the specified interval was too great, the returned interval is the maximum allowed value. If the specified interval was too low, the returned interval is the minimum allowed value.
6	The **Ping** command request specified more folders to monitor for changes than is allowed by the limit that was configured by the server administrator. The response specifies the limit in the **MaxFolders** element.
7	The client specified a folder that has been moved or deleted. The client SHOULD issue a **FolderSync** request.
8	An error has occurred. The client SHOULD reissue the **Ping** request.

2.2.1.12 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].

2.2.1.13 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 e-mail messages.<23>

2.2.1.13.1 Request

The following code shows the XSD for the **ResolveRecipients** command request.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
        xmlns:tns="ResolveRecipients:"
        attributeFormDefault="unqualified"
        elementFormDefault="qualified"
        targetNamespace="ResolveRecipients:"
```


Copyright © 2008 Microsoft Corporation.

```
        xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="ResolveRecipients">
    <xs:complexType>
      <xs:choice minOccurs="1" maxOccurs="unbounded">
        <xs:element name="To" minOccurs="0" 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" maxOccurs="1">
          <xs:complexType>
            <xs:all minOccurs="1" maxOccurs="1">
              <xs:element name="CertificateRetrieval" minOccurs="0" maxOccurs="1">
                <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" maxOccurs="1">
                <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" maxOccurs="1">
                <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" maxOccurs="1">
                <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:all>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```


Copyright © 2008 Microsoft Corporation.

2.2.1.13.1.1 CertificateRetrieval

The <CertificateRetrieval> element specifies whether S/MIME certificates SHOULD be returned by the server for each resolved recipient.

Parent elements	Child elements	Data type	Number allowed
<Options> (request only)	None	**Integer**	0...1 (optional)

The following table shows valid values for the <CertificateRetrieval>.

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.

2.2.1.13.1.2 MaxAmbiguousRecipients

The <MaxAmbiguousRecipients> element limits the number of suggestions that are returned for each ambiguous recipient node in the response.

Parent elements	Child elements	Data type	Number allowed
<Options> (request only)	None	**Integer**	0...1

The value of <MaxAmbiguousRecipients> 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, can be used by the client to determine the total number of suggestions available for that recipient.

2.2.1.13.1.3 Options

The <Options> element contains the options for resolving the list of recipients.

Parent elements	Child elements	Data type	Number allowed
<ResolveRecipients> (request only)	<CertificateRetrieval><MaxCertificates><MaxAmbiguousRecipients><Availability>	**Container**	0...1 (optional)

2.2.1.13.1.4 MaxCertificates

The <MaxCertificates> element limits the total number of certificates that is returned by the server.

Parent elements	Child elements	Data type	Number allowed
<Options>	None	<Integer>	0...1

Copyright © 2008 Microsoft Corporation.

The value of <MaxCertificates> is limited to a range of 0–9999. This limit ensures that no individual recipient receives an incomplete set of certificates. For example, if the limit is reached while enumerating certificates for an **address list**, that address list won't get back any certificates and an appropriate certificate status code is returned. The client can then use the certificate count returned to determine the number of certificates that are available for that recipient node.

2.2.1.13.1.5 ResolveRecipients

The <ResolveRecipients> element is the top-level element in the XML document. It identifies the body of the HTTP **POST** as containing a **ResolveRecipients** command.

Parent elements	Child elements	Data type	Number allowed
None	<To> (request only) <Options> (request only) <Status> (response only) <Response> (response only)	**Container**	1 (required)

The <ResolveRecipients> element can also include one or more explicit namespace attributes.

2.2.1.13.1.6 To

The <To> element specifies one or more recipients to be resolved. The <To> element is also an **ambiguous name resolution (ANR)** search field.

Parent elements	Child elements	Data type	Number allowed
<ResolveRecipients> (request only) <Response> (response only)	None	**String**, limited to 256 characters.	0...1000 (optional)

The server attempts to match the <To> 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.<24>

If the <To> element specifies an ambiguous name and the <Availability> element is included in the request, the response will not include free/busy data for that user. The Availability element is only included when <To> includes a valid SMTP address or name that resolves to a unique individual on the server.

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> value 160 is returned in the response.

If the **ResolveRecipients** command request includes the <Availability> element and the <To> element specifies a distribution list, then the availability data is returned as a single string that merges the data for the individual members of the distribution list. If the distribution list contains more than 20 members, a <Status> value of 161 is returned in the response indicating that the merged free busy information of such a large distribution list is not useful. For more information about the <Status> element, see section 2.2.1.13.2.11.

Copyright © 2008 Microsoft Corporation.

2.2.1.13.1.7 Availability

The <Availability> element indicates to the server that free/busy data is being requested by the client. The <Availability> element identifies the start time and end time of the free/busy data to retrieve.<25>

Parent elements	Child elements	Data type	Number allowed
<Options> (request only) <Response> (response only)	<StartTime> (request only)<EndTime> (request only)<Status> (response only)<MergedFreeBusy> (response only)	**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> and <EndTime>. 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.

2.2.1.13.1.8 StartTime

The <StartTime> element identifies the start of the span of free/busy time requested by the client.<26>

Parent elements	Child elements	Data type	Number allowed
<Availability> (request only)	None	**DateTime**	1…1 (required)

If the <Availability> element is included in the request, the request MUST also include the <StartTime> and <EndTime> elements.

If the client sends an invalid <StartTime> value, then the server returns a <Status> value of 5 for the **ResolveRecipients** command.

2.2.1.13.1.9 EndTime

The <EndTime> element identifies the end of the span of free/busy time requested by the client.<27>

Parent elements	Child elements	Data type	Number allowed
<Availability> (request only)	None	**DateTime**	1…1 (required)

If the **Availability** element is included in the request, the request MUST also include the <StartTime> and <EndTime> elements.

If the client sends an invalid <EndTime> value, then the server returns a <Status> value of 5 for the **ResolveRecipients** command.

Copyright © 2008 Microsoft Corporation.

If the <EndTime> value specified in the request is smaller than the <StartTime> value plus 30 minutes, or the duration spanned by the <StartTime> and the <EndTime> is greater than 42 days, then the server returns a <Status> value of 5 for the **ResolveRecipients** command.

2.2.1.13.2 Response

The following code shows the XSD for the **ResolveRecipients** command response.

```
<?xml version="1.0" ?>
<xs:schema xmlns:tns="ResolveRecipients:" attributeFormDefault="unqualified"
elementFormDefault="qualified"
targetNamespace="ResolveRecipients:" xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="ResolveRecipients">
   <xs:complexType>
      <xs:choice>
         <xs:element minOccurs="1" name="Status" type="xs:unsignedByte" />
         <xs:element minOccurs="0" maxOccurs="1" name="Response">
            <xs:complexType>
            <xs:sequence>
               <xs:element minOccurs="1" maxOccurs="1" name="To" type="xs:string">
                 <xs:simpleType>
                   <xs:restriction base="xs:string">
                     <xs:maxLength value="256"/>
                   </xs:restriction>
                 </xs:simpleType>
               </xs:element>
               <xs:element minOccurs="1" maxOccurs="1" name="Status" type="xs:string"/>
               <xs:element minOccurs="1" maxOccurs="1" name="RecipientCount"
type="xs:integer"/>
               <xs:element minOccurs="1" maxOccurs="unbounded" name="Recipient"
type="xs:string">
                  <xs:complexType>
                  <xs:sequence>
                     <xs:element minOccurs="1" maxOccurs="1" name="Type"
type="xs:unsignedbyte"/>
                     <xs:element minOccurs="1" maxOccurs="1" name="DisplayName"
type="xs:string"/>
                     <xs:element minOccurs="1" maxOccurs="1" name="EmailAddress"
type="xs:string"/>
                     <xs:element minOccurs="0" maxOccurs="unbounded" name="Certificates">
                        <xs:complexType>
                        <xs:sequence>

                           <xs:element minOccurs="1" maxOccurs="1" name="Status"
type="xs:unsignedbyte"/>
                           <xs:element minOccurs="1" maxOccurs="1" name="CertificateCount"
type="xs:integer"/>
                           <xs:element minOccurs="1" maxOccurs="1" name="RecipientCount"
type="xs:integer"/>
                           <xs:element minOccurs="0" maxOccurs="unbounded" name="Certificate"
type="xs:string"/>
                           <xs:element minOccurs="0" maxOccurs="1" name="MiniCertificate"
type="xs:string"/>

                        </xs:sequence>
                        </xs:complexType>
                     </xs:element>

                  </xs:sequence>
```


Copyright © 2008 Microsoft Corporation.

```
                    </xs:complexType>
                  </xs:element>
               </xs:sequence>
               </xs:complexType>
            </xs:element>
         </xs:choice>
      </xs:complexType>
   </xs:element>
   </xs:schema>
```

2.2.1.13.2.1 Certificate

The <Certificate> element contains the Base64-encoded X509 certificate BLOB.

Parent elements	Child elements	Data type	Number allowed
<Certificates>	None	**String** (Base64-encoded)	0…N

This element is returned by the server only if the client specified a value of 2 in the <CertificateRetrieval> element in the request.

2.2.1.13.2.2 CertificateCount

The <CertificateCount> element specifies the number of valid certificates that were found for the recipient.

Parent elements	Child elements	Data type	Number allowed
<Certificates> (response only)	None	**Integer**	1 per <Certificates> parent element

If a status code of 8 is returned with the <Certificates> element, the <CertificateCount> element specifies the number of recipient certificates that was not returned.

2.2.1.13.2.3 Certificates

The <Certificates> element contains information about the certificates for a recipient.

Parent elements	Child elements	Data type	Number allowed
<Recipient> (response only)	<Status> <CertificateCount> <RecipientCount> <MiniCertificate> <Certificate>	**Container**	0…N

2.2.1.13.2.4 DisplayName

The <DisplayName> element contains the display name of the recipient.

Parent elements	Child elements	Data type	Number allowed
<Recipient> (response only)	None	**String**	1 per <Recipient> parent element

Copyright © 2008 Microsoft Corporation.

2.2.1.13.2.5 EmailAddress

The <EmailAddress> element contains the e-mail address, in SMTP format, of the recipient.

Parent elements	Child elements	Data type	Number allowed
<Recipient> (response only)	None	**String**	1 per <Recipient> parent element

2.2.1.13.2.6 MiniCertificate

The <MiniCertificate> element contains the Base64-encoded mini-certificate BLOB.

Parent elements	Child elements	Data type	Number allowed
<Certificates>	None	**String** (Base64-encoded)	0...1 per **Certificates** parent element

This element is returned only if the client specified a value of 3 in the <CertificateRetrieval> element in the request.

2.2.1.13.2.7 Recipient

The <Recipient> element represents a single recipient that has been resolved.

Parent elements	Child elements	Data type	Number allowed
<Response>	<Type> <DisplayName> <EmailAddress> <Certificates>	**Container**	0...N

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 of <Recipient> if the client requested certificates to be returned in the response.

2.2.1.13.2.8 RecipientCount

The <RecipientCount> element specifies the number of recipients that are returned in the **ResolveRecipients** command response or the count of members belonging to a distribution list.

Parent elements	Child elements	Data type	Number allowed
<Response> <Certificates>	None	**Integer**	1...1 (required)

Copyright © 2008 Microsoft Corporation.

As a child of the <Response> element, the recipient count specifies the number of recipients that are returned in the **ResolveRecipients** command response. As a child of the <Certificates> element, the <RecipientCount> specifies the number of members belonging to a distribution list.

When returned in the <Certificates> element, the <RecipientCount> can be used to determine whether all recipients belonging to a distribution list have valid certificates by comparing values of the <CertificateCount> and <RecipientCount> elements.

2.2.1.13.2.9 ResolveRecipients

The <ResolveRecipients> element is the top-level element in the XML document. It identifies the body of the HTTP **POST** as containing a **ResolveRecipients** command.

Parent elements	Child elements	Data type	Number allowed
None	<To> (request only) <Options> (request only) <Status> (request only) <Response> (response only)	**Container**	1 (required)

The <ResolveRecipients> element can also include one or more explicit namespace attributes.

2.2.1.13.2.10 Response

The <Response> element contains information as to whether the recipient was resolved; if the recipient was resolved, the element also contains the type of recipient, the e-mail address that the recipient resolved to, and, optionally, the S/MIME certificate for the recipient.

Parent elements	Child elements	Data type	Number allowed
<ResolveRecipients>	<To> <Status> <RecipientCount> <Recipient>	**Container**	0…1 (optional)

2.2.1.13.2.11 Status

The <Status> element provides the status of the response. The meaning of the returned status code depends on whether the <Status> element was returned as a child of the <ResolveRecipients> element, the <Response> element, or the <Certificates> element.

Parent elements	Child elements	Data type	Number allowed
<ResolveRecipients> (response only) <Response> (response only) <Certificates> (response only) <Availability> (response only)	None	**Integer**	1 per <ResolveRecipients>, <Certificates>, <Availablity> parent element

Copyright © 2008 Microsoft Corporation.

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.

The following table shows valid values for the <Status> element when it is returned as a child of the <Response> element.

Value	Meaning
1	The recipient was resolved successfully. For more details about the <Recipient> element, see section 2.2.1.13.2.7.
2	The recipient was found to be ambiguous. The returned list of recipients are suggestions. No certificate nodes were 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. No certificate nodes were returned.
4	The recipient did not resolve to any contact or **GAL** entry.

The following table shows valid values for the <Status> element when it is returned as a child 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.

The following table shows valid values for the <Status> element when it is returned as a child of the <Availability> element.<28>

Value	Meaning
1	Free/busy data was successfully retrieved for a given recipient. This value does not indicate that the response is complete.
160	There were more than 100 recipients identified by the **To** elements in the **ResolveRecipient** request.
161	The distribution list 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.

Copyright © 2008 Microsoft Corporation.

Value	Meaning
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.

2.2.1.13.2.12 To

The <To> element specifies a recipient to be resolved and is an ANR search field.

Parent elements	Child elements	Data type	Number allowed
<ResolveRecipients> (request only) <Response> (response only)	None	**String**, limited to 256 characters.	1...1 (required)

The server attempts to match the <To> value specified in a request to common directory service user attributes, and then return the matches. The <To> element(s) that are returned in the response correspond directly to the <To> element(s) that are specified in the request.<29>

If the request message includes the <Availability> element and includes a <To> element for an ambiguous user, the response does not include a <MergedFreeBusy> element 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.

2.2.1.13.2.13 Type

The <Type> element indicates the type of recipient, either a contact entry (2) or a GAL entry (1).

Parent elements	Child elements	Data type	Number allowed
<Recipient> (response only)	None	Unsigned byte	1...1 (required)

2.2.1.13.2.14 Availability

The <Availability> element 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.<30>

Parent elements	Child elements	Data type	Number allowed
<Options> (request only) <Response> (response only)	<StartTime> (request only)<EndTime> (request only)<Status> (response only)<MergedFreeBusy> (response only)	**Container**	0...1 (optional)

2.2.1.13.2.15 MergedFreeBusy

The <MergedFreeBusy> element contains a string that identifies the free/busy information for the users or distribution list identified in the request.<31>

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
<Availability> (response only)	None	**String**	0...1 (optional)

The <MergedFreeBusy> string has a maximum length of 32KB. To retrieve more than 32KB of availability data, the client MUST issue a new request with the appropriate start time and end time.

Each digit in the <MergedFreeBusy> string indicates the free/busy status for the user or distribution list for every 30 minute interval. The following table lists the valid values:

Digit	Availability
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 have 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 <StartTime> and <EndTime> by 30 minutes, and rounding the result up to the next integer. If a zero digit value is calculated, the result is not rounded up to 1.

The <MergedFreeBusy> string is populated from the <StartTime> 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 may need to add or remove time intervals from the <MergedFreeBusy> string, as there are days that have more or less than 24 hours.

Copyright © 2008 Microsoft Corporation.

If the **Availability** element is included in the response, the response MUST also include the <Status> element. The <MergedFreeBusy> element is also included if the <Status> value indicates success.

2.2.1.14 Search

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

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 can 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 ANR in the global catalog server—for example, e-mail **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 device 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 ANR-indexed GAL text properties, you can also search by e-mail address, company name, and so on.

The ANR system indexes the following properties:

- Display name
- Alias
- FirstName
- LastName
- EmailAddress

The list above contains the currently supported properties.

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 device could have to sort the results to achieve a meaningful list. For example, a search for "123" might return all GAL entries that have mailing addresses or e-mail addresses that begin with 123.

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 2.2.1.14.1.4. The **Search** command does not return more than 1,000 entries.

Copyright © 2008 Microsoft Corporation.

Searching Outside the GAL<32>

Typical Usage

Essentially, search involves the following three phases:

1. The client issues a request for windowed 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** for an e-mail search result).

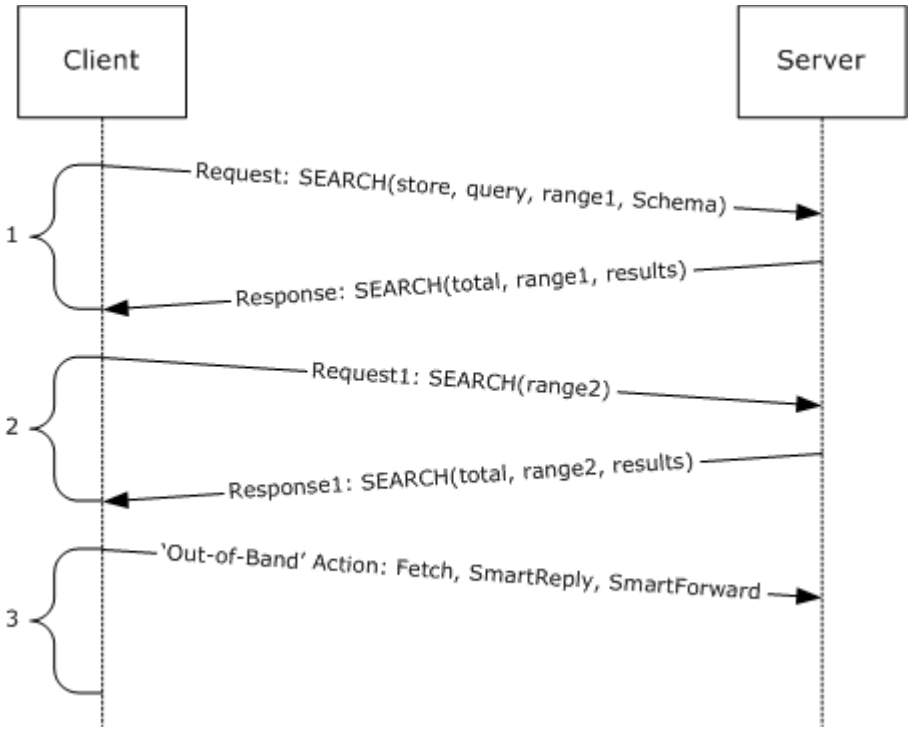

Figure 2: Search command process

2.2.1.14.1 Request

The following code shows the XSD for the **Search** command request.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
        id="Search"
        targetNamespace="Search:"
        xmlns:calendar="Calendar:"
        xmlns:contacts2="Contacts2:"
        xmlns:contacts="Contacts:"
```


Copyright © 2008 Microsoft Corporation.

```
        xmlns:email="Email:"
        xmlns:mstns="Search:"
        xmlns="Search:"
        xmlns:airSync="AirSync:"
        xmlns:airsyncbase="AirSyncBase:"
        xmlns:documentLibrary="DocumentLibrary:"
        xmlns:xs="http://www.w3.org/2001/XMLSchema"
        attributeFormDefault="qualified"
        elementFormDefault="qualified">

    <xs:import namespace="DocumentLibrary:"/>
    <xs:import namespace="AirSync:"/>
    <xs:import namespace="AirSyncBase:"/>
    <xs:import namespace="Email:"/>
    <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:complexType name="EmptyTag" />
    <xs:complexType name="queryType" mixed="true">
        <xs:sequence>
            <xs:choice maxOccurs="unbounded">
                <xs:element name="And" type="queryType"/>
                <xs:element name="Or" type="queryType"/>
                <xs:element name="FreeText" type="xs:string" />
                <xs:element ref="airSync:Class" />
                <xs:element ref="airSync:CollectionId" />
                <xs:element name="ConversationId" type="xs:string" minOccurs="0"
maxOccurs="1"/>
                <xs:element name="EqualTo" minOccurs="0">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element ref="documentLibrary:LinkId" minOccurs="1"
maxOccurs="1"/>
                            <xs:element minOccurs="1" name="Value">
                                <xs:simpleType>
                                    <xs:restriction base="xs:string">
                                        <xs:maxLength value="1024"/>
                                    </xs:restriction>
                                </xs:simpleType>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="GreaterThan">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element ref="email:DateReceived" minOccurs="1"
maxOccurs="1"/>
                            <xs:element minOccurs="1" name="Value">
                                <xs:simpleType>
                                    <xs:restriction base="xs:string">
                                        <xs:maxLength value="1024"/>
                                    </xs:restriction>
                                </xs:simpleType>
```


Copyright © 2008 Microsoft Corporation.

```
                        </xs:element>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="LessThan">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element ref="email:DateReceived" minOccurs="1"
maxOccurs="1"/>
                        <xs:element minOccurs="1" name="Value">
                            <xs:simpleType>
                                <xs:restriction base="xs:string">
                                    <xs:maxLength value="1024"/>
                                </xs:restriction>
                            </xs:simpleType>
                        </xs:element>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:choice>
    </xs:sequence>
</xs:complexType>
<xs:element name="Search">
    <xs:complexType>
        <xs:choice maxOccurs="unbounded">
            <xs:element name="Store">
                <xs:complexType>
                    <xs:all>
                        <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" minOccurs="1" type="queryType" />
                        <xs:element name="Options" minOccurs="0" maxOccurs="1">
                            <!-- Must differentiate between document library and Mailbox
options...!-->
                            <xs:complexType>
                                <xs:choice maxOccurs="unbounded">
                                    <xs:element ref="airSync:MIMESupport" minOccurs="0"
maxOccurs="1" />
                                    <xs:element ref="airsyncbase:BodyPreference"
minOccurs="0" maxOccurs="unbounded" />
                                    <xs:element name="Schema">
                                        <xs:complexType>
                                            <xs:choice maxOccurs="unbounded">
                                                <xs:element ref="airSync:Class" />
                                            </xs:choice>
                                        </xs:complexType>
                                    </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>
```


Copyright © 2008 Microsoft Corporation.

```
                                        </xs:simpleType>
                                    </xs:element>
                                    <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="EmptyTag" />
                                    <xs:element name="RebuildResults" type="EmptyTag" />
                                </xs:choice>
                            </xs:complexType>
                        </xs:element>
                    </xs:all>
                </xs:complexType>
            </xs:element>
        </xs:choice>
    </xs:complexType>
</xs:element>
</xs:schema>
```

2.2.1.14.1.1 Name

The <Name> element in the **Search** command request specifies which store to search.

Parent elements	Child elements	Data type	Number allowed
<Store>	None	**String**	1...1 (required)

The following values are also valid:

- Mailbox—The client specifies"Mailbox" when it intends to search the **message database**.
- 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.

2.2.1.14.1.2 Query

The <Query> element specifies the keywords to use for matching the entries in the store that is being searched.

Parent elements	Child elements	Data type	Number allowed
<Store>	<And> <Or>	**Container**	1...1 (Required, request only)

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
	<FreeText> <Class> <CollectionId> <ConversationId> <EqualTo> <GreaterThan> <LessThan>		

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 ANR-indexed text properties in the GAL are compared with the <Query> element value. Search comparisons are performed by using case-insensitive matching.

For a Windows SharePoint Services document library search, the [MS-ASCMD] 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 e-mail folders, including Draft, Inbox and subfolders, Outbox, and Sent Items
- The basic keyword query can be composed of the following:
  - The basic operators <And> and <Or><33>
  - Date/Time filter specified by using the *GreaterThan* and *LessThan* operators
  - <FreeText> elements that contain keywords

The basic keyword query is executed against all indexed properties.<34>

2.2.1.14.1.3 Options

The <Options> element is a container for search options.

Parent elements	Child elements	Data type	Number allowed
<Store>	<airsyncbase:BodyPreference> (request only) <airsync:MIMESupport> (request only) <Schema> <Range> <DeepTraversal>	**Container**	0…1 (Optional, Request only)

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
	<RebuildResults> <UserName> <Password>		

The <UserName> and <Password> can only be sent in the request after receiving a status 14 (see section 2.2.1.14.2.7 for more details). The server requires these credentials to access the requested resources. The device MUST only send these over a secure or trusted connection, and only in response to a status 14. <UserName> and <Password> are defined as strings consisting of at most 100 characters.

The supported options vary according to the store that is being searched. The following table lists the valid options for each store.

Options	Store
<Range> <UserName> <Password>	GAL
<Range> <Schema> <DeepTraversal> <RebuildResults> <Airsyncbase:BodyPreference>	Mailbox
<Range> <UserName> <Password>	Document Library

2.2.1.14.1.4 Range

The <Range> element is used in both the request and response XML documents. In the request XML, the <Range> element specifies the maximum number of matching entries to return. In the response XML, the <Range> element specifies the number of matching entries that are being returned.

Parent elements	Child elements	Data type	Number allowed
<Options> (request) <Store> (response)	None	Zero-based range in the form m-n	0...1 (optional)

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.

Copyright © 2008 Microsoft Corporation.

If the request does not include a <Range> element, the default range of 0–99 is assumed and a maximum of 100 items could be returned.

In the request XML, the <Range> element value specifies the maximum number of entries to be returned to the client.

In the response XML, the <Range> element value specifies the actual number of entries that are returned in the response. The <Total> element in the response XML indicates the total number of entries that matched the <Query> element value.

Search results 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.

2.2.1.14.1.5 Schema

The <Schema> element is a container for elements that specify the class of the item to search for.

Parent elements	Child elements	Data type	Number allowed
<Options>	<airsync:Class>	Container	0…1 (optional)

2.2.1.14.1.6 MIMESupport

The <MIMESupport> element is included in the <Options> element of a client **Search** command request to enable MIME support for e-mail items that are sent from the server to the client.

Parent elements	Child elements	Data type	Number allowed
<Options> (request only)	None	**Integer**	0...1 (optional)

The following table lists the valid values for the element.

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 **Search** response can include the S/MIME BLOB of a signed/encrypted message.

The **Search** request can include the following in the <Options> element:

- The <MIMESupport> element to tell the server to return MIME for S/MIME-only/All/None messages.
- The <BodyPreference> element with its child element, <Type>, 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 <Body> element, which is a child of the <Properties> element. The <Body> element is a complex element and MUST contain the following child nodes in an S/MIME search response:

Copyright © 2008 Microsoft Corporation.

- The <Type> element with a value of 4 to inform the device that the data is a MIME BLOB.
- The <EstimatedDataSize> element to specify the rough total size of the data.
- The <Truncated> element to indicate whether the MIME BLOB is truncated.
- The <Data> element that contains the full MIME BLOB.

For more details about the <Body> element or the <BodyPreference> element, see [MS-ASAIRS] section 2.2.1.3 or 2.2.1.4, respectively.

Request

```
<?xml version="1.0" encoding="utf-8"?>
<Search xmlns="Search:" xmlns:A="AirSyncBase:" xmlns:B="AirSync:" xmlns:C="Email:">
   <Store>
      <Name>Mailbox</Name>
      <Query>
         <And>
            <B:Class>Email</B:Class>
               <FreeText>text</FreeText>
         </And>
      </Query>
      <Options>
         <RebuildResults/>
         <DeepTraversal/>
         <Range>0-999</Range>
         <A:BodyPreference>
            <A:Type>4</A:Type>
            <A:TruncationSize>1024</A:TruncationSize>
<A:Preview>120</A:Preview>
         </A:BodyPreference>
         <B:MIMESupport>2</B:MIMESupport>
      </Options>
   </Store>
</Search>
```

Response

```
<Search xmlns:A0="AirSync:" xmlns:A1="POOMCONTACTS:" xmlns:A2="POOMMAIL:"
xmlns:A3="AirNotify:" xmlns:A4="POOMCAL:" xmlns:A5="Move:" xmlns:A6="GetItemEstimate:"
xmlns:A7="FolderHierarchy:" xmlns:A8="MeetingResponse:" xmlns:A9="POOMTASKS:"
xmlns:A10="ResolveRecipients:" xmlns:A11="ValidateCert:" xmlns:A12="POOMCONTACTS2:"
xmlns:A13="Ping:" xmlns:A14="Provision:" xmlns:A16="Gal:" xmlns:A17="AirSyncBase:"
xmlns:A18="Settings:" xmlns:A19="DocumentLibrary:" xmlns:A20="ItemOperations:"
xmlns="Search:">
   <Status>1</Status>
   <Response>
      <Store>
         <Status>1</Status>
         <Result>
            <A0:Class>Email</A0:Class>

<LongId>RgAAAAAty%2f%2b4QxHTJOZnIR0P9qkBwA6pk60fqkEQbWH4Wm%2bnjh7AJKAUQo6AAA6pk60fqkEQbWH4Wm
%2bnjh7AJKAURrEAAAJ</LongId>
            <A0:CollectionId>6</A0:CollectionId>
            <Properties>
```


Copyright © 2008 Microsoft Corporation.

```
                <A2:To>"NSyncUser1" &lt;NSyncUser1@contoso.com></A2:To>
                <A2:From>"NSyncUser1" &lt;NSyncUser1@contoso.com></A2:From>
                <A2:Subject>Subject</A2:Subject>
                <A2:DateReceived>2007-04-02T19:20:32.000Z</A2:DateReceived>
                <A2:DisplayTo>NSyncUser1</A2:DisplayTo>
                <A2:Read>1</A2:Read>
                <A17:Body>
                   <A17:Type>4</A17:Type>
<A17:Preview>The beginning of this message</A17:Preview>
                   <A17:EstimatedDataSize>2288</A17:EstimatedDataSize>
                   <A17:Truncated>1</A17:Truncated>
                   <A17: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: NSyncUser1 &lt;NSyncUser1@contoso.com> To: NSyncUser1 &lt;NSyncUser1@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
Body1234567890123456789012345678901234567890123456789012345678901234567890123456=
2345678901234567890123456789012345678901234567890123456789012345678901234567890123456=
7890123456789012345678901234567890123456789012345678901234567890123456789012345678901=
2345678901234567890123456789012345678901234567890123456789012345678901234567890123456789</A17:Data>
                </A17:Body>
                <A2:MessageClass>IPM.Note</A2:MessageClass>
                <A2:InternetCPID>28591</A2:InternetCPID>
                <A2:Flag/>
                <A2:ContentClass>urn:content-classes:message</A2:ContentClass>
                <A17:NativeBodyType>1</A17:NativeBodyType>
             </Properties>
          </Result>
          <Range>0-0</Range>
          <Total>1</Total>
       </Store>
    </Response>
</Search>
```


2.2.1.14.1.7 Search

The <Search> element is the top-level element in the XML document for the **Search** command. The element identifies the body of the HTTP **POST** as containing a **Search** command.

Parent elements	Child elements	Data type	Number allowed
None	<Store> (request only) <Status> (response only) <Response> (response only)	**Container**	1…1 (required)

2.2.1.14.1.8 Store

In the **Search** command request XML, the <Store> element is a container for elements that specify the location, string, and options for the search. In the **Search** command response XML, the <Store> element contains the <Status>, <Result>, <Range>, and <Total> elements that contain the returned mailbox entries.

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
<Search> (request) <Response> (response)	<Name> (request only) <Query> (request only) <Options> (request only) <Status> (response only) <Result> (response only) <Range> (response only) <Total> (response only)	**Container**	1...1 (required)

2.2.1.14.1.9 And

The <And> element is a container that specifies items on which to perform an AND operation.

Parent elements	Child elements	Data type	Number allowed
<Query> <>	<And> <Or> <FreeText> <Class> <CollectionId> <GreaterThan> <LessThan>	**Container**	0...N (Optional, request only)

The server performs an AND operation on the child nodes of the <And> element.<35>

2.2.1.14.1.10 Or

The <Or> element is a container that specifies items on which to perform an OR operation.

Parent elements	Child elements	Data type	Number allowed
<Query> <And> <Or>	<And> <Or> <FreeText> <Class> <CollectionId> <GreaterThan> <LessThan>	**Container**	0...1 (Optional, request only)

The server performs an OR operation on the child nodes of the <Or> element.<36>

2.2.1.14.1.11 Class

The <Class> element specifies the item classes that the client wants returned for a given collection.

Parent elements	Child elements	Data type	Number allowed
<Query>	None	**String**	0...4

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
<And> <Or> <Schema>			

The **Search** request can specify multiple <Class> elements, each of which is observed by the server when it sends the **Search** response to the client.<37>

If one or more <Class> elements are not specified, the server will return all supported content classes.<38>

2.2.1.14.1.12 DeepTraversal

The <DeepTraversal> element indicates that the client wants the server to search all subfolders for the folders that are specified in the query.

Parent elements	Child elements	Data type	Number allowed
<Options>	None	Empty tag	0...1 (optional)

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

2.2.1.14.1.13 EqualTo

The <EqualTo> element is a container that specifies a property and a value that are compared for equality during a search.

Parent elements	Child elements	Data type	Number allowed
<Query> <>	<LinkId> <Value>	**Container**	0...1 (optional, Request only)

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 link ID.

2.2.1.14.1.14 GreaterThan

The <GreaterThan> element is a container that specifies a property and a value that are compared for a "greater than" **condition** during a search.

Parent elements	Child elements	Data type	Number allowed
<Query> <And> <Or>	<DateReceived> <Value>	**Container**	0...1 (optional, Request only)

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.<39>The <DateRecieved> element MUST be present before the <Value> element.

Copyright © 2008 Microsoft Corporation.

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.

2.2.1.14.1.15 LessThan

The <LessThan> element is a container that specifies a property and a value that are compared for a "less than" condition during a search.

Parent elements	Child elements	Data type	Number allowed
<Query> <And> <Or>	<DateReceived> <Value>	**Container**	0...1 (Optional, request only)

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.<40>The <DateRecieved> 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.

2.2.1.14.1.16 Value

The <Value> element specifies the value that is to be used in a comparison.

Parent elements	Child elements	Data type	Number allowed
<EqualTo> <GreaterThan> <LessThan>	None	<String> (1,024 bytes maximum length)	0...1 (Optional, request only)

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.

2.2.1.14.1.17 FreeText

The <FreeText> element specifies a string for which to search.

Parent elements	Child elements	Data type	Number allowed
<Query> <And> <Or>	None	**String**	0...1 (Optional, request only)

The <FreeText> element is used together with the <And> and <Or> elements to build the query.<41>

2.2.1.14.1.18 CollectionId

The <CollectionId> element specifies the folder in which to search.<42>

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
<Query> <And> <Or>	None	**String**	0...N (optional, Request only)

Multiple folders can be specified by including multiple <CollectionId> elements.

If the <DeepTraversal> element is present, it applies to all folders under each <CollectionId>.

2.2.1.14.1.19 ConversationId

The <ConversationId> element specifies the conversation for which to search. <43><44> The value is a **GUID**. For details, see [MS-ASCON].

Parent elements	Child elements	Data type	Number allowed
<Query> <And> <Or>	None	**String**	0...1 (optional)

2.2.1.14.1.20 RebuildResults

The <RebuildResults> element is used within the **Search** request to force the server to rebuild the search folder that corresponds to a given query.

Parent elements	Child elements	Data type	Number allowed
<Options> (request)	None	Empty tag	0...1 (optional)

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.
- Enough time (order of days) has passed since the folder was first accessed (see three sentences down)

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.

If a long period of time has elapsed since a given query was issued, the client SHOULD send a new **Search** request with the given query and include the <RebuildResults> option to ensure accurate results for that query.

Copyright © 2008 Microsoft Corporation.

2.2.1.14.1.21 LinkId

The <LinkId> element specifies a URI that identifies a resource.

Parent elements	Child elements	Data type	Number allowed
<EqualTo>	None	**URI**	1…1 (required)

The link ID is a URI that is assigned by the server to certain resources, such as Windows SharePoint Services or **UNC** documents. The client MUST store the link ID that is associated with the items that are retrieved by using the **Search** command if it wants to act upon them later.

2.2.1.14.1.22 DateReceived

The <DateReceived> element specifies the date that a mailbox item was received.

Parent elements	Child elements	Data type	Number allowed
<GreaterThan> <LessThan>	None	**DateTime**	1…1 (required)

2.2.1.14.2 Response

2.2.1.14.2.1 LongId

The <LongId> element specifies a unique identifier that is assigned by the server to each result set that is being returned in the **Search** response.

Parent elements	Child elements	Data type	Number allowed
<Result> (response only)	None	**String** (up to 256 characters)	0…1 (optional)

The value of the <LongId> element can be used in the <LongId> parameter of the **ItemOperations**, **SmartReply**, **SmartForward**, or **MeetingResponse** command requests to reference the result set.

The client MUST store the value of <LongId> as an opaque string of up to 256 characters.

2.2.1.14.2.2 Properties

In a **Search** response, the <Properties> element encapsulates the properties for each search result.

Parent elements	Child elements	Data type	Number allowed
Schema (request only) **Result** (response only)	<airsyncbase:Body> (response only) 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].	**Container**	1…1 (Required, response only)

Copyright © 2008 Microsoft Corporation.

The **Search** command response <Properties> element is a container for properties that apply to an individual entry that matches the <Query> element 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.

2.2.1.14.2.3 Range

The **Search** command <Range> element is used in both the request and response XML documents. In the request XML, the <Range> element specifies the range of matching entries to return. In the response XML, the <Range> element specifies the number of matching entries that are being returned.

Parent elements	Child elements	Data type	Number allowed
<Options> (request) <Store> (response)	None	Zero-based range in the form m-n	0...1 (optional)

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 numeric value, a hyphen, and another numeric value: "m-n." The m and n indicates the lowest and 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.

In the request XML, the <Range> element value specifies the range of entries to be returned to the client.

In the response XML, the <Range> element value specifies the actual number of entries that are returned in the response. The <Total> element in the response XML indicates the total number of entries that matched the <Query> element value.

Search results 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.

2.2.1.14.2.4 Response

The <Response> element is a container for the results that are returned from the server.

Parent elements	Child elements	Data type	Number allowed
<Search> (response only)	<Store> (response only)	**Container**	1...1 (required)

2.2.1.14.2.5 Result

The **Search** command response <Result> element is a container for an individual matching mailbox item.

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
<Store> <Response>	<Properties> <Class> <LongId> <ParentId>	**Container**	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 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 <LinkId> is pointing. The client can choose to ignore this entry if it does not require it.
- If the <LinkId> 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 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 <LinkId> 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.

2.2.1.14.2.6 Search

The <Search> element is the top-level element in the XML document for the **Search** command. The element identifies the body of the HTTP **POST** as containing a **Search** command.

Parent elements	Child elements	Data type	Number allowed
None	<Store> (request only) <Response> (response only) <Status> (response only)	**Container**	1...1 (required)

2.2.1.14.2.7 Status

The **Search** command response <Status> element indicates whether the server encountered an error while it was processing the search query.

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
<Store> (response only) <Response> (response only)	None	**Integer**	1...1 (required)

The following table specifies valid values for the <Status> element as a child of the <Search> node in the search response.

Value	Meaning
1	Success
3	Server error

The following table specifies valid values for the <Status> element as a child of the <Store> element in the search response.

Value	Meaning
1	Success.
2	Protocol violation/XML validation error.
3	Server error.
4	Bad Link
5	Access Denied
6	Not found
7	Connection Failed (try again)
8	The search query is too complex.
10	Search timed out.
11	Bad <CollectionId> (the client MUST perform a **FolderSync**).
12	Server reached the end of the range that is retrievable by synchronization.
13	Access Blocked (policy restriction)
14	Credentials Required to Continue

The <Status> element value indicates only that the **Search** command was processed correctly. It does not indicate whether any matches were found. The **Total** and **Range** response XML elements indicate how many matches were found and returned, respectively.

The response will contain multiple <Status> elements. The <Status> element indicates the processing status of the overall **Search** command when the <Search> element is the immediate parent of the <Status> element. When the immediate parent of the <Status> element is the <Store> element, that <Status> element indicates the processing status for only that store. This structure was chosen to enable possible future expansion of the command to searching multiple locations, address lists, and contacts folders.

Copyright © 2008 Microsoft Corporation.

2.2.1.14.2.8 Store

In the **Search** command request XML, the <Store> element is a container for elements that specify the location, string, and options for the search. In the **Search** command response XML, the <Store> element contains the <Status>, <Result>, <Range>, and <Total> elements that contain the returned mailbox entries.

Parent elements	Child elements	Data type	Number allowed
<Search> (request) <Response> (response)	<Name> (request) <Query> (request only) <Options> (request only) <Status> (response only) <Result> (response only) <Range> (response only) <Total> (response only)	**Container**	1...1 (required)

2.2.1.14.2.9 Total

The **Search** command response XML element <Total> indicates the total number of mailbox entries that matched the search <Query> element value.

Parent elements	Child elements	Data type	Number allowed
<Store>	None	**Integer**	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 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 device 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 device can then enable the user to retrieve the full results.

2.2.1.15 SendMail

The **SendMail** command is used by clients to send MIME-formatted e-mail messages to the server.

Messages SHOULD NOT be saved directly to the local Sent Items folder by the client; instead, clients SHOULD use the <SaveInSentItems> element to automatically have the messages saved 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. 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.

Copyright © 2008 Microsoft Corporation.

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

2.2.1.15.1 Request

The following code shows the XSD for the **SendMail** command request.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
        xmlns:tns="ComposeMail:"
        attributeFormDefault="unqualified"
        elementFormDefault="qualified"
        targetNamespace="ComposeMail:"
        xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:element name="SendMail">
        <xs:complexType>
            <xs:all>
                <xs:element name="ClientId" minOccurs="1" maxOccurs="1">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:minLength value="1"/>
                            <xs:maxLength value="40"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="SaveInSentItems" minOccurs="0" maxOccurs="1" />
                <xs:element name="Mime" type="xs:string" minOccurs="1" maxOccurs="1" />
            </xs:all>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

2.2.1.15.1.1 SendMail

The <SendMail> element is the top-level element in the XML stream. It indicates that the body of the HTTP **POST** contains a **SendMail** command.

Parent elements	Child elements	Data type	Number allowed
None	<ClientId> (request only) <SaveInSentItems> (request only) <Mime> (request only) <Status> (response only)	**Container**	1…1 (required)

2.2.1.15.1.2 ClientId

The <ClientId> element specifies the client's unique message ID (MID).

Parent elements	Child elements	Data type	Number allowed
<SendMail> (request only)	None	**String** (Up to 40 characters)	1…1 (required)

Copyright © 2008 Microsoft Corporation.

The <ClientId> MUST be unique for each message, as the server will use the <ClientId> to identify duplicate messages. The <ClientId> can be a simple counter incremented for each new message.

2.2.1.15.1.3 SaveInSentItems

The <SaveInSentItems> element specifies whether a copy of the message should 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.

Parent elements	Child elements	Data type	Number allowed
<SendMail> (request only)	None	Empty tag	0...1 (optional)

2.2.1.15.1.4 Mime

The <Mime> element contains the MIME-encoded message.

Parent elements	Child elements	Data type	Number allowed
<SendMail> (request only)	None	**Byte array**	1...1 (required)

The <Mime> content is transferred as an opaque BLOB within the WBXML tags.

2.2.1.15.2 Response

2.2.1.15.2.1 SendMail

The <SendMail> element is the top-level element in the XML stream. It indicates that the body of the HTTP **POST** contains a **SendMail** command.

Parent elements	Child elements	Data type	Number allowed
None	<ClientId> (request only) <Mime> (request only) <Status> (response only)	**Container**	1...1 (required)

2.2.1.15.2.2 Status

The <Status> element indicates the success or failure of a **SendMail** command request.

Parent elements	Child elements	Data type	Number allowed
<SendMail> (response only)	None	**Integer**	0...1 (optional)

If the command succeeds, the <Status> element is returned with code 1. If the command fails, the <Status> element contains a code that indicates the type of failure.

Valid <Status> values are listed in 2.2.2.14.

2.2.1.16 Settings

The **Settings** command supports get and set operations on global properties.<45>

Copyright © 2008 Microsoft Corporation.

The <Get> and <Set> 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:

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

The <PropertyName> is a named property (that is, the actual name of the property). The **Settings** command can be used on the following named properties:

- OOF
- Device Password
- Device Information
- User Information

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 many <PropertyName> nodes. Each property MUST be processed in order. There can be cases in which one property call affects another property call or the same property is in the **Settings** request more than once. The responses will come back in the same order in which they were requested.

Each response has a global status response, which is mainly for protocol errors, followed by the property responses.

The <Status> node MUST return Success if **Settings** is returning property responses. If the command was not successful, the processing of the request cannot begin, no property responses are returned, and <Status> MUST indicate a protocol error.

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

The Return data is specified by the individual properties.

2.2.1.16.1 Request

The following code shows the XSD for the **Settings** command request.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
        xmlns:tns="Settings:"
```


Copyright © 2008 Microsoft Corporation.

```
        attributeFormDefault="unqualified"
        elementFormDefault="qualified"
        targetNamespace="Settings:"
        xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:simpleType name="EmptyStringType">
        <xs:restriction base="xs:string">
            <xs:maxLength value="0"/>
        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="NonEmptyStringType">
        <xs:restriction base="xs:string">
            <xs:minLength value="1"/>
        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="DeviceInformationStringType">
        <xs:restriction base="xs:string">
            <xs:maxLength value="1024"/>
        </xs:restriction>
    </xs:simpleType>

    <xs:element name="Settings">
        <xs:complexType>
            <xs:all>
                <xs:element name="Oof" minOccurs="0" maxOccurs="1">
                    <xs:complexType>
                        <xs:choice>
                            <xs:element name="Get" minOccurs="0" maxOccurs="1" >
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="BodyType"
type="tns:NonEmptyStringType" minOccurs="1" maxOccurs="1" />
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                            <xs:element name="Set">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="OofState"
type="tns:NonEmptyStringType" minOccurs="0" maxOccurs="1" />
                                        <xs:element name="StartTime"
type="tns:NonEmptyStringType" minOccurs="0" maxOccurs="1" />
                                        <xs:element name="EndTime"
type="tns:NonEmptyStringType" minOccurs="0" maxOccurs="1" />
                                        <xs:element name="OofMessage" minOccurs="0"
maxOccurs="3">
                                            <xs:complexType>
                                                <xs:all>
                                                    <xs:element name="AppliesToInternal"
type="tns:EmptyStringType" minOccurs="0" maxOccurs="1" />
                                                    <xs:element name="AppliesToExternalKnown"
type="tns:EmptyStringType" minOccurs="0" maxOccurs="1" />
                                                    <xs:element
name="AppliesToExternalUnknown" type="tns:EmptyStringType" minOccurs="0" maxOccurs="1" />
                                                    <xs:element name="Enabled"
type="tns:NonEmptyStringType" minOccurs="0" maxOccurs="1" />
                                                    <xs:element name="ReplyMessage"
type="xs:string" minOccurs="0" maxOccurs="1" />
```


Copyright © 2008 Microsoft Corporation.

```
                                                    <xs:element name="BodyType"
type="tns:NonEmptyStringType" minOccurs="0" maxOccurs="1" />
                                                        </xs:all>
                                                    </xs:complexType>
                                                </xs:element>
                                            </xs:sequence>
                                        </xs:complexType>
                                    </xs:element>
                                </xs:choice>
                            </xs:complexType>
                        </xs:element>
                        <xs:element name="DevicePassword" minOccurs="0" maxOccurs="1">
                            <xs:complexType>
                                <xs:choice>
                                    <xs:element name="Set">
                                        <xs:complexType>
                                            <xs:sequence>
                                                <xs:element name="Password" type="xs:string"
minOccurs="1" maxOccurs="1" />
                                            </xs:sequence>
                                        </xs:complexType>
                                    </xs:element>
                                </xs:choice>
                            </xs:complexType>
                        </xs:element>
                        <xs:element name="DeviceInformation" minOccurs="0" maxOccurs="1">
                            <xs:complexType>
                                <xs:choice>
                                    <xs:element name="Set">
                                        <xs:complexType>
                                            <xs:all>
                                                <xs:element name="Model"
type="tns:DeviceInformationStringType" maxOccurs="1" minOccurs="0" />
                                                <xs:element name="IMEI"
type="tns:DeviceInformationStringType" maxOccurs="1" minOccurs="0" />
                                                <xs:element name="FriendlyName"
type="tns:DeviceInformationStringType" maxOccurs="1" minOccurs="0" />
                                                <xs:element name="OS"
type="tns:DeviceInformationStringType" maxOccurs="1" minOccurs="0" />
                                                <xs:element name="OSLanguage"
type="tns:DeviceInformationStringType" maxOccurs="1" minOccurs="0" />
                                                <xs:element name="PhoneNumber"
type="tns:DeviceInformationStringType" maxOccurs="1" minOccurs="0" />
                                                <xs:element name="UserAgent"
type="tns:DeviceInformationStringType" maxOccurs="1" minOccurs="0" />
                                                <xs:element name="EnableOutboundSMS" minOccurs="0"
maxOccurs="1">
                                                    <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="tns:DeviceInformationStringType" maxOccurs="1" minOccurs="0" />
                                            </xs:all>
                                        </xs:complexType>
                                    </xs:element>
                                </xs:choice>
```


Copyright © 2008 Microsoft Corporation.

```
                    </xs:complexType>
                </xs:element>
                <xs:element name="UserInformation" minOccurs="0" maxOccurs="1">
                    <xs:complexType>
                        <xs:choice>
                            <xs:element minOccurs="1" maxOccurs="1" name="Get"
type="tns:EmptyStringType"/>
                        </xs:choice>
                    </xs:complexType>
                </xs:element>
            </xs:all>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

2.2.1.16.1.1 Settings

The <Settings> element is the top-level element in the XML document for the **Settings** command.

Parent elements	Child elements	Data type	Number allowed
None	<Oof> <DeviceInformation> <DevicePassword> <UserInformation> <Status> (response only)	**Container**	1…1 (required)

The <Settings> element encapsulates one or more named property nodes that contain actions and arguments that apply to those properties.

2.2.1.16.1.2 Oof

The <Oof> element specifies a named property node for retrieving and setting **out of office (OOF)** information.

Parent elements	Child elements	Data type	Number allowed
<Settings>	<Get> <Set> (request only) <Status> (response only)	**Container**	0…1

The **Settings** command 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

Copyright © 2008 Microsoft Corporation.

The <Get> command within the <Oof> element enables the client to retrieve OOF information from the server. The client specifies the <BodyType> to be retrieved and the server will return all OOF information and messages.

There is one <OofMessage> node per audience in an <OOF> <Get> response. If a sender group is not allowed and is disabled (an unknown external sender can be disabled by the administrator), an <OofMessage> node is not reported to the client in a <Get> response. If the sender group is allowed, but is disabled and has no reply message (specified by the <ReplyMessage> element), 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> command 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

2.2.1.16.1.3 Get

The <Get> command enables the client to retrieve information from the server for any named property that supports <Get>.

Parent elements	Data type	Number allowed
<Oof> <UserInformation>	**Container** or **Empty**	0…1 1…1

The <Get> element is an **Empty** element when it is the child of <UserInformation> in a request, meaning it has no value or data type. It is distinguished only by the presence or absence of the <Get/> tag. The <Get> element is a **Container** when it is the child of <Oof>, or included in a <UserInformation> response.

Child elements in an <Oof> request	Child elements in an <Oof> response	Child elements in a <UserInformation> request or response
<BodyType>	<OofState> <StartTime> <EndTime> <OofMessage>	<EmailAddresses> (response only)

Only the **Oof** and **User Information** named properties support <Get>.

In an <Oof> request, the client specifies the body type to be retrieved and the server will return all OOF settings and messages for that body type.

Copyright © 2008 Microsoft Corporation.

2.2.1.16.1.4 Set

The <Set> command enables the client to set information on the server for any named property that supports <Set>.

Parent elements	Data type	Number allowed
<Oof> (request only) <DeviceInformation> <DevicePassword>	**Container**	0...1 (optional)

Child elements in an <Oof> request	Child elements in an <Oof> response
<OofState> <StartTime> <EndTime> <OofMessage>	None

Child elements in a <DeviceInformation> request	Child elements in a <DeviceInformation> response
<Model> <IMEI> <FriendlyName> <OS> <OSLanguage> <PhoneNumber> <UserAgent> <EnableOutboundSMS> <MobileOperator>	<None>

Child elements in a <DevicePassword> request	Child elements in a <DevicePassword> response
<Password>	<Status>

The named properties that support <Set> are **OOF**, **Device Information**, and **Device Password**.

2.2.1.16.1.4.1 OOF Property

The <Set> command enables the client to set the following in the <OOF> property:

- OOF state
- Start time and end time, if the user wants to schedule an OOF message.

Copyright © 2008 Microsoft Corporation.

- OOF message or messages for one or more of the supported audiences

2.2.1.16.1.4.2 Device Information Property

<Set> enables the client to specify values for any of the Device Information parameters. The following statements apply to the <Set> command request implementation:

- The client MUST specify all supported Device Information parameters in the <Set> request.
- The client or server makes no assumptions about ordering. The Device Information parameters can be specified in any order.
- To delete a given Device Information value, the client sends the <Set> command with an empty element for that parameter. Active Sync will set that parameter to NULL.

2.2.1.16.1.4.3 Device Password Property

The <Set> command enables the client to set or clear the recovery password of the device.

2.2.1.16.1.5 OofState

The <OofState> element specifies the availability of the OOF property.

Parent elements	Child elements	Data type	Number allowed
<Get> (OOF response only) <Set> (OOF request only)	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 values of <OofState> match those of the Availabilities Service enumeration. <OofState> MUST be set to 2 if the <StartTime> and <EndTime> elements are present.

2.2.1.16.1.6 StartTime

The <StartTime> element is used with the <EndTime> element to specify the range of time during which the user is out of office.

Parent elements	Child elements	Data type	Number allowed
<Get> (<Oof> response only) <Set> (<Oof> request only)	None	**DateTime**	0...1 (optional)

The <StartTime> element can be present within the <Get> element of the **Settings** response for the <Oof> property, or within the <Set> element of the **Settings** request for the <Oof> property.

Copyright © 2008 Microsoft Corporation.

If <StartTime> is present, the <EndTime> element MUST be present also. Otherwise, the client will receive a protocol error.

2.2.1.16.1.7 EndTime

The <EndTime> element is used with the <StartTime> element to specify the range of time during which the user is out of office.

Parent elements	Child elements	Data type	Number allowed
<Get> (<Oof> response only) <Set> (<Oof> request only)	None	**DateTime**	0...1 (optional)

The <EndTime> element can be present within the <Get> element of the **Settings** response for the <Oof> property or within the <Set> element of the **Settings** request for the <Oof> property.

If <EndTime> is present, the <StartTime> element MUST be present also, and <OofState> MUST be set to 2. Otherwise, the client will receive a protocol error.

2.2.1.16.1.8 OofMessage

The <OofMessage> element contains a set of elements that specify the OOF message for a particular audience.

Parent elements	Child elements	Data type	Number allowed
<Get> (<Oof> response only) <Set> (<Oof> request only)	<AppliesToInternal> <AppliesToExternalKnown> <AppliesToExternalUnknown> <Enabled> <ReplyMessage> <BodyType>	**Container**	0...3

The <Oof> property supports the following three audiences for an OOF message: <46>

- 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 an 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

Copyright © 2008 Microsoft Corporation.

<AppliesToExternalUnknown> are not allowed and are disabled by the administrator but are sent by the client in the <Set> request, <Set> returns a successful <Status> value of 1 even though the user does not have access to these settings. Similarly, <AppliesToExternalKnown> and <AppliesToExternalUnknown> are returned to the client in a <Get> response even if the sender group is not allowed and is disabled.

In an OOF <Set> request, the client MUST NOT include the same AppliesTo* element in more than one <OofMessage> element.

2.2.1.16.1.9 BodyType

The <BodyType> element is a string that specifies the format of the OOF message.

Parent elements	Child elements	Data type	Number allowed
<Get> (request only) <OofMessage>	None	**String**	1...1 (required) (On <Get>) 0...1 (optional) under <OofMessage>, required if a message is set

The following are the permitted values for the <BodyType> element:

- *Text*
- **HTML**

If <BodyType> has the value **HTML**, all message strings are sent in the HTML format. If <BodyType> has the value *Text*, the message strings are sent in **plain text**. Because there is no default value, the <BodyType> node MUST be present on all <Get> operations and on any <OofMessage> where the <ReplyMessage> has been set.

2.2.1.16.1.10 AppliesToInternal

The <AppliesToInternal> element 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.)

Parent elements	Child elements	Data type	Number allowed
<OofMessage>	None	**Empty**	0...1 (Choice of <AppliesToInternal>, <AppliesToExternalKnown>, and <AppliesToExternalUnknown>)

The <AppliesToInternal> element is an **Empty** 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>—Specifies whether an OOF message is sent to this audience while the sending user is OOF.
- <ReplyMessage>—Specifies the OOF message itself.

Copyright © 2008 Microsoft Corporation.

- <BodyType>—Specifies the format of the OOF message.

The <AppliesToInternal>, <AppliesToExternalKnown>, and <AppliesToExternalUnknown> elements, each of which indicates the audience to which an OOF message pertains, are mutually exclusive.

2.2.1.16.1.11 AppliesToExternalKnown

The <AppliesToExternalKnown> element 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.)

Parent elements	Child elements	Data type	Number allowed
<OofMessage>	None	<Empty>	0...1 (Choice of <AppliesToInternal>, <AppliesToExternalKnown>, and <AppliesToExternalUnknown>)

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>—Specifies whether an OOF message is sent to this audience while the sending user is OOF.
- <ReplyMessage>—Specifies the OOF reply message.
- <BodyType>—Specifies the format of the OOF message.

The <AppliesToInternal>, <AppliesToExternalKnown>, and <AppliesToExternalUnknown> elements, each of which indicates the audience to which an OOF message pertains, are mutually exclusive.

2.2.1.16.1.12 AppliesToExternalUnknown

The <AppliesToExternalUnknown> element 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.)

Parent elements	Child elements	Data type	Number allowed
<OofMessage>	None	**Empty**	0...1 (Choice of AppliesToInternal, AppliesToExternalKnown, and AppliesToExternalUnknown)

The <AppliesToExternalUnknown> element is an **Empty** element, meaning it has no value or data type. It is distinguished only by the presence or absence of the <AppliesToExternalUnknown/> 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 unknown external users.

The following are the peer elements of the <AppliesToExternalKnown> element:

Copyright © 2008 Microsoft Corporation.

- <Enabled>—Specifies whether an OOF message is sent to this audience while the sending user is OOF.
- <ReplyMessage>—Specifies the OOF reply message.
- <BodyType>—Specifies the format of the OOF message.

The <AppliesToInternal>, <AppliesToExternalKnown>, and <AppliesToExternalUnknown> elements, each of which indicates the audience to which an OOF message pertains, are mutually exclusive.

2.2.1.16.1.13 Enabled

The <Enabled> element specifies whether an OOF message is sent to this audience while the sending user is OOF.

Parent elements	Child elements	Data type	Number allowed
<OofMessage>	None	**String**	0...1 (optional)

The <Enabled> element is used in the OOF <Get> response to retrieve the current value. The <Enabled> element is used in the OOF <Set> request to set the value.

The value of <Enabled> is 1 if an OOF message is sent while the sending user is OOF; otherwise, the value is 0.

2.2.1.16.1.14 ReplyMessage

The <ReplyMessage> element specifies the message to be shown to a particular audience when the user is OOF.

Parent elements	Child elements	Data type	Number allowed
<OofMessage>	None	**String**	0...1 (optional)

The <ReplyMessage> can be used in an OOF <Get> response to convey the requested OOF message, or in an OOF <Set> request to set the message that the client wants to send to a particular audience. In a <Set>, any <ReplyMessage> MUST also specify a <BodyType>.

The <OOF> property 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:<8>

- <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.

Copyright © 2008 Microsoft Corporation.

2.2.1.16.1.15 DeviceInformation

The <DeviceInformation> element is the container node that is used for sending the client device's properties to the server.

Parent elements	Child elements	Data type	Number allowed
<Settings>	<Set>	<Container>	0...1 (optional)

It is important to have pertinent information about a user's device for monitoring and troubleshooting. The <DeviceInformation> element is used in the **Settings** command to send 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 (see [MS-ASMS])

This information is reflected in the output to administrative tasks (for example, reporting).<47>

The device information is represented as a flat list of settings under the <DeviceInformation> node in the **Settings** command. <DeviceInformation> 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> elements because this information is write-only from the device.

2.2.1.16.1.16 Model

The <Model> element specifies a name that generally describes the device of the client.

Parent elements	Child elements	Data type	Number allowed
<Set> (<DeviceInformation> request only)	None	**String**	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.

2.2.1.16.1.17 IMEI

The <IMEI> element specifies a 15-character code that MUST uniquely identify a device.

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
<Set> (<DeviceInformation> request only)	None	**String**	0...1 (optional)

The server does not validate the IMEI format.

The device ID parameter that is currently included in the request URL is not precisely defined; licensees 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.

2.2.1.16.1.18 FriendlyName

The <FriendlyName> element specifies a name that MUST uniquely describe the client device.

Parent elements	Child elements	Data type	Number allowed
<Set> (<DeviceInformation> request only)	None	**String**	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 friendly name of the device is typically specified during partnership creation if the user cradles the device to the desktop.

2.2.1.16.1.19 OS

The <OS> element specifies the operating system of the client device.

Parent elements	Child elements	Data type	Number allowed
<Set> (<DeviceInformation> request only)	None	**String**	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 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 use the following convention:

<Operating System Product Name> <Operating System Major Version> <Operating System Minor Version>

2.2.1.16.1.20 OSLanguage

The <OSLanguage> element specifies the language that is used by the operating system of the client device.

Parent elements	Child elements	Data type	Number allowed
<Set> (<DeviceInformation> request only)	None	**String**	0...1 (optional)

Copyright © 2008 Microsoft Corporation.

Knowledge of the user's language facilitates localization if the server is required to send localizable content to the client device. The server does not validate the value of the <OSLanguage> element.

2.2.1.16.1.21 PhoneNumber

The <PhoneNumber> element specifies a unique number that identifies the client device.

Parent elements	Child elements	Data type	Number allowed
<Set> (<DeviceInformation> request only)	None	**String**	0...1 (optional)

The telephone number facilitates troubleshooting and device management by providing a well-known and unique identifier for the client device. The server does not validate the value of the <PhoneNumber> element.

2.2.1.16.1.22 UserAgent

The <UserAgent> element specifies the user agent.

Parent elements	Child elements	Data type	Number allowed
<Set> (<DeviceInformation> request only)	None	**String**	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.

2.2.1.16.1.23 DevicePassword

The <DevicePassword> element is a container node that is used to send the recovery password of the client device to the server.

Parent elements	Child elements	Data type	Number allowed
<Settings>	<Set>	**Container**	0...1 (optional)

Use the <Set> operation on the <DevicePassword> property enable the device to send or store a recovery password on the server. The recovery password is be 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.

2.2.1.16.1.24 Password

The <Password> element specifies the recovery password of the client device, which is stored by the server.

Parent elements	Child elements	Data type	Number allowed
<Set> (<DevicePassword> request only)	None	**String**	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 <Set> request with an empty <Password> element.

Copyright © 2008 Microsoft Corporation.

2.2.1.16.1.25 UserInformation

The <UserInformation> element is a container node that is used to request a list of a user's e-mail addresses from the server.

Parent elements	Child elements	Data type	Number allowed
<Settings>	<Get> (<UserInformation> request only) <Status> (<UserInformation> response only)	**Container**	0...1 (optional)

The list of a user's e-mail addresses can be useful, for example, for ensuring that the user is not included when performing a Reply to All operation to an e-mail message.

In a request, the <UserInformation> element contains the <Get> command to indicate that the server is to return all available e-mail addresses for the user.

The **Settings** command supports read-only access to the list of a user's various e-mail addresses via the <Get> command. The client is unable to write this information.

2.2.1.16.1.26 EnableOutboundSMS

The <EnableOutboundSMS> element specifies whether the server will send outbound SMS messages through the mobile device. For more details, see [MS-ASMS].<48>

Parent elements	Child elements	Data type	Number allowed
<Set> (<DeviceInformation> request only)	None	**Integer**	0...1 (optional)

If this element is set to 1, then the mobile device can be used to send outbound SMS messages; otherwise, the mobile device cannot be used to send outbound SMS messages.

2.2.1.16.1.27 MobileOperator

The <MobileOperator> element specifies name of the mobile operator to which a mobile device is connected. For more details, see [MS-ASMS].<49>

Parent elements	Child elements	Data type	Number allowed
Set (<DeviceInformation> request only)	None	**String**	0...1 (optional)

2.2.1.16.2 Response

2.2.1.16.2.1 Settings

The <Settings> element is the top-level element in the XML document for the **Settings** command.

Parent elements	Child elements	Data type	Number allowed
None	<Oof> <DeviceInformation> <DevicePassword> <UserInformation>	**Container**	1...1 (required)

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
	<Status> (response only)		

The <Settings> element encapsulates one or more named property nodes that contain actions and arguments that apply to those properties.

2.2.1.16.2.2 Status

The <Status> element contains a code that indicates the success or failure of the **Settings** command and the success or failure of actions that are associated with a specific property node (<Oof>, <DeviceInformation>, <DevicePassword>, <UserInformation>).

Parent elements (response only)	Child elements	Data type	Number allowed for <Settings> parent	Number allowed for <Oof> parent, <Set> parent, or <UserInformation> parent
<Settings> <Oof> <UserInformation> <Set> (only in a <DeviceInformation> or <DevicePassword> response)	None	**Integer**	1...1 (required)	0...1

The following table lists the valid values for the <Status> element in the context of the **Settings** command response. This is the status at the top level.

Value	Meaning
1	Success.
2	Protocol error.

The following table lists the valid values for <Status> in a **Settings** command <DeviceInformation> response.

Value	Meaning
1	Success.
2	Protocol error. The XML code is formatted incorrectly.

The following table lists the values for <Status> in a **Settings** command <DevicePassword> response.

Value	Meaning
1	Success.
2	Protocol error. The XML code is formatted incorrectly.
5	Invalid arguments. The specified password is too long.

Copyright © 2008 Microsoft Corporation.

Value	Meaning
7	Denied by policy. The administrator has disabled password recovery in this deployment.

The following table lists the values for <Status> in a **Settings** command <UserInformation> response.

Value	Meaning
1	Success.
2	Protocol error. The XML code is formatted incorrectly.

The status is specified for the **Settings** command response and for each property node (<Oof>, <DeviceInformation>, <DevicePassword>, <UserInformation>) within **Settings**.

If <OOF> nodes <AppliesToExternalKnown> or <AppliesToExternalUnknown> are not allowed and are disabled by the administrator but are sent by the client in the <Set> request, <Oof> returns a successful <Status> value of 1 even though the user does not have access to these settings.

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.

2.2.1.16.2.3 Oof

The <Oof> element specifies a named property node for retrieving and setting OOF information.

Parent elements	Child elements	Data type	Number allowed
<Settings>	<Get> <Set> (request only) <Status> (response only)	**Container**	0…1

The **Settings** command 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> command within the <Oof> element enables the client to retrieve OOF information from the server. The client specifies the <BodyType> to be retrieved and the server will return all OOF information and messages.

There is one <OofMessage> node per audience in an OOF <Get> response. If a sender group is not allowed and is disabled (an unknown external sender can be disabled by the administrator), an <OofMessage> node is not reported to the client in a <Get> response. If the sender group is allowed, but is disabled and has no Reply message (specified by the <ReplyMessage> element), an <OofMessage> node is still reported to the client.

Copyright © 2008 Microsoft Corporation.

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 MUST result in an Access Denied status code.

OOF Set Request and Response

The <Set> command 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

2.2.1.16.2.4 Get

The <Get> command enables the client to retrieve information from the server for any named property that supports <Get>.

Parent elements	Data type	Number allowed
<Oof> <UserInformation>	**Container** or **Empty**	0...1 1...1

The <Get> element is an **Empty** element when it is the child of <UserInformation> in a request, meaning it has no value or data type. It is distinguished only by the presence or absence of the <Get/> tag. The <Get> element is a **Container** when it is the child of <Oof>, or included in a <UserInformation> request.

Child elements in an <Oof> request	Child elements in an <Oof> response	Child elements in a <UserInformation> request or response
<BodyType>	<OofState> <StartTime> <EndTime> <OofMessage>	None

Only the **OOF** and **User Information** named properties support <Get>.

In an <Oof> request, the client specifies the body type to be retrieved and the server will return all OOF settings and messages for that body type.

2.2.1.16.2.5 Set

The <Set> command enables the client to set information on the server for any named property that supports <Set>.

Parent elements	Data type	Number allowed
<Oof> (request only) <DeviceInformation> <DevicePassword>	**Container**	0...1 (optional)

Copyright © 2008 Microsoft Corporation.

Child elements in an <Oof> request	Child elements in an <Oof> response
<OofState> <StartTime> <EndTime> <OofMessage>	None

Child elements in an <DeviceInformation> request	Child elements in an <DeviceInformation> response
<Model> <IMEI> <FriendlyName> <OS> <OSLanguage> <Phone Number> <UserAgent> <EnableOutboundSMS> <MobileOperator>	<None>

Child elements in an <DevicePassword> request	Child elements in an <DevicePassword> response
Password	Status

The named properties that support <Set> are **OOF**, **Device Information**, and **Device Password**.

2.2.1.16.2.5.1 OOF Property

The <Set> command enables the client to set the following in the OOF property:

- 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

2.2.1.16.2.5.2 Device Information Property

<Set> enables the client to specify values for any of the Device Information parameters. The following statements apply to the <Set> command request implementation:

- The client MUST specify all supported Device Information parameters in the <Set> request.
- The client or server makes no assumptions about ordering. The Device Information parameters can be specified in any order.

Copyright © 2008 Microsoft Corporation.

- To delete a given Device Information value, the client MUST send the <Set> command with an empty element for that parameter. The server will set that parameter to NULL.

2.2.1.16.2.5.3 Device Password Property

The <Set> command enables the client to set or clear the recovery password of the device.

2.2.1.16.2.6 OofState

The <OofState> element specifies the availability of the <OOF> property.

Parent elements	Child elements	Data type	Number allowed
<Get> (<Oof> response only) <Set> (<Oof> request only)	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 values of <OofState> match those of the Availabilities Service enumeration. <OofState> MUST be set to 2 if the <StartTime> and <EndTime> elements are present.

2.2.1.16.2.7 StartTime

The <StartTime> element is used with the <EndTime> element to specify the range of time during which the user is OOF.

Parent elements	Child elements	Data type	Number allowed
<Get> (<Oof> response only) <Set> (<Oof> request only)	None	**DateTime**	1...1 (required)

The <StartTime> element can be present within the <Get> element of the **Settings** response for the <Oof> property, or within the <Set> element of the **Settings** request for the <Oof> property.

If <StartTime> is present, the <EndTime> element MUST also be present. Otherwise, the client will receive a protocol error.

2.2.1.16.2.8 EndTime

The <EndTime> element is used with the <StartTime> element to specify the range of time during which the user is OOF.

Parent elements	Child elements	Data type	Number allowed
<Get> (<Oof> response only)	None	**DateTime**	1...1 (required)

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
<Set> (<Oof> request only)			

The <EndTime> element can be present within the <Get> element of the **Settings** response for the <Oof> property, or within the <Set> element of the **Settings** request for the <Oof> property.

If <EndTime> is present, the <StartTime> element MUST also be present, and <OofState> MUST be set to 2. Otherwise, the client will receive a protocol error.

2.2.1.16.2.9 OofMessage

The <OofMessage> element contains a set of elements that specify the OOF message for a particular audience.

Parent elements	Child elements	Data type	Number allowed
<Get> (<Oof> response only) <Set> (<Oof> request only)	<AppliesToInternal> <AppliesToExternalKnown> <AppliesToExternalUnknown> <Enabled> <ReplyMessage> <BodyType>	**Container**	0...3

The <OofMessage> element supports the following three audiences: <50>

- 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 an 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 <Set> request, <Set> returns a successful <Status> value of 1 even though the user does not have access to these settings. Similarly, <AppliesToExternalKnown> and <AppliesToExternalUnknown> are returned to the client in a <Get> response even if the sender group is not allowed and is disabled.

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 MUST result in an Access Denied status code.

Copyright © 2008 Microsoft Corporation.

In an OOF <Set> request, the client MUST NOT include the same AppliesTo* element in more than one <OofMessage> element.

2.2.1.16.2.10 BodyType

The <BodyType> element is a string that specifies the format of the OOF message.

Parent elements	Child elements	Data type	Number allowed
<Get> (request only) <OofMessage>	None	**String**	1...1 (required)

The following are the permitted values for the <BodyType> element:

- *Text*
- *HTML*

If <BodyType> has the value of **HTML**, all message strings are sent in the HTML format. If <BodyType> has the value *Text*, the message strings are sent in plain text. Because there is no default value, the <BodyType> node MUST be present.

2.2.1.16.2.11 AppliesToInternal

The <AppliesToInternal> element 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.)

Parent elements	Child elements	Data type	Number allowed
<Get> (request only) <OofMessage>	None	**Empty**	0...1 (Choice of <AppliesToInternal>, <AppliesToExternalKnown>, and <AppliesToExternalUnknown>)

The <AppliesToInternal> element is an **Empty** 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>—Specifies whether an OOF message is sent to this audience while the sending user is OOF.
- <ReplyMessage>—Specifies the OOF message itself.
- <BodyType>—Specifies the format of the OOF message.

The <AppliesToInternal>, <AppliesToExternalKnown>, and <AppliesToExternalUnknown> elements, each of which indicates the audience to which an OOF message pertains, are mutually exclusive.

Copyright © 2008 Microsoft Corporation.

2.2.1.16.2.12 AppliesToExternalKnown

The <AppliesToExternalKnown> element 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.)

Parent elements	Child elements	Data type	Number allowed
<OofMessage>	None	**Empty**	0...1 (Choice of <AppliesToInternal>, <AppliesToExternalKnown>, and <AppliesToExternalUnknown>)

The <AppliesToExternalKnown> element is an **Empty** 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>—Specifies whether an OOF message is sent to this audience while the sending user is OOF.
- <ReplyMessage>—Specifies the OOF reply message.
- <BodyType>—Specifies the format of the OOF message.

The <AppliesToInternal>, <AppliesToExternalKnown>, and <AppliesToExternalUnknown> elements, each of which indicates the audience to which an OOF message pertains, are mutually exclusive.

2.2.1.16.2.13 AppliesToExternalUnknown

The <AppliesToExternalUnknown> element 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.)

Parent elements	Child elements	Data type	Number allowed
<OofMessage>	None	<Empty>	0...1 (Choice of <AppliesToInternal>, <AppliesToExternalKnown>, and <AppliesToExternalUnknown>)

The <AppliesToExternalUnknown> element is an **Empty** element, meaning it has no value or data type. It is distinguished only by the presence or absence of the <AppliesToExternalUnknown/> 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 unknown external users.

The following are the peer elements of the <AppliesToExternalKnown> element:

- <Enabled>—Specifies whether an OOF message is sent to this audience while the sending user is OOF.

Copyright © 2008 Microsoft Corporation.

- <ReplyMessage>—Specifies the OOF reply message.
- <BodyType>—Specifies the format of the OOF message.

The <AppliesToInternal>, <AppliesToExternalKnown>, and <AppliesToExternalUnknown> elements, each of which indicates the audience to which an OOF message pertains, are mutually exclusive.

2.2.1.16.2.14 Enabled

The <Enabled> element specifies whether an OOF message is sent to this audience while the sending user is OOF.

Parent elements	Child elements	Data type	Number allowed
<OofMessage>	None	**Integer**	0...1 (optional)

The <Enabled> element is used in the OOF <Get> response to retrieve the current value. The <Enabled> element is used in the OOF <Set> request to set the value.

The value of <Enabled> is 1 if an OOF message is sent while the sending user is OOF; otherwise, the value is 0.

2.2.1.16.2.15 ReplyMessage

The <ReplyMessage> element specifies the message to be shown to a particular audience when the user is OOF.

Parent elements	Child elements	Data type	Number allowed
<OofMessage>	None	**String**	0...1 (optional)

The <ReplyMessage> can be used in an OOF <Get> response to convey the requested OOF message, or in an OOF <Set> request to set the message that the client wants to send to a particular audience.

2.2.1.16.2.16 DeviceInformation

The <DeviceInformation> element is the container node that is used for sending the client device's properties of the client device to the server.

Parent elements	Child elements	Data type	Number allowed
<Settings>	<Set>	**Container**	0...1 (optional)

It is important to have pertinent information about a user's device for monitoring and troubleshooting. The <DeviceInformation> element is used in the **Settings** command to send the following information about a client device to the server:<51>

- Device model
- International Mobile Equipment Identity (IMEI)
- Device friendly name
- Device operating system

Copyright © 2008 Microsoft Corporation.

- Telephone number
- Device operating system language

The device information is represented as a flat list of settings under the <DeviceInformation> node in the **Settings** command. <DeviceInformation> 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> property supports only the <Set> operation because this information is write-only from the device.

2.2.1.16.2.17 DevicePassword

The <DevicePassword> element is a container node that is used to send the recovery password of the client device to the server.

Parent elements	Child elements	Data type	Number allowed
<Settings>	<Set>	**Container**	0...1 (optional)

Use the <Set> operation on the <DevicePassword> property 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.

2.2.1.16.2.18 UserInformation

The **UserInformation** element is a container node that is used to request a list of a user's e-mail addresses from the server.

Parent elements	Child elements	Data type	Number allowed
<Settings>	<Get> <Status> (<UserInformation> response only)	**Container**	0...1 (optional)

The list of a user's e-mail addresses can be useful, for example, for ensuring that the user is not included when performing a Reply to All operation to an e-mail message.

In a request, the <UserInformation> element contains the <Get> command to indicate that the server is to return all available e-mail addresses for the user.

The **Settings** command supports read-only access to the list of a user's various e-mail addresses via the <Get> command. The client is unable to write this information.

2.2.1.16.2.19 EmailAddresses

The <EmailAddresses> element contains one or more e-mail addresses for the user.

Parent elements	Child elements	Data type	Number allowed
<Get> (response only)	<SMTPAddress> (response only)	**Container**	0...1 (optional)

2.2.1.16.2.20 SMTPAddress

The <SmtpAddress> element specifies one of the user's e-mail addresses.

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
<EmailAddresses>	None	**String**	1...N (optional)

2.2.1.17 SmartForward

The **SmartForward** command is used by clients to forward messages without retrieving the full, original message from the server.

Messages SHOULD NOT be saved directly to the local Sent Items folder by the client; instead, messages can use the <SaveInSentItems> element to automatically have the messages saved 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. 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 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 an error.

When **SmartForward** is applied to an **appointment**, the original message is included by the server as an attachment to the outgoing message. When smart-forwarding a normal message or a meeting, **SmartForward's** behavior is the same as that of the **SmartReply** command.

The **SmartForward** command is similar to the **SendMail** command, but the outgoing message consists of the new message followed by the text of the original message. The full text of the original message is sent. Using the server copy of the original message saves network bandwidth by not downloading the original message 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**, this command will always send the outgoing message by using the UTF8 character set for the body of the message.

2.2.1.17.1 Request

The following code shows the XSD for the **SmartForward** command request.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
        xmlns:tns="ComposeMail:"
        attributeFormDefault="unqualified"
        elementFormDefault="qualified"
        targetNamespace="ComposeMail:"
        xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:element name="SmartForward">
        <xs:complexType>
            <xs:all>
                <xs:element name="ClientId" minOccurs="1" maxOccurs="1">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:minLength value="1"/>
```


Copyright © 2008 Microsoft Corporation.

```
                        <xs:maxLength value="40"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element name="Source" minOccurs="1" maxOccurs="1">
                <xs:complexType>
                    <xs:all>
                        <xs:element name="FolderId" minOccurs="0" maxOccurs="1">
                            <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" maxOccurs="1">
                            <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" maxOccurs="1">
                            <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"
maxOccurs="1" />
                    </xs:all>
                </xs:complexType>
            </xs:element>
            <xs:element name="SaveInSentItems" minOccurs="0" maxOccurs="1" />
            <xs:element name="ReplaceMime" minOccurs="0" maxOccurs="1" />
            <xs:element name="Mime" type="xs:string" minOccurs="1" maxOccurs="1" />
        </xs:all>
    </xs:complexType>
</xs:element>
</xs:schema>
```

2.2.1.17.1.1 SmartForward

The <SmartForward> element is the top-level element in the XML stream. It indicates that the body of the HTTP **POST** contains a **SmartForward** command.

Parent elements	Child elements	Data type	Number allowed
None	<ClientId> (request only) <SaveInSentItems> (request only) <ReplaceMime> (request only) <Source> (request only) <Mime> (request only)	**Container**	1…1 (required)

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
	<Status> (response only)		

2.2.1.17.1.2 ClientId

The <ClientId> element specifies the client's unique message ID (MID).

Parent elements	Child elements	Data type	Number allowed
<SmartForward> (request only)	None	**String** (Up to 40 characters)	1...1 (required)

The <ClientId> MUST be unique for each message, as the server will use the <ClientId> to identify duplicate messages. The <ClientId> can be a simple counter incremented for each new message.

2.2.1.17.1.3 SaveInSentItems

The <SaveInSentItems> element specifies whether a copy of the message should 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.

Parent elements	Child elements	Data type	Number allowed
<SmartForward> (request only)	None	Empty tag	0...1 (optional)

2.2.1.17.1.4 ReplaceMime

The <ReplaceMime> element specifies 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 inline -- if not present, the message includes the reply/forward text.

Parent elements	Child elements	Data type	Number allowed
<SmartForward> (request only)	None	Empty tag	0...1 (optional)

2.2.1.17.1.5 Source

The <Source> element contains information about the source message.

Parent elements	Child elements	Data type	Number allowed
<SmartForward> (request only)	<FolderId> <ItemId> <LongId> <InstanceId>	**Container**	1...1 (required)

2.2.1.17.1.6 FolderId

The <FolderId> element contains the **folder ID (FID)** for the source message, typically from the **FolderSync** command. If the <FolderId> element is present, the <ItemId> element MUST also be present.

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
<Source>	None	**String** (up to 64 characters)	0...1 (optional)

2.2.1.17.1.7 ItemId

The <ItemId> element contains the item ID for the source message, typically from the **Sync** command. If the <ItemId> element is present, the <FolderId> element MUST also be present.

Parent elements	Child elements	Data type	Number allowed
<Source>	None	**String** (up to 64 characters)	0...1 (optional)

2.2.1.17.1.8 LongId

The <LongId> element contains the long ID for the source message, typically from the **Search** command. If the <LongId> element is present, none of the <FolderId>, <ItemId>, or <InstanceId> elements is present.

Parent elements	Child elements	Data type	Number allowed
<Source>	None	**String** (up to 256 characters)	0...1 (optional)

2.2.1.17.1.9 InstanceId

The <InstanceId> element contains the instance of a recurrence for the source item. If the <InstanceId> element is present, both the <FolderId> and <ItemId> elements SHOULD be present.

Parent elements	Child elements	Data type	Number allowed
<Source>	None	**String**	0...1 (optional)

2.2.1.17.1.10 Mime

The <Mime> element contains the MIME-encoded message.

Parent elements	Child elements	Data type	Number allowed
<SmartForward> (request only)	None	**Byte array**	1...1 (required)

The <Mime> content is transferred as an opaque BLOB within the WBXML tags.

2.2.1.17.2 Response

2.2.1.17.2.1 SmartForward

The <SmartForward> element is the top-level element in the XML stream. It indicates that the body of the HTTP **POST** contains a **SmartForward** command.

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
None	<Status> (response only)	**Container**	1...1 (required)

2.2.1.17.2.2 Status

The <Status> element indicates the success or failure of a **SmartForward** command request.

Parent elements	Child elements	Data type	Number allowed
<SmartForward> (response only)	None	**Integer**	0...1 (optional)

If the command succeeds, the <Status> element is returned with code 1. If the command fails, the <Status> element contains a code that indicates the type of failure. If there is no <Status> element returned, the command succeeded.

Valid <Status> values are listed in section 2.2.2.14.

2.2.1.18 SmartReply

The **SmartReply** command is used by clients to reply to messages without retrieving the full, original message from the server.

The **SmartReply** command is similar to the **SendMail** command, but the outgoing message consists of the new message followed by the text of the original message. The full text of the original message is sent. Using the server copy of the original message saves network bandwidth by not downloading the original message and then uploading it again with the reply.

Messages SHOULD NOT be saved directly to the local Sent Items folder by the client; instead, messages can use the <SaveInSentItems> element to automatically have the messages saved 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. 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 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 an error.

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, this command will always send the outgoing message by using the UTF8 character set for the body of the message.

2.2.1.18.1 Request

The following code shows the XSD for the **SmartReply** command request.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
```


Copyright © 2008 Microsoft Corporation.

```
        xmlns:tns="ComposeMail:"
        attributeFormDefault="unqualified"
        elementFormDefault="qualified"
        targetNamespace="ComposeMail:"
        xmlns:xs="http://www.w3.org/2001/XMLSchema">

    <xs:element name="SmartReply">
        <xs:complexType>
            <xs:all>
                <xs:element name="ClientId" minOccurs="1" maxOccurs="1">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:minLength value="1"/>
                            <xs:maxLength value="40"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="Source" minOccurs="1" maxOccurs="1">
                    <xs:complexType>
                        <xs:all>
                            <xs:element name="FolderId" minOccurs="0" maxOccurs="1">
                                <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" maxOccurs="1">
                                <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" maxOccurs="1">
                                <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"
maxOccurs="1" />
                        </xs:all>
                    </xs:complexType>
                </xs:element>
                <xs:element name="SaveInSentItems" minOccurs="0" maxOccurs="1" />
                <xs:element name="ReplaceMime" minOccurs="0" maxOccurs="1" />
                <xs:element name="Mime" type="xs:string" minOccurs="1" maxOccurs="1" />
            </xs:all>
        </xs:complexType>
    </xs:element>
</xs:schema>
```


Copyright © 2008 Microsoft Corporation.

2.2.1.18.1.1 SmartReply

The <SmartReply> element is the top-level element in the XML stream. It indicates that the body of the HTTP **POST** contains a **SmartReply** command.

Parent elements	Child elements	Data type	Number allowed
None	<ClientId> (request only) <SaveInSentItems> (request only) <ReplaceMime> (request only) <Source> (request only) <Mime> (request only) <Status> (response only)	**Container**	1...1 (required)

2.2.1.18.1.2 ClientId

The required element <ClientId> specifies the client's unique message ID (MID).

Parent elements	Child elements	Data type	Number allowed
<SmartReply> (request only)	None	**String** (Up to 40 characters)	1...1 (required)

The <ClientId> MUST be unique for each message, as the server will use the <ClientId> to identify duplicate messages. The <ClientId> can be a simple counter incremented for each new message.

2.2.1.18.1.3 SaveInSentItems

The <SaveInSentItems> element specifies whether a copy of the message should 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.

Parent elements	Child elements	Data type	Number allowed
<SmartReply> (request only)	None	Empty tag	0...1 (optional)

2.2.1.18.1.4 ReplaceMime

The <ReplaceMime> element specifies 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 inline – if not present, the message includes the reply/forward text.

Parent elements	Child elements	Data type	Number allowed
<SmartReply> (request only)	None	Empty tag	0...1 (optional)

2.2.1.18.1.5 Source

The <Source> element contains information about the source message.

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
<SmartReply> (request only)	<FolderId> <ItemId> <LongId> <InstanceId>	**Container**	1...1 (required)

2.2.1.18.1.6 FolderId

The <FolderId> element contains the folder ID (FID) for the source message, typically from the **FolderSync** command. If the <FolderId> element is present, the <ItemId> element MUST also be present.

Parent elements	Child elements	Data type	Number allowed
<Source>	None	**String** (up to 64 characters)	0...1 (optional)

2.2.1.18.1.7 ItemId

The <ItemId> element contains the item ID for the source message, typically from the **Sync** command. If the <ItemId> element is present, the <FolderId> element MUST also be present.

Parent elements	Child elements	Data type	Number allowed
<Source>	None	**String** (up to 64 characters)	0...1 (optional)

2.2.1.18.1.8 LongId

The <LongId> element contains the long ID for the source message, typically from the **Search** command. If the <LongId> element is present, none of the <FolderId>, <ItemId>, or <InstanceId> elements is present.

Parent elements	Child elements	Data type	Number allowed
<Source>	None	**String** (up to 256 characters)	0...1 (optional)

2.2.1.18.1.9 InstanceId

The <InstanceId> element contains the instance of a recurrence for the source item. If the <InstanceId> element is present, both the <FolderId> and <ItemId> elements SHOULD be present.

Parent elements	Child elements	Data type	Number allowed
<Source>	None	**String**	0...1 (optional)

2.2.1.18.1.10 Mime

The <Mime> element contains the MIME-encoded message.

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
<SmartReply> (request only)	None	**Byte array**	1...1 (required)

The <Mime> content is transferred as an opaque BLOB within the WBXML tags.

2.2.1.18.2 Response

2.2.1.18.2.1 SmartReply

The <SmartReply> element is the top-level element in the XML stream. It indicates that the body of the HTTP **POST** contains a **SmartReply** command.

Parent elements	Child elements	Data type	Number allowed
None	<Status> (response only)	**Container**	1...1 (required)

2.2.1.18.2.2 Status

The <Status> element indicates the success or failure of a **SmartReply** command request.

Parent elements	Child elements	Data type	Number allowed
<SmartReply> (response only)	None	**Integer**	0...1 (optional)

If the command succeeds, the <Status> element is returned with code 1. If the command fails, the <Status> element contains a code that indicates the type of failure. If there is no <Status> element returned, the command succeeded.

Valid <Status> values are listed in 2.2.2.14. In particular, a <Status> value of 117 indicates that the server does not allow a reply to the message.

2.2.1.19 Sync

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

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. 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.

Copyright © 2008 Microsoft Corporation.

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.

[MS-ASCMD] 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 in the current request are ghosted. For more information, see section 2.2.1.19.1.12.

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

Copyright © 2008 Microsoft Corporation.

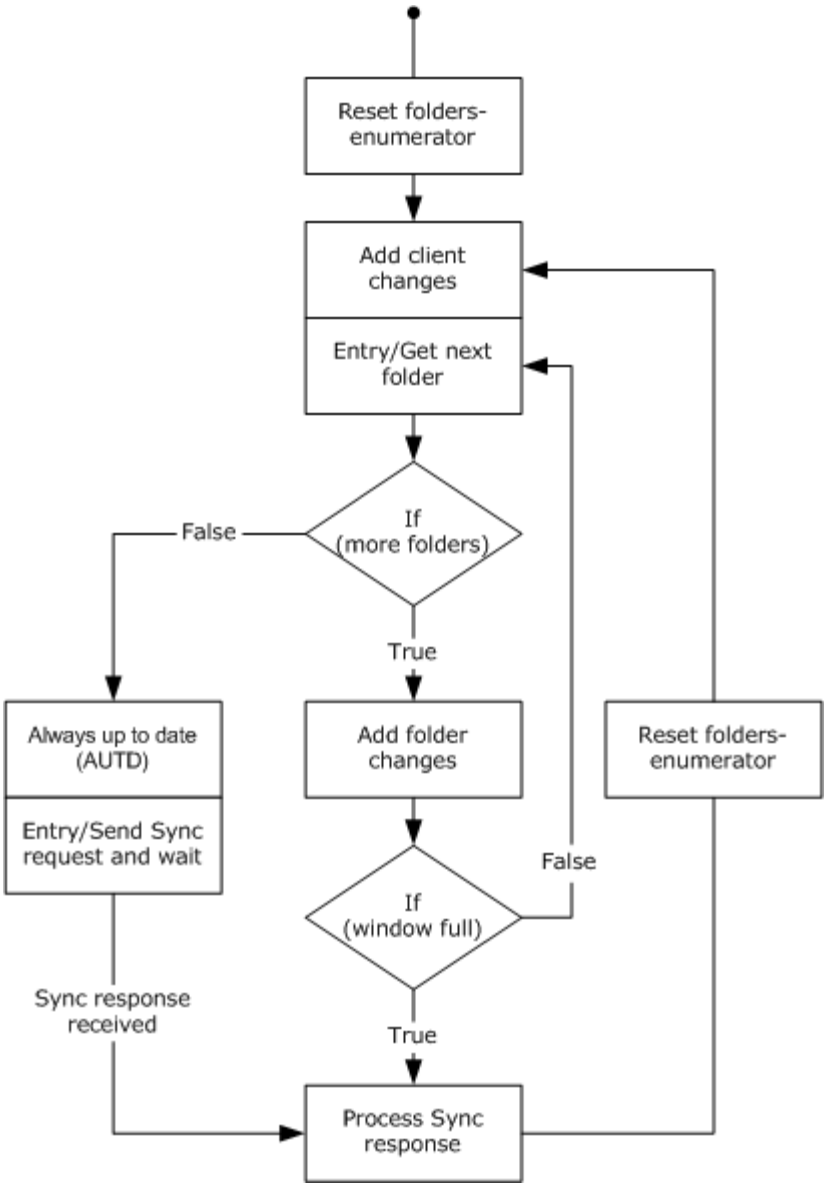

Figure 3: Sync command client processing

2.2.1.19.1 Request

The following XML is optional in the **Sync** command request body.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
        xmlns:tns="AirSync:"
        attributeFormDefault="unqualified"
        elementFormDefault="qualified"
        targetNamespace="AirSync:"
        xmlns:xs="http://www.w3.org/2001/XMLSchema"
        xmlns:contacts="Contacts:"
```


Copyright © 2008 Microsoft Corporation.

```
        xmlns:contacts2="Contacts2:"
        xmlns:calendar="Calendar:"
        xmlns:email="Email:"
        xmlns:airsyncbase="AirSyncBase:"
        xmlns:tasks="Tasks:"
        xmlns:notes="Notes:">

    <xs:import namespace="Contacts2:"/>
    <xs:import namespace="Contacts:"/>
    <xs:import namespace="Email:"/>
    <xs:import namespace="Calendar:"/>
    <xs:import namespace="AirSyncBase:"/>
    <xs:import namespace="Tasks:"/>
    <xs:import namespace="Notes:"/>

    <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="CollectionId">
        <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="ConversationMode"/>
    <xs:element name="Options" minOccurs="0" maxOccurs="2">
      <xs:complexType>
        <xs:choice minOccurs="1" maxOccurs="unbounded">
          <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="Class" type="xs:string" maxOccurs="1" />
          <xs:element name="MaxItems">
            <xs:simpleType>
              <xs:restriction base="xs:integer">
                <xs:minInclusive value="1"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
```


Copyright © 2008 Microsoft Corporation.

```
    </xs:element>
    <xs:element name="SyncKey">
        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:maxLength value="64"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="Class" type="xs:string"/>
    <xs:element name="Sync">
      <xs:complexType>
        <xs:sequence minOccurs="1" maxOccurs="1">
          <xs:element name="Collections" minOccurs="0" maxOccurs="1">
            <xs:complexType>
              <xs:sequence>
                <xs:element minOccurs="0" maxOccurs="unbounded" name="Collection">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element name="SyncKey">
                        <xs:simpleType>
                          <xs:restriction base="xs:string">
                            <xs:maxLength value="64"/>
                          </xs:restriction>
                        </xs:simpleType>
                      </xs:element>
                      <xs:element minOccurs="1" name="CollectionId">
                        <xs:simpleType>
                          <xs:restriction base="xs:string">
                            <xs:maxLength value="64"/>
                          </xs:restriction>
                        </xs:simpleType>
                      </xs:element>
                      <xs:element minOccurs="0" name="Supported">
                        <xs:complexType mixed="true">
                          <xs:sequence minOccurs="0">
                            <xs:choice maxOccurs="unbounded">
                              <xs:group ref="contacts:GhostingProps"/>
                              <xs:group ref="contacts2:GhostingProps"/>
                              <xs:group ref="calendar:GhostingProps"/>
                            </xs:choice>
                          </xs:sequence>
                        </xs:complexType>
                      </xs:element>
                      <xs:element minOccurs="0" name="DeletesAsMoves"/>
                      <xs:element minOccurs="0" name="GetChanges"/>
                      <xs:element minOccurs="0" name="WindowSize">
                        <xs:simpleType>
                          <xs:restriction base="xs:integer">
                            <xs:minInclusive value="0"/>
                          </xs:restriction>
                        </xs:simpleType>
                      </xs:element>
                      <xs:element name="ConversationMode" minOccurs="0" maxOccurs="1"/>
                      <xs:element minOccurs="0" name="Options">
                        <xs:complexType>
                          <xs:choice maxOccurs="unbounded">
                            <xs:element name="FilterType" minOccurs="0">
                              <xs:simpleType>
                                <xs:restriction base="xs:unsignedByte">
```


Copyright © 2008 Microsoft Corporation.

```
                                <xs:minInclusive value="0"/>
                                <xs:maxInclusive value="8"/>
                              </xs:restriction>
                            </xs:simpleType>
                          </xs:element>
                          <xs:element name="Class" type="xs:string" minOccurs="0"
maxOccurs="1" />
                          <xs:element ref="airsyncbase:BodyPreference" minOccurs="0"
maxOccurs="unbounded" />
                          <xs:element minOccurs="0" name="Conflict" type="xs:unsignedByte"/>
                          <xs:element minOccurs="0" maxOccurs="1" 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 minOccurs="0" maxOccurs="1" 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="MaxItems" minOccurs="0" maxOccurs="1">
                            <xs:simpleType>
                              <xs:restriction base="xs:integer">
                                <xs:minInclusive value="1"/>
                              </xs:restriction>
                            </xs:simpleType>
                          </xs:element>
                        </xs:choice>
                      </xs:complexType>
                    </xs:element>
                    <xs:element minOccurs="0" name="Commands">
                      <xs:complexType>
                        <xs:choice maxOccurs="unbounded">
                          <xs:element minOccurs="0" maxOccurs="unbounded" name="Change">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element name="ServerId">
                                  <xs:simpleType>
                                    <xs:restriction base="xs:string">
                                      <xs:maxLength value="64"/>
                                    </xs:restriction>
                                  </xs:simpleType>
                                </xs:element>
                                <xs:element name="ApplicationData">
                                  <xs:complexType>
                                    <xs:sequence>
                                      <xs:choice maxOccurs="unbounded">
                                        <xs:element ref="email:Flag"/>
                                        <xs:element ref="email:Read"/>
                                        <xs:element ref="email:Categories"/>
                                        <xs:element ref="calendar:OrganizerName"/>
                                        <xs:element ref="calendar:OrganizerEmail"/>
                                        <xs:element ref="calendar:Exceptions"/>
```


Copyright © 2008 Microsoft Corporation.

```
                                        <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="airsyncbase:NativeBodyType"/>
                                        <xs:element ref="airsyncbase:Body"/>
                                        <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:AssistnamePhoneNumber"/>
                                        <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"/>
```


Copyright © 2008 Microsoft Corporation.

```
                <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: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="contacts:YomiLastName"/>
                <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:element ref="notes:Subject"/>
                <xs:element ref="notes:MessageClass"/>
                <xs:element ref="notes:LastModifiedDate"/>
                <xs:element ref="notes:Categories"/>
              </xs:choice>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element minOccurs="0" maxOccurs="unbounded" name="Delete">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ServerId">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:maxLength value="64"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
```


Copyright © 2008 Microsoft Corporation.

```
                          <xs:element minOccurs="0" maxOccurs="unbounded" name="Add">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element name="Class" type="xs:string" minOccurs="0"
maxOccurs="1" />
                                <xs:element name="ClientId">
                                  <xs:simpleType>
                                    <xs:restriction base="xs:string">
                                      <xs:maxLength value="64"/>
                                    </xs:restriction>
                                  </xs:simpleType>
                                </xs:element>
                                <xs:element name="ApplicationData">
                                  <xs:complexType>
                                    <xs:sequence>
                                      <xs:choice maxOccurs="unbounded">
                                        <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="airsyncbase:NativeBodyType"/>
                                        <xs:element ref="airsyncbase:Body"/>
                                        <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:AssistnamePhoneNumber"/>
                                        <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"/>
```


Copyright © 2008 Microsoft Corporation.

```
<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="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:element ref="notes:Subject"/>
<xs:element ref="notes:MessageClass"/>
<xs:element ref="notes:LastModifiedDate"/>
<xs:element ref="notes:Categories"/>
<xs:element ref="email:To"/>
```


Copyright © 2008 Microsoft Corporation.

```
                                <xs:element ref="email:From"/>
                                <xs:element ref="email:DateReceived"/>
                                <xs:element ref="email:InternetCPID"/>
                                <xs:element ref="email:Importance"/>
                                <xs:element ref="email:Flag"/>
                                <xs:element ref="email:Read"/>
                              </xs:choice>
                            </xs:sequence>
                          </xs:complexType>
                        </xs:element>
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                  <xs:element minOccurs="0" maxOccurs="unbounded" name="Fetch">
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element name="ServerId">
                          <xs:simpleType>
                            <xs:restriction base="xs:string">
                              <xs:maxLength value="64"/>
                            </xs:restriction>
                          </xs:simpleType>
                        </xs:element>
                      </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 name="Wait" minOccurs="0" maxOccurs="1">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="1"/>
      <xs:maxInclusive value="59"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="HeartbeatInterval" minOccurs="0" maxOccurs="1">
   <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="1"/>
      <xs:maxInclusive value="3540"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="WindowSize" minOccurs="0" maxOccurs="1">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="0"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="Partial" minOccurs="0" maxOccurs="1" />
```


Copyright © 2008 Microsoft Corporation.

```
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:schema>
```

2.2.1.19.1.1 Sync

The <Sync> element is the top-level element in the XML document. It identifies the body of the HTTP **POST** as containing a **Sync** command.

Parent elements	Child elements	Data type	Number allowed
None	<Collections> <Partial> (request only) <Wait> (request only) <HeartbeatInterval> (request only) <WindowSize> (request only) <Limit> (response only) <Status> (response only)	**Container**	1...1 (required)

The <Sync> element can also include one or more explicit namespace attributes.

The <Limit> element and <Collections> element are mutually exclusive in a **Sync** response. That is, a **Sync** response can include either a <Limit> element or a <Collections> element, but not both.

2.2.1.19.1.1.1 Empty Sync Request

If the client sends a **Sync** request with no client changes, 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.

When the server receives the empty **Sync** response, 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 4.3.10.

2.2.1.19.1.2 Wait

The <Wait> element specifies, in a request, the number of minutes that the server SHOULD delay a response and, in a response, the number of minutes that the server can wait for any changes before responding.

Parent elements	Child elements	Data type	Number allowed
Sync	None	**Integer**	0...1 (optional)

Copyright © 2008 Microsoft Corporation.

Valid values for <Wait> 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> value of 14 and a <Limit> element.

Either <Wait> or <HeartbeatInterval> (section 2.2.1.19.1.3) can be specified, but not both. If both are specified, the server response will include a <Status> value of 4.

When <Wait> is used in a **Sync** request, the element indicates to the server that a response SHOULD be delayed either until the wait-interval, which is indicated by the contents of the <Wait> element, elapses or until any of the collections that are included in the request have changed.

It is at the discretion of the client to send the **Wait** element; the server is only guaranteed to respond immediately when **Wait** is not present. The client typically wants a server response immediately in the following cases:

- The client adds new items by using the **Add** element. In this case, an immediate response is required because the client requires the server-provided item ID to track changes to those new items.
- The client sends up a large change by using the **Change** element. In this case, a delayed response increases the possibility that the client has to resend the change because of a lost connection.

Although the server is only guaranteed to respond immediately when **Wait** and **HeartbeatInterval** are not present, the server SHOULD always respond immediately to a **Sync** request that includes an **Add** or a **Change**, unless the addition or change involves only flags.

A hard delete of tasks or calendar items will cause a waited **Sync** to finish. The benefit of this is a better user experience. For example, a user will not get **reminders** for deleted meetings. A **hard delete** is infrequent and rarely results in an extra roundtrip. A flagging change or a move out of (and not into) a folder which is being synced SHOULD NOT cause the request to finish early.

2.2.1.19.1.3 HeartbeatInterval

The <HeartbeatInterval> element specifies, in a request, the number of seconds that the server SHOULD delay a response and, in a response, the number of seconds that the server can wait for any changes before responding.

Parent elements	Child elements	Data type	Number allowed
<Sync>	None	**Integer**	0...1 (optional)

Valid values for <HeartbeatInterval> are 1 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> value of 14 and a <Limit> element.

Either <HeartbeatInterval> or <Wait> (section 2.2.1.19.1.2) can be specified, but not both. If both are specified, the server response will include a <Status> value of 4.

When <HeartbeatInterval> is used in a **Sync** request, the element indicates to the server that a response SHOULD be delayed either until the interval, which is indicated by the contents of the <HeartbeatInterval> element, elapses or until any of the collections that are included in the request have changed.

It is at the discretion of the client to send the <HeartbeatInterval> element; the server is only guaranteed to respond immediately when neither <HeartbeatInterval> nor <Wait> (section

Copyright © 2008 Microsoft Corporation.

2.2.1.19.1.2) are present. The client typically requires a server response immediately in the following cases:

- The client adds new items by using the <Add> element. In this case, an immediate response is required because the client requires the server-provided item ID to track changes to those new items.
- The client sends up a large change by using the <Change> element. In this case, a delayed response increases the possibility that the client has to resend the change because of a lost connection.

Although the server is only guaranteed to respond immediately when <HeartbeatInterval> and <Wait> (section 2.2.1.19.1.2) are not present, the server SHOULD always respond immediately to a **Sync** request that includes an <Add> or a <Change>, unless the addition or change involves only flags.

A hard delete of tasks or calendar items will cause a waited **Sync** to finish. The benefit of this is a better user experience. For example, a user will not get reminders for deleted meetings. A hard delete is infrequent and rarely results in an extra roundtrip. A flagging change or a move out of (and not into) a folder which is being synced SHOULD NOT cause the request to finish early.

2.2.1.19.1.4 Partial

The <Partial> element 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.

Parent elements	Child elements	Data type	Number allowed
<Sync> (request)	None	**Empty**	0...1 (optional)

The <Partial> element is an **Empty** 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** request is valid with just a <Partial> element and either a <Wait> or <HeartbeatInterval> element, a <WindowSize> element, a <Collections> element, or any combination of the three. A **Sync** request requires, at least, either a <Partial> element or a <Collections> element.

When a 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, etc) without sending up every unchanged collection.

2.2.1.19.1.5 WindowSize

The <WindowSize> element is sent from the client to the server to specify a maximum number of changed items in a collection or a request that SHOULD be included in the synchronization response.

Parent elements	Child elements	Data type	Number allowed
<Collection> (request) <Sync>	None	**Integer**	0...1 (optional)

Copyright © 2008 Microsoft Corporation.

The server sends the requested number of changes, and if there are more, the server includes a <MoreAvailable> element in the response. The maximum value for the <WindowSize> element is 512.

The <WindowSize> element appears only in requests that are sent to the server from the client. If <WindowSize> is omitted, the server behaves as if a <WindowSize> element with a value of 100 were submitted.

A good value for <WindowSize> is 100. Small values increase the load on the server, increase bandwidth, and decrease battery life because of the additional requests that are required to obtain all changes. Larger values cause larger responses, which are more susceptible to communication errors. A lower <WindowSize> value 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 window size is changed during a synchronization transaction, the server returns a <MoreAvailable> element in the response but no items are returned. 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. <WindowSize> can still be specified at the collection level and the server MUST honor both settings.

When <WindowSize> is not sent by the client, the server assumes a default <WindowSize> of 100. This value is used by most clients and this default will save those clients some bytes. The default is also in effect at the collection level. The maximum value for <WindowSize> is 512. The server interprets values above 512 and 0 (zero) as 512.

The 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** 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** response for any particular collection depends on the <WindowSize> of the collection, the overall number of changes that are already included in the response, and the global <WindowSize>. 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.

The following synchronization request specifies that up to 100 changes be sent from the server back to the client. If there are more than 100 changes on the server, the <MoreAvailable> element is included in the response.

Request

```
<Collection>
      <Class>Email</Class>
      <SyncKey>1</SyncKey>
      <CollectionId>1</CollectionId>
      <DeletesAsMoves/>
```


Copyright © 2008 Microsoft Corporation.

```
        <GetChanges/>
        <WindowSize>100</WindowSize>
  </Collection>
```

2.2.1.19.1.6 Add

The <Add> element can be used to create a new object in a collection on the client or on the server.

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 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 you add a calendar item, the <Timezone> property MUST be specified first and the <StartTime> and <EndTime> properties MUST be present in the <ApplicationData> element.

Parent elements	Child elements	Data type	Number allowed
<Commands> <Responses> (response only)	<ServerId> (response only, see below) <ClientId> <ApplicationData> <Status> (response only)	**Container**	0...N (optional)

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 e-mail items from the client to the server, or to modify the contents of the recipient information cache. If a client attempts to add e-mails to the server, or attempts to add items to the recipient information cache, then a <Status> value of 4 is returned.

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 command 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 SHOULD assume that the command succeeded unless informed otherwise.

2.2.1.19.1.7 ApplicationData

The <ApplicationData> element encloses data for a particular object, such as a contact, e-mail message, calendar appointment, or task item. The <ApplicationData> element can be used to add or change items on the client device or server. The format of this data is determined by the schema for the object.

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
<Add> <Change>	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 (required)

The following <ApplicationData> element is used to add a contact item, identified by the <ServerId> element, to a folder on the client device.

Response

```
<Add>
          <ServerId> 2:6</ServerId>
          <ApplicationData>
<A:Body></A:Body>
             <A:Email1Address>"jdobney@fourthcoffee.com"
&lt;jdobney@fourthcoffee.com&gt;</A:Email1Address>
                <A:FileAs>Dobney, JoLynn Julie</A:FileAs>
                <A:FirstName>JoLynn</A:FirstName>
                <A:HomePhoneNumber>425 555 1234</A:HomePhoneNumber>
                <A:MiddleName>Julie</A:MiddleName>
                <A:MobilePhoneNumber>425 555 1111</A:MobilePhoneNumber>
                <A:CompanyName>Fourth Coffee</A:CompanyName>
                <A:LastName>Dobney</A:LastName>
                <A:BusinessPhoneNumber>425 555 5555</A:BusinessPhoneNumber>
                <A:JobTitle>Usability Engineer</A:JobTitle>
          </ApplicationData>
</Add>
```

2.2.1.19.1.8 Change

The <Change> element modifies properties of an existing object on the client device or the server. The object to change is identified by its <ServerId> element.

Parent elements	Child elements	Data type	Number allowed
<Commands> <Responses> (response only)	<ServerId> <ApplicationData> <Status> (response only)	**Container**	0...N (optional)

One or more <Change> elements can appear as a child of the <Commands> element for a particular collection.

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

- If there is only a <Flag> or <Read> change (that is, if only a <Flag> or <Read> node is present), all other properties will remain unchanged and nothing else has to be sent.
- If an <Exceptions> node is not specified, the properties for that exceptions node will remain unchanged. If an <Exception> node within the <Exceptions> node is not present, that particular **exception** will remain unchanged.
- If <Body>, <Data>, <Picture>, or <RTF> nodes are not present, the corresponding properties will remain unchanged.

Copyright © 2008 Microsoft Corporation.

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.

2.2.1.19.1.9 ClientId

The <ClientId> is a unique identifier that is generated by the client to temporarily identify a new object that is being created by using the <Add> command. The client includes the <ClientId> element in the <Add> command 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**.

Parent elements	Child elements	Data type	Number allowed
<Add>	None	**String** (Typically an integer)	Request: 1 (required)

The <ClientId> element is a unique identifier that consists of up to 40 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.

2.2.1.19.1.10 Collection

The <Collection> element wraps commands and options that apply to a particular collection.

Parent elements	Child elements	Data type	Number allowed
<Collections>	<SyncKey> <Supported> (request only) <CollectionId> <DeletesAsMoves> (request only) <GetChanges> (request only) <WindowSize> (request only) <Option> (request only) <ConversationMode> (request only) <Status> (response only) <MoreAvailable> (response only) <Commands> <Responses> (response only)	**Container**	1...N (required)

The <Collection> element contains identification information (<CollectionID>), synchronization state (**SyncKey**), commands (<GetChanges>, <Commands>), and options (<WindowSize>, <Options>, <DeleteAsMoves>, <MoreAvailable>). Only one <Collection> can be specified in a **Sync** command.

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

1. <SyncKey>

Copyright © 2008 Microsoft Corporation.

2. <CollectionId>
3. <Supported>
4. <DeletesAsMoves>
5. <GetChanges>
6. <WindowSize>
7. <ConversationMode>
8. <Options>
9. <Commands>

The <Collection> element appears in both <Sync> requests and responses. The form is similar, although some child elements are valid in only one context.

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

2.2.1.19.1.11 SyncKey

The <SyncKey> element contains a value that is used by the server to mark the synchronization **state** of a collection.

Parent elements	Child elements	Data type	Number allowed
<Collection>	None	**String** (Up to 64 characters)	1 (required)

A synchronization key of value 0 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.

2.2.1.19.1.12 Supported

The <Supported> element is used by the client to specify which contact and calendar properties in a **Sync** request are managed by the client. <52>Properties that are not named by the <Supported> element are said to be ghosted.

Parent elements	Child elements	Data type	Number allowed
<Collection>	Any contact or **calendar** property. [Only container elements (<Children>, <Categories>) are valid. Their child elements (<Child>, <Category>) are not valid.]	**Container**	0...1 (optional)

Copyright © 2008 Microsoft Corporation.

By default, a server preserves the value of ghosted properties if the corresponding element is not included in a **Sync** command. If an element is listed as a child of the <Supported> element, then the client is signifying that it will always transmit the current value of this element to the server. If the client does not transmit a <Supported> element in a **Sync** request, the server will delete the currently stored value on subsequent **Sync** requests.The supported properties list is sent on the initial synchronization only; the server remembers the list for subsequent synchronizations.

The initial **Sync** request MUST include a <CollectionId> node, which MUST always precede the <Supported> node. See the <Collection> element (section 2.2.1.19.1.12) for the order of elements within the <Collection> node. This order is strictly enforced.

```
<Collection>
   <SyncKey>0</SyncKey>
   <CollectionId>2</CollectionId>
   <Supported>
        <c:FirstName/>
        <c:MiddleName/>
        <c:LastName/>
        <c:HomePhoneNumber/>
        <c:MobilePhoneNumber/>
        <c:BusinessPhoneNumber/>
        <c:Email1Address/>
    </Supported>
</Collection>
```

For more information on which properties are ghosted, consult the ActiveSync class protocols, [MS-ASCAL] and [MS-ASCNTC].

2.2.1.19.1.13 GetChanges

The <GetChanges> element requests the server to include in its response any pending changes to the collection that is specified by the <ServerId> element. If there have been changes since the last synchronization, the server response includes a <Commands> element that contains additions, deletions, and changes.

Parent elements	Child elements	Data type	Number allowed
<Collection> (request only)	None	**Boolean**	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), which is the default, indicates that the client wants the server changes to be returned. The default is assumed when the <GetChanges> element is either empty or not present.

For requests with a <SyncKey> value of 0, the <GetChanges> element is a protocol error.

2.2.1.19.1.14 ConversationMode

The optional element <ConversationMode><53> 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.

Copyright © 2008 Microsoft Corporation.

Setting <ConversationMode> to 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.

Parent elements	Child elements	Data type	Number allowed
<Collection> (request only)	None	**Boolean**	0...1 (optional)

Specifying <ConversationMode> for collections that do not store e-mails results in an invalid XML error, status code 103.

2.2.1.19.1.15 CollectionId

The <CollectionId> element specifies the server ID of the folder to be synchronized.

Parent elements	Child elements	Data type	Number allowed
<Collection>	None	**String** (Up to 64 characters)	1 (required)

The server ID of the folder is obtained from the <ServerId> element of a previous **FolderSync** or **FolderCreate** command.

2.2.1.19.1.16 Collections

The <Collections> element serves as a container for the <Collection> element.

Parent elements	Child elements	Data type	Number allowed
<Sync>	<Collection>	**Container**	0...1 (optional)

The <Collections> element appears both in synchronization requests and responses. The structure is identical.

The <Collections> element is optional. If <Collections> is present, it can contain multiple <Collection> elements.

Request/Response

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:">
<Collections>
    <Collection>
        ...
    </Collection>
</Collections>
```

2.2.1.19.1.17 Commands

The <Commands> element is a container for commands that apply to a collection. Available commands are <Add>, <Delete>, <Change>, and <Fetch>. Client commands are sent in the **POST** request; server commands are sent in the **POST** response.

Copyright © 2008 Microsoft Corporation.

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

Parent elements	Child elements	Data type	Number allowed
<Collection>	<Add> <Delete> <Change> <Fetch> <SoftDelete> (response only)	**Container**	0...1 (optional)

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

Request/Response

```
<Collection>
<Commands>
 <Add>
 ...
 </Add>
 <Delete>
 ...
 </Delete >
 <Change>
 ...
 </Change >
 <Fetch>
 ...
 </Fetch>
 </Commands>
</Collection>
```

2.2.1.19.1.18 Delete

The <Delete> element deletes an object on the client or server. The object is identified by its <ServerId> element.

Parent elements	Child elements	Data type	Number allowed
<Commands>	<ServerId>	**Container**	0...N (optional)

2.2.1.19.1.19 Fetch

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.

Parent elements	Child elements	Data type	Number allowed
<Commands> (request only) <Responses> (response only)	<ServerId> <Status> (response only) <ApplicationData> (response only)	**Container**	0...N (optional)

Copyright © 2008 Microsoft Corporation.

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

2.2.1.19.1.20 DeletesAsMoves

The <DeletesAsMoves> element indicates that any deleted items SHOULD be moved to the Deleted Items folder.

Parent elements	Child elements	Data type	Number allowed
<Collection> (request only)	None	**Boolean**	0...1 (optional)

The <DeletesAsMoves> element appears only in requests to the server from the client. If the <DeleteAsMoves> 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.

2.2.1.19.1.21 Options

The <Options> element is a container that encloses elements that control certain aspects of how the synchronization is performed.

Parent elements	Child elements	Data type	Number allowed
<Collection> (request only)	<FilterType> <Conflict> <MIMETruncation> <MIMESupport> <Class> <MaxItems> <airsyncbase:BodyPreference> <>	**Container**	0...2 (optional)

This element is optional, but if it is present, it MUST include at least one child element. The <Options> element appears only in requests to the server from the client.

Additional synchronization options enable the client to specify truncation and content settings. These settings are encapsulated within a <BodyPreference> node within the <Options> element as follows:

```
<airsyncbase:BodyPreference>
    <airsyncbase:Type>1</Type>
    <airsyncbase:TruncationSize>512</TruncationSize>
    <airsyncbase:AllOrNone/>
    <airsyncbase:Preview>255</Preview>
</airsyncbase:BodyPreference>
```

Because synchronization options are specified on a collection, the client can specify a unique <BodyPreference> for each collection that it is being synchronized. For more details about the <BodyPreference> element, see [MS-ASAIRS] section 2.2.1.4.

Copyright © 2008 Microsoft Corporation.

The server preserves the options across requests. Therefore, the options MUST only be sent once per collection unless the client sends a <SyncKey> value of 0. Whenever the client specifies new options by including an <Options> node in the request, the server MUST replace the original options with the new options.

The following <Options> element specifies that items in the collection that are older than three days SHOULD NOT be returned to the client, that items MUST be truncated to 512 characters if they are larger, and that, if there are any item conflicts, the server MUST replace the client items.

Request

```
<Collection>
    <Options>
        <FilterType>2</FilterType>
        <Conflict>1</Conflict>
    </Options>
</Collection>
```

2.2.1.19.1.22 Conflict

The <Conflict> element 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.

Parent elements	Child elements	Data type	Number allowed
<Options>	None	**Unsigned byte**	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 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> 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 use the element on an object-by-object basis in a single **Sync** command.

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.

If a <Delete> command conflicts with an <Add> or <Change> command, the <Delete> takes precedence.

Copyright © 2008 Microsoft Corporation.

2.2.1.19.1.23 FilterType

The <FilterType> element specifies an optional time window for the objects that are sent from the server to the client. It applies to e-mail and calendar collections. If <FilterType> is specified, the server sends only objects that are dated within the specified time window.

Parent elements	Child elements	Data type	Number allowed
<Options> (request only)	None	**Unsigned byte**	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 back	Yes	No	No
2	3 days back	Yes	No	No
3	1 week back	Yes	No	No
4	2 weeks back	Yes	Yes	No
5	1 month back	Yes	Yes	No
6	3 months back	No	Yes	No
7	6 months back	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> commands for objects on the client when they become older than the window.

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 value and the present time are returned to the client. For example, an appointment that occurred two weeks ago is returned when the FilterType value is set to 5 (1 month back).

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

If the <FilterType> element is omitted, all objects are sent from the server without regard for their age. Clients MUST send a maximum of 1 <FilterType> element.

The server ignores the <FilterType> element if it is included in a contact **Sync** request, no error is thrown.

The following <Options> element in a synchronization request on an Inbox specifies that only e-mail messages that date back three days are returned to the client in the server synchronization response.

```
<Options>
    <FilterType>2</FilterType>
```


Copyright © 2008 Microsoft Corporation.

```
</Options>
```

2.2.1.19.1.24 MIMETruncation

The <MIMETruncation> element is included in the <Options> element of a client **Sync** command request to specify to the server whether the MIME data of the e-mail item SHOULD be truncated when it is sent from the server to the client.

Parent elements	Child elements	Data type	Number allowed
<Options> (request only)	None	**Unsigned byte**	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.

If the size of the MIME data exceeds the value that is specified by the client in the <MIMETruncation> element, the string that is returned in the <MIMEData> element is truncated up to the <MIMETruncation> value. The value of the <MIMESize> element will then contain the original size, in characters, of the MIME data.

The size of the truncated message returned in the response is not the exact size of <MIMETruncation> value, the <MIMETruncation> size is an approximate value. This is because line feeds are treated as one character locally, but are counted as two characters during truncation.

2.2.1.19.1.25 MIMESupport

The <MIMESupport> element is included in the <Options> element of a client **Sync** command request to enable MIME support for e-mail items that are sent from the server to the client.

Parent elements	Child elements	Data type	Number allowed
<Options> (request only)	None	**Unsigned byte**	0...1 (optional)

The following table shows valid values for the element.

Copyright © 2008 Microsoft Corporation.

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 **Sync** request MUST include the following in the <Options> element:

- The <MIMESupport> element to tell the server to return MIME for S/MIME-only/All/None messages.
- The <BodyPreference> element with its child element, <Type>, which contains a value of 4 to inform the server that the device can read the MIME BLOB.

The response from the server MUST include the <Body> element, which is a child of the <ApplicationData> element. The <Body> element is a complex element and MUST contain the following child nodes in an S/MIME synchronization response:

- The <Type> element with a value of 4 to inform the device that the data is a MIME BLOB.
- The <EstimatedDataSize> element to specify the rough total size of the data.
- The <Truncated> element to indicate whether the MIME BLOB is truncated.
- The <Data> element that contains the full MIME BLOB.

For more details about the <Body> element or the <BodyPreference> element, see [MS-ASAIRS] section 2.2.1.3 or 2.2.1.4, respectively.

2.2.1.19.1.26 Class

The optional element <Class><54> assigns the filters within the <Options> container to a given class.

Parent elements	Child elements	Data type	Number allowed
<Options> (request only)	None	**String**	0...1 (optional)

Options for the same <Class> within the same <Collection> MUST NOT be 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 class "SMS". To also sync e-mail messages at the same time, include another <Options> node with class "Email".

The valid <Class> element values are:

- *Tasks*
- *Email*
- *Calendar*

Copyright © 2008 Microsoft Corporation.

- *Contacts*
- *Document*
- *SMS*

2.2.1.19.1.27 MaxItems

The optional element <MaxItems><55> specifies the maximum number of recipients to keep synchronized from within the recipient information cache. This element MUST only be included in a request when the <CollectionId> maps to the recipient information cache. The value of <MaxItems> does not specify the limit of estimates available, it only specifies the number of items, as a complete replacement would be double the number of items in the store (n deletes plus n additions).

Parent elements	Child elements	Data type	Number allowed
<Options>	None	**Integer**	0...1 (optional)

2.2.1.19.2 Response

The following code shows the XSD for the **Sync** command response.

```
<?xml version="1.0" ?>
<xs:schema xmlns:tns="AirSync:" attributeFormDefault="unqualified"
elementFormDefault="qualified" targetNamespace="AirSync:"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:contacts="CONTACTS:"
xmlns:contacts2="CONTACTS2:" xmlns:calendar="CAL:" xmlns:email="EMAIL:"
xmlns:email2="EMAIL2:" xmlns:airsyncbase="AirSyncBase:" xmlns:tasks="TASKS:">
  <xs:import namespace="CONTACTS:" />
  <xs:import namespace="CONTACTS2:" />
  <xs:import namespace="EMAIL:" />
  <xs:import namespace="EMAIL2:" />
  <xs:import namespace="CAL:" />
  <xs:import namespace="AirSyncBase:"/>
  <xs:import namespace:="TASKS:"/>
  <xs:element name="Sync" minOccurs="0">
    <xs:complexType>
      <xs:element minOccurs="1" name="Status" type="xs:unsignedByte" />
      <xs:choice>
      <xs:element minOccurs="0" maxOccurs="1" name="Limit" type="xs:integer" />
      <xs:element minOccurs="0" name="Collections">
        <xs:complexType>
        <xs:sequence>
          <xs:element name="Collection" minOccurs="0" maxOccurs="unbounded">
            <xs:complexType>
            <xs:sequence>
              <xs:choice maxOccurs="unbounded">

<xs:element name="SyncKey" type="xs:string" />
                <xs:element name="CollectionId" type="xs:string" />
                <xs:element name="Status" type="xs:unsignedByte" />
                <xs:element minOccurs="0" maxOccurs="1" name="Commands">
                  <xs:complexType>
                  <xs:sequence>
                    <xs:element minOccurs="0" maxOccurs="unbounded" name="Delete">
                      <xs:complexType>
```


Copyright © 2008 Microsoft Corporation.

```
  <xs:sequence>
    <xs:element name="ServerId" type="xs:string" />
  </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element minOccurs="0" maxOccurs="unbounded" name="SoftDelete">
  <xs:complexType>
  <xs:sequence>
    <xs:element name="ServerId" type="xs:string" />
  </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element minOccurs="0" maxOccurs="unbounded" name="Change">
  <xs:complexType>
  <xs:sequence>
    <xs:element name="ServerId" type="xs:string" />
    <xs:element name="ApplicationData">
      <xs:complexType>
      <xs:sequence>
        <xs:choice maxOccurs="unbounded">
          <xs:element ref="calendar:Timezone" />
          <xs:element ref="calendar:DtStamp" />
          <xs:element ref="calendar:StartTime" />
          <xs:element ref="calendar:Subject" />
     <xs:element ref="calendar:UID" />
          <xs:element ref="calendar:OrganizerName" />
          <xs:element ref="calendar:OrganizerEmail" />
          <xs:element ref="calendar:Location" />
          <xs:element ref="calendar:EndTime" />
          <xs:element ref="calendar:Recurrence" />
          <xs:element ref="calendar:Body" />
          <xs:element ref="calendar:Categories" />
          <xs:element ref="calendar:Sensitivity" />
          <xs:element ref="calendar:BusyStatus" />
          <xs:element ref="calendar:AllDayEvent" />
          <xs:element ref="calendar:Reminder" />
          <xs:element ref="calendar:Exceptions" />
          <xs:element ref="calendar:MeetingStatus" />
          <xs:element ref="calendar:Rtf" />
          <xs:element ref="calendar:Attendees" />
          <xs:element ref="contacts:Rtf" />
          <xs:element ref="contacts:Anniversary" />
          <xs:element ref="contacts:AssistantName" />
          <xs:element ref="contacts:AssistnamePhoneNumber" />
          <xs:element ref="contacts:Birthday" />
          <xs:element ref="contacts:Body" />
          <xs:element ref="contacts:BodySize" />
          <xs:element ref="contacts:BodyTruncated" />
          <xs:element ref="contacts:Business2PhoneNumber" />
          <xs:element ref="contacts:BusinessCity" />
          <xs:element ref="contacts:BusinessCountry" />
          <xs:element ref="contacts:BusinessPostalCode" />
          <xs:element ref="contacts:BusinessState" />
          <xs:element ref="contacts:BusinessStreet" />
          <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" />
```


Copyright © 2008 Microsoft Corporation.

```
<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:HomeCity" />
<xs:element ref="contacts:HomeCountry" />
<xs:element ref="contacts:HomePostalCode" />
<xs:element ref="contacts:HomeState" />
<xs:element ref="contacts:HomeStreet" />
<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:OtherCity" />
<xs:element ref="contacts:OtherCountry" />
<xs:element ref="contacts:OtherPostalCode" />
<xs:element ref="contacts:OtherState" />
<xs:element ref="contacts:OtherStreet" />
<xs:element ref="contacts:PagerNumber" />
<xs:element ref="contacts:picture" />
<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="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="tasks:Body" />
<xs:element ref="tasks:BodySize" />
<xs:element ref="tasks:BodyTruncated" />
<xs:element ref="tasks:Subject" />
<xs:element ref="tasks:Categories" />
<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:Recurrence" />
<xs:element ref="tasks:Complete" />
<xs:element ref="tasks:DateCompleted" />
<xs:element ref="tasks:Sensitivity" />
```


Copyright © 2008 Microsoft Corporation.

```
                        <xs:element ref="tasks:ReminderTime" />
                        <xs:element ref="tasks:ReminderSet" />
                        <xs:element ref="tasks:Rtf" />
                      </xs:choice>
                    </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element minOccurs="0" maxOccurs="unbounded" name="Add">
                <xs:complexType>
                <xs:sequence>
                  <xs:element name="ServerId" type="xs:string" />
                  <xs:element name="ApplicationData">
                    <xs:complexType>
                    <xs:sequence>
                      <xs:choice maxOccurs="unbounded">
                        <xs:element ref="email:To" />
                        <xs:element ref="email:From" />
                        <xs:element ref="email:Reply-To" />
                        <xs:element ref="email:Subject" />
                        <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:Attachments" >
                         <xs:complexType>
                             <xs:sequence>
                                 <xs:element ref="airsyncbase:Attachment">
                                     <xs:complexType>
                                         <xs:all>
<xs:element ref="airsyncbase:DisplayName" />
                                             <xs:element
ref="airsyncbase:FileReference" />
                                             <xs:element ref="airsyncbase:Method" />
                                             <xs:element
ref="airsyncbase:EstimatedDataSize" />
                                             <xs:element ref="airsyncbase:ContentId" />
                                             <xs:element
ref="airsyncbase:ContentLocation" />
                                             <xs:element ref="airsyncbase:IsInline" />
                                             <xs:element ref="email2:AttOrder" />
                                             <xs:element ref="email2:AttDuration" />
                                         </xs:all>
                                     </xs:complexType>
                                 </xs:element>
                             </xs:sequence>
                         </xs:complexType>
                        </xs:element>
                        <xs:element ref="email:BodyTruncated"/>
                        <xs:element ref="airsyncbase:Body" >
                         <xs:complexType>
                           <xs:sequence>
                             <xs:element ref="airsyncbase:Type" />
                             <xs:element ref="airsyncbase:EstimatedDataSize" />
                             <xs:element ref="airsyncbase:Truncated" />
                             <xs:element ref="airsyncbase:Data" />
```


Copyright © 2008 Microsoft Corporation.

```
        </xs:sequence>
      </xs:complexType>
  </xs:element>
     <xs:element ref="email:MessageClass"/>
     <xs:element ref="email:AttRemoved" />
     <xs:element ref="email:MeetingRequest" />
     <xs:element ref="email:InternetCPID" />
     <xs:element ref="email:Flag" />
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="tasks:Subject"./>
          <xs:element ref="email:Status" />
          <xs:element ref="email:FlagType" />
          <xs:element ref="tasks:ReminderSet" />
           <xs:element ref="tasks:ReminderTime" />
        </xs:sequence>
      </xs:complexType>
     </xs:element>
     <xs:element ref="email:ContentClass" />
     <xs:element ref="airsyncbase:NativeBodyType" />
     <xs:element ref="calendar:Timezone" />
     <xs:element ref="calendar:DtStamp" />
     <xs:element ref="calendar:StartTime"/>
     <xs:element ref="calendar:Subject" />
     <xs:element ref="calendar:UID" />
     <xs:element ref="calendar:OrganizerName" />
     <xs:element ref="calendar:OrganizerEmail" />
     <xs:element ref="calendar:Location" />
     <xs:element ref="calendar:EndTime" />
     <xs:element ref="calendar:Recurrence"/>
     <xs:element ref="calendar:Body" />
     <xs:element ref="calendar:Categories" />
     <xs:element ref="calendar:Sensitivity" />
     <xs:element ref="calendar:BusyStatus" />
     <xs:element ref="calendar:AllDayEvent" />
     <xs:element ref="calendar:Reminder" />
     <xs:element ref="calendar:Exceptions"/>
     <xs:element ref="calendar:MeetingStatus" />
     <xs:element ref="calendar:Rtf" />
     <xs:element ref="calendar:Attendees" />
     <xs:element ref="contacts:Rtf" />
     <xs:element ref="contacts:Anniversary"/>
     <xs:element ref="contacts:AssistantName" />
     <xs:element ref="contacts:AssistnamePhoneNumber" />
     <xs:element ref="contacts:Birthday" />
     <xs:element ref="contacts:Body" />
     <xs:element ref="contacts:BodySize" />
     <xs:element ref="contacts:BodyTruncated" />
     <xs:element ref="contacts:Business2PhoneNumber" />
     <xs:element ref="contacts:BusinessCity" />
     <xs:element ref="contacts:BusinessCountry" />
     <xs:element ref="contacts:BusinessPostalCode" />
     <xs:element ref="contacts:BusinessState" />
     <xs:element ref="contacts:BusinessStreet" />
     <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" />
```


Copyright © 2008 Microsoft Corporation.

```
<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:HomeCity" />
<xs:element ref="contacts:HomeCountry" />
<xs:element ref="contacts:HomePostalCode" />
<xs:element ref="contacts:HomeState" />
<xs:element ref="contacts:HomeStreet"/>
<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:OtherCity"/>
<xs:element ref="contacts:OtherCountry" />
<xs:element ref="contacts:OtherPostalCode" />
<xs:element ref="contacts:OtherState"/>
<xs:element ref="contacts:OtherStreet" />
<xs:element ref="contacts:PagerNumber"/>
<xs:element ref="contacts:picture" />
<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="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="tasks:Body" />
 <xs:element ref="tasks:BodySize" />
 <xs:element ref="tasks:BodyTruncated" />
 <xs:element ref="tasks:Subject" />
 <xs:element ref="tasks:Categories" />
 <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:Recurrence" />
 <xs:element ref="tasks:Complete " />
 <xs:element ref="tasks:DateCompleted" />
 <xs:element ref="tasks:Sensitivity" />
```


Copyright © 2008 Microsoft Corporation.

```
                    <xs:element ref="tasks:ReminderTime" />
                    <xs:element ref="tasks:ReminderSet" />
                    <xs:element ref="tasks:Rtf" />
                  </xs:choice>
                </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element minOccurs="0" maxOccurs="1" name="Responses">
        <xs:complexType>
        <xs:sequence>
          <xs:element minOccurs="0" maxOccurs="unbounded" name="Change">
            <xs:complexType>
            <xs:sequence>
              <xs:element name="ServerId" type="xs:string" />
              <xs:element name="Status" type="xs:unsignedByte" />
            </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element minOccurs="0" maxOccurs="unbounded" name="Add">
            <xs:complexType>
            <xs:sequence>
              <xs:element name="ClientId" type="xs:unsignedByte" />
              <xs:element name="ServerId" type="xs:string" />
              <xs:element name="Status" type="xs:unsignedByte" />
            </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="Fetch">
            <xs:complexType>
            <xs:sequence>
              <xs:element name="ServerId" type="xs:string"></xs:element>
              <xs:element name="Status" type="xs:unsignedByte"></xs:element>
              <xs:element name="ApplicationData">
                <xs:complexType>
                <xs:sequence>
                  <xs:choice maxOccurs="unbounded">
                  <xs:element ref="email:To" />
                  <xs:element ref="email:From" />
                  <xs:element ref="email:Reply-To" />
                  <xs:element ref="email:Subject" />
                  <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:Attachments" >
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element ref="airsyncbase:Attachment">
                                <xs:complexType>
                                    <xs:all>
<xs:element ref="airsyncbase:DisplayName" />
```


Copyright © 2008 Microsoft Corporation.

```
                                                    <xs:element
ref="airsyncbase:FileReference" />
                                                    <xs:element ref="airsyncbase:Method" />
                                                    <xs:element
ref="airsyncbase:EstimatedDataSize" />
                                                    <xs:element ref="airsyncbase:ContentId" />
                                                    <xs:element
ref="airsyncbase:ContentLocation" />
                                                    <xs:element ref="airsyncbase:IsInline" />
                                                </xs:all>
                                            </xs:complexType>
                                        </xs:element>
                                      </xs:sequence>
                                    </xs:complexType>
                                  </xs:element>
                                  <xs:element ref="email:BodyTruncated" />
                                  <xs:element ref="airsyncbase:Body" >
                                    <xs:complexType>
                                      <xs:sequence>
                                        <xs:element ref="airsyncbase:Type" />
                                        <xs:element ref="airsyncbase:EstimatedDataSize" />
                                        <xs:element ref="airsyncbase:Truncated" />
                                        <xs:element ref="airsyncbase:Data" />
                                      </xs:sequence>
                                    </xs:complexType>
                                </xs:element>
                                  <xs:element ref="email:MessageClass" />
                                  <xs:element ref="email:AttRemoved" />
                                  <xs:element ref="email:MeetingRequest" />
                                  <xs:element ref="email:InternetCPID" />
                                  <xs:element ref="email:Flag" />
                                    <xs:complexType>
                                      <xs:sequence>
                                        <xs:element ref="tasks:Subject"./>
                                        <xs:element ref="email:Status" />
                                        <xs:element ref="email:FlagType" />
                                        <xs:element ref="tasks:ReminderSet" />
                                         <xs:element ref="tasks:ReminderTime" />
                                      </xs:sequence>
                                    </xs:complexType>
                                   </xs:element>
                                   <xs:element ref="email:ContentClass" />
                                   <xs:element ref="airsyncbase:NativeBodyType" />
                                </xs:choice>
                              </xs:sequence>
                              <xs:complexType>
                            </xs:element>
                          </xs:sequence>
                          </xs:complexType>
                        </xs:element>
                       </xs:sequence>
                       </xs:complexType>
                     </xs:element>
                     <xs:element minOccurs="0" maxOccurs="1" name="MoreAvailable"></xs:element>
                   </xs:choice>
                 </xs:sequence>
                 </xs:complexType>
               </xs:element>
             </xs:sequence>
```


Copyright © 2008 Microsoft Corporation.

```
        </xs:complexType>
      </xs:element>
    </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

2.2.1.19.2.1 Add

The <Add> command can be used to create a new object in a collection on the client or on the server.

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 <ClientId> and the <ServerId> that was assigned to the new item.

When the client sends a **Sync** command 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 <ServerId> and data of the item to be added to the collection on the client.

Parent elements	Child elements	Data type	Number allowed
<Commands> <Responses> (response only)	<ServerId> (response only) <ClientId> <ApplicationData> <Status> (response only)	**Container**	0...N (optional)

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

If the <ServerId> in an <Add> element from the server matches the <ServerId> 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 command 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 command succeeded unless informed otherwise.

2.2.1.19.2.2 ApplicationData

The <ApplicationData> element encloses data for a particular object, such as a contact, e-mail message, calendar appointment, or task item. The <ApplicationData> element can be used to add or change items on the client device or server. The format of this data is determined by the schema for the object.

Parent elements	Child elements	Data type	Number allowed
<Add> <Change>	Data elements from the content classes. For details about the content classes, see [MS-ASCAL], [MS-ASCNTC], [MS-ASDOC], [MS-ASEMAIL], and [MS-ASTASK].	**Container**	1 (required)

Copyright © 2008 Microsoft Corporation.

The following <ApplicationData> element is used to add a contact item, identified by the <ServerId> element, to a folder on the client device.

Response

```
<Add>
          <ServerId> 2:6</ServerId>
          <ApplicationData>
<A:Body></A:Body>
             <A:Email1Address>"jdobney@fourthcoffee.com"
&lt;jdobney@fourthcoffee.com&gt;</A:Email1Address>
                <A:FileAs>Dobney, JoLynn Julie</A:FileAs>
                <A:FirstName>JoLynn</A:FirstName>
                <A:HomePhoneNumber>425 555 1234</A:HomePhoneNumber>
                <A:MiddleName>Julie</A:MiddleName>
                <A:MobilePhoneNumber>425 555 1111</A:MobilePhoneNumber>
                <A:CompanyName>Fourth Coffee</A:CompanyName>
                <A:LastName>Dobney</A:LastName>
                <A:BusinessPhoneNumber>425 555 5555</A:BusinessPhoneNumber>
                <A:JobTitle>Usability Engineer</A:JobTitle>
          </ApplicationData>
</Add>
```

2.2.1.19.2.3 Change

The <Change> element modifies properties of an existing object on the client device or the server. The object to change is identified by its <ServerId> element.

Parent elements	Child elements	Data type	Number allowed
<Commands> <Responses> (response only)	<ServerId> <ApplicationData> <Status> (response only)	**Container**	0...N (optional)

One or more <Change> elements can appear as a child of the <Commands> element for a particular collection.

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

- If there is only a <Flag> or <Read> change (that is, if only a <Flag> or <Read> node is present), all other properties will remain unchanged and nothing else has to be sent.
- If an <Exceptions> node is not specified, the properties for that <Exceptions> node will remain unchanged. If an <Exception> node within the <Exceptions> node is not present, that particular <Exception> will remain unchanged.
- If <Body>, <Data>, <Picture>, or <RTF> nodes 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.

Copyright © 2008 Microsoft Corporation.

2.2.1.19.2.4 ClientId

The <ClientId> is a unique identifier that is generated by the client to temporarily identify a new object that is being created by using the <Add> command. The client includes the <ClientId> element in the <Add> command 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.

Parent elements	Child elements	Data type	Number allowed
<Add>	None	**String** (Typically an integer)	Request: 1 (required) Response: 1

The <ClientId> element is a unique identifier that consists of up to 40 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.

2.2.1.19.2.5 Collection

The <Collection> element wraps commands and options that apply to a particular collection.

Parent elements	Child elements	Data type	Number allowed
<Collections>	<SyncKey> <Supported> (request only) <CollectionId> <DeletesAsMoves> (request only) <GetChanges> (request only) <WindowSize> (request only) <Options> (request only) <Status> (response only) <MoreAvailable> (response only) <Commands> <Responses> (response only)	**Container**	1..512 (required)

The <Collection> element contains identification information (<Class>, <CollectionID>), synchronization state (<SyncKey>), commands (<GetChanges>, <Commands>), and options (<WindowSize>, <Options>, <DeleteAsMoves>, <MoreAvailable>).

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

- <Class>
- <SyncKey>
- <CollectionId>
- <Supported>

Copyright © 2008 Microsoft Corporation.

- <DeletesAsMoves>
- <GetChanges>
- <WindowSize>
- <Options>
- <Commands>

The <Collection> element appears in both **Sync** requests and responses. The form is similar, although some child elements are valid in only one context.

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

2.2.1.19.2.6 CollectionId

The <CollectionId> element specifies the server ID of the folder to be synchronized.

Parent elements	Child elements	Data type	Number allowed
<Collection>	None	<String> (Up to 64 characters)	1 (required)

The server ID of the folder is obtained from the <ServerId> element of a previous **FolderSync** or **FolderCreate** command.

2.2.1.19.2.7 Collections

The <Collections> element serves as a container for the <Collection> element.

Parent elements	Child elements	Data type	Number allowed
<Sync>	<Collection>	**Container**	0...1 (optional)

The <Collections> element appears both in **Sync** requests and responses. The structure is identical.

The <Collections> element is optional. If <Collections> is present, it can contain multiple <Collection> elements, but MUST contain at least one.

Request/Response

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:">
<Collections>
    <Collection>
        ...
    </Collection>
</Collections>
```


Copyright © 2008 Microsoft Corporation.

2.2.1.19.2.8 Commands

The <Commands> element is a container for commands that apply to a collection. Available commands are <Add>, <Delete>, <Change>, <Fetch>, and <SoftDelete>. Client commands 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 command. It is a child of the <Collection> element.

Parent elements	Child elements	Data type	Number allowed
<Collection>	<Add> <Delete> <Change> <Fetch> <SoftDelete> (response only)	**Container**	0...1 (optional)

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

Request/Response

```
<Collection>
<Commands>
        <Add>
            …
        </Add>
        <Delete>
            …
        </Delete >
        <Change>
            …
        </Change >
        <Fetch>
            …
        </Fetch>
        <SoftDelete>
            …
        </SoftDelete>
    </Commands>
</Collection>
```

2.2.1.19.2.9 Delete

The <Delete> element deletes an object on the client or server. The object is identified by its <ServerId> element.

Parent elements	Child elements	Data type	Number allowed
<Commands>	<ServerId>	**Container**	0...N (optional)

2.2.1.19.2.10 SoftDelete

The <SoftDelete> element deletes an object when it falls outside the <FilterType> results. The object is identified by its <ServerId> element.

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
<Commands>	<ServerId>	**Container**	0...N (optional)

2.2.1.19.2.11 Fetch

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 2.2.1.8) is the preferred way to fetch items.

Parent elements	Child elements	Data type	Number allowed
<Commands> (request only) <Responses> (response only)	<ServerId> <Status> (response only) <ApplicationData> (response only)	**Container**	0...N (optional)

The <Fetch> element cannot be used to get truncated calendar, contact, or task items from the server.

2.2.1.19.2.12 Limit

The <Limit> element specifies either the maximum number of collections that can be synchronized or the maximum/minimum value that is allowed for the <Wait> or <HeartbeatInterval> interval.

Parent elements	Child elements	Data type	Number allowed
<Sync> (response only)	None	**Integer**	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 indicates whether the limit applies to the <Wait> or <HeartbeatInterval> interval or the number of collections, as follows:

- A status code 14 indicates that <Limit> specifies the minimum or maximum wait-interval that is acceptable. When the value of the <Wait> or <HeartbeatInterval> element is outside the acceptable range, the server responds with the closest acceptable value. The server will send a response that includes a status code 14 and a <Limit> element.
- A status code 15 indicates that <Limit> specifies the maximum number of collections that can be synchronized.

2.2.1.19.2.13 MoreAvailable

The <MoreAvailable> element is included in a synchronization response from the server to the client if there are more changes than the number that are requested in the <WindowSize> element.

Parent elements	Child elements	Data type	Number allowed
<Collection> (response only)	None	**Empty**	0...1 (optional)

The <MoreAvailable> element is an **Empty** element, meaning it has no value or data type. It is distinguished only by the presence or absence of the <MoreAvailable/> tag.

Copyright © 2008 Microsoft Corporation.

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 <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> 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.

2.2.1.19.2.14 Responses

The <Responses> element contains responses to commands that are processed by the server. Each response is wrapped in an element with the same name as the command, such as <Add> and <Delete>. The response contains a status code and other information, depending on the command.

Parent elements	Child elements	Data type	Number allowed
<Collection> (responses)	<Add>, <Fetch> (If the command succeeded.) <Change>, <Delete> (If the command failed.)	**Container**	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 commands 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.

The following <Responses> element is part of a server response to a synchronization request. It shows items in the server collection that have been added, deleted, changed, or fetched.

Response

```
<Collection>
    <Responses>
        <Add>
            ...
        </Add>
        <Change>
            ...
        </Change>
        <Fetch>
            ...
        </Fetch>
        <Delete>
            ...
        </Delete>
    </Responses>
</Collection>
```


Copyright © 2008 Microsoft Corporation.

2.2.1.19.2.15 ServerId

The <ServerId> is 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 and MUST be able to locate an object given a server ID. In a synchronization request, commands such as <Change> and <Delete> identify objects by using their server IDs.

Parent elements	Child elements	Data type	Number allowed
<Add> (response only) <Delete> <Change> <Fetch>	None	**String** (Up to 64 characters)	1 (required)

The client MUST store the server ID as an opaque string of up to 64 characters.

2.2.1.19.2.16 Status

The <Status> element indicates the success or failure of a command. If the command failed, the <Status> element contains a code that indicates the type of failure. The values are summarized in the following table.

Parent elements	Child elements	Data type	Number allowed
<Sync> (response only) <Collection> (response only) <Add> (response only) <Delete> (response only) <Change> (response only) <Fetch> (response only)	None	**Integer**	1 (required)

The following table lists valid values for the element.

Value	Meaning
1	Success.
2	Protocol version mismatch.
3	Invalid synchronization key.
4	Protocol error.
5	Server error.
6	Error in client/server conversion.
7	Conflict matching the client and server object.
8	Object not found.
9	User account could be out of disk space.
10	An error occurred while setting the notification GUID.

Copyright © 2008 Microsoft Corporation.

Value	Meaning
11	The device has not been provisioned for notifications yet.
12	The folder hierarchy has changed.
13	The server received an empty or partial **Sync** command request and the cached set of notifiable collections is missing. Resend the full **Sync** request.
14	The **Sync** request was processed successfully but the <Wait> or <HeartbeatInterval> interval that is specified by the client is outside the range set by the server administrator. If the interval is too great, the response contains a <Limit> element that specifies the maximum allowed value. If the interval is too low, the response contains a <Limit> element that specifies the minimum allowed value.
15	The **Sync** request was processed successfully, but specified more folders to monitor for changes than is allowed by the limit configured by the server administrator. The response includes the <Limit> element, which specifies the maximum number of folders that can be synchronized.
16	The server encountered a conflict. Please retry the same request.

The <Status> element is sent only in responses from the server to the client.

2.2.1.19.2.17 Sync

The <Sync> element is the top-level element in the XML document. It identifies the body of the HTTP **POST** as containing a <Sync> command.

Parent elements	Child elements	Data type	Number allowed
None	<Collections> <Partial> (request only) <Wait> (request only) <HeartbeatInterval> (request only) <WindowSize> (request only) <Limit> (response only) <Status> (response only)	**Container**	0...1 (optional)

The <Sync> element can also include one or more explicit namespace attributes.

The <Limit> element and <Collections> element are mutually exclusive in a **Sync** response. That is, a **Sync** response can include either a <Limit> element or a <Collections> element, but not both.

2.2.1.19.2.17.1 Empty Sync Response

The server sends a **Sync** response 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 2.2.1.19.1.1.1. For an example empty **Sync** request and response, see section 4.3.10.

Copyright © 2008 Microsoft Corporation.

2.2.1.19.2.18 SyncKey

The <SyncKey> element contains a value that is used by the server to represent the synchronization state of a collection.

Parent elements	Child elements	Data type	Number allowed
<Collection>	None	**String** (Up to 64 characters)	1 (required)

A synchronization key of value 0 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 uses 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 always sends 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.

2.2.1.19.3 Content Class Specific XSDs

The following sections contain the XSDs for content class specific **Sync** command requests and responses.

2.2.1.19.3.1 Sync Command for Calendar Items

2.2.1.19.3.1.1 Sync Command Request for Calendar Items

```
<?xml version="1.0" ?>
<xs:schema xmlns:calendar="Calendar:" attributeFormDefault="unqualified"
elementFormDefault="qualified"  xmlns:airsyncbase="AirSyncBase:"
targetNamespace="Calendar:" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="Calendar:">
  <xs:import namespace="AirSyncBase:"/>
  <xs:element name="TimeZone" type="xs:string" />
  <xs:element name="AllDayEvent" type="xs:unsignedByte" />
  <xs:element name="BusyStatus">
    <xs:simpleType>
      <xs:restriction base="xs:unsignedByte">
        <xs:minInclusive value="0"/>
        <xs:maxInclusive value="5"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="OrganizerName" type="xs:string" />
  <xs:element name="OrganizerEmail" type="xs:string" />
  <xs:element name="DtStamp" type="xs:string" />
  <xs:element name="EndTime" type="xs:string" />
  <xs:element name="Location" type="xs:string" />
  <xs:element name="Reminder" type="xs:unsignedInt" />
  <xs:element name="Sensitivity">
    <xs:simpleType>
```


Copyright © 2008 Microsoft Corporation.

```
      <xs:restriction base="xs:unsignedByte">
        <xs:minInclusive value="0"/>
        <xs:maxInclusive value="3"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="Subject" type="xs:string" />
  <xs:element name="StartTime" type="xs:string" />
  <xs:element name="UID">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="300"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="MeetingStatus">
    <xs:simpleType>
      <xs:restriction base="xs:unsignedByte">
        <xs:enumeration value="1"/>
        <xs:enumeration value="0"/>
        <xs:enumeration value="3"/>
        <xs:enumeration value="5"/>
        <xs:enumeration value="7"/>
        <xs:enumeration value="9"/>
        <xs:enumeration value="11"/>
        <xs:enumeration value="13"/>
        <xs:enumeration value="15"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="Attendees">
    <xs:complexType>
      <xs:sequence minOccurs="0">
        <xs:element name="Attendee" maxOccurs="unbounded">
          <xs:complexType>
            <xs:all>
              <xs:element name="Email" type="xs:string" />
              <xs:element name="Name" type="xs:string" />
              <xs:element name="AttendeeStatus" minOccurs="0">
                <xs:simpleType>
                  <xs:restriction base="xs:unsignedByte">
                    <xs:enumeration value="0"/>
                    <xs:enumeration value="2"/>
                    <xs:enumeration value="3"/>
                    <xs:enumeration value="4"/>
                    <xs:enumeration value="5"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
              <xs:element name="AttendeeType" minOccurs="0">
                <xs:simpleType>
                  <xs:restriction base="xs:unsignedByte">
                    <xs:enumeration value="1"/>
                    <xs:enumeration value="2"/>
                    <xs:enumeration value="3"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
            </xs:all>
```


Copyright © 2008 Microsoft Corporation.

```
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Categories">
    <xs:complexType>
      <xs:sequence minOccurs="0">
        <xs:element maxOccurs="300" name="Category" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="DisallowNewTimeProposal" type="xs:boolean" />
  <xs:element name="ResponseRequested" type="xs:boolean" />
  <xs:element name="Recurrence">
    <xs:complexType>
      <xs:all minOccurs="0">
        <xs:element minOccurs="1" name="Type">
          <xs:simpleType>
            <xs:restriction base="xs:unsignedByte">
              <xs:minInclusive value="0"/>
              <xs:maxInclusive value="6"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element minOccurs="0" name="Occurrences" type="xs:unsignedShort" />
        <xs:element minOccurs="0" name="Interval">
          <xs:simpleType>
            <xs:restriction base="xs:unsignedShort">
              <xs:minInclusive value="0"/>
              <xs:maxInclusive value="999"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element minOccurs="0" name="WeekOfMonth">
          <xs:simpleType>
            <xs:restriction base="xs:unsignedByte">
              <xs:minInclusive value="1"/>
              <xs:maxInclusive value="5"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element minOccurs="0" name="DayOfWeek">
          <xs:simpleType>
            <xs:restriction base="xs:unsignedShort">
              <xs:minInclusive value="0"/>
              <xs:maxInclusive value="127"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element minOccurs="0" name="MonthOfYear">
          <xs:simpleType>
            <xs:restriction base="xs:unsignedByte">
              <xs:minInclusive value="1"/>
              <xs:maxInclusive value="12"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element minOccurs="0" name="Until" type="xs:string" />
```


Copyright © 2008 Microsoft Corporation.

```
        <xs:element minOccurs="0" name="DayOfMonth">
          <xs:simpleType>
            <xs:restriction base="xs:unsignedByte">
              <xs:minInclusive value="1"/>
              <xs:maxInclusive value="127"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element minOccurs="0" name="CalendarType">
            <xs:simpleType>
                <xs:restriction base="xs:unsignedByte">
                    <xs:minInclusive value="0"/>
                    <xs:maxInclusive value="23"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
        <xs:element minOccurs="0" name="IsLeapMonth">
            <xs:simpleType>
                <xs:restriction base="xs:unsignedByte">
                    <xs:minInclusive value="0"/>
                    <xs:maxInclusive value="1"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:element>
      </xs:all>
    </xs:complexType>
  </xs:element>
  <xs:element name="Exceptions">
    <xs:complexType>
      <xs:sequence minOccurs="0">
        <xs:element name="Exception" maxOccurs="1000">
          <xs:complexType>
            <xs:all>
              <xs:element minOccurs="0" name="Deleted" type="xs:unsignedByte" />
              <xs:element name="ExceptionStartTime" type="xs:string" />
              <xs:element minOccurs="0" name="Subject" type="xs:string" />
              <xs:element minOccurs="0" name="StartTime" type="xs:string" />
              <xs:element minOccurs="0" name="EndTime" type="xs:string" />
              <xs:element minOccurs="0" ref="airsyncbase:Body" />
              <xs:element minOccurs="0" name="Location" type="xs:string" />
              <xs:element minOccurs="0" name="Categories">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element maxOccurs="300" name="Category" type="xs:string" />
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element minOccurs="0" name="Sensitivity">
                <xs:simpleType>
                  <xs:restriction base="xs:unsignedByte">
                    <xs:minInclusive value="0"/>
                    <xs:maxInclusive value="3"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
              <xs:element minOccurs="0" name="BusyStatus">
                <xs:simpleType>
                  <xs:restriction base="xs:unsignedByte">
                    <xs:minInclusive value="0"/>
```


Copyright © 2008 Microsoft Corporation.

```
            <xs:maxInclusive value="5"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element minOccurs="0" name="AllDayEvent" type="xs:unsignedByte" />
      <xs:element minOccurs="0" name="Reminder" type="xs:unsignedInt" />
      <xs:element minOccurs="0" name="DtStamp" type="xs:string" />
      <xs:element minOccurs="0" name="MeetingStatus">
        <xs:simpleType>
          <xs:restriction base="xs:unsignedByte">
            <xs:enumeration value="1"/>
            <xs:enumeration value="0"/>
            <xs:enumeration value="3"/>
            <xs:enumeration value="5"/>
            <xs:enumeration value="7"/>
            <xs:enumeration value="9"/>
            <xs:enumeration value="11"/>
            <xs:enumeration value="13"/>
            <xs:enumeration value="15"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element minOccurs="0" name="Attendees">
          <xs:complexType>
              <xs:sequence minOccurs="0">
                  <xs:element name="Attendee" maxOccurs="unbounded">
                      <xs:complexType>
                          <xs:all>
                              <xs:element name="Email" type="xs:string" />
                              <xs:element name="Name" type="xs:string" />
                              <xs:element name="AttendeeStatus" minOccurs="0">
                                <xs:simpleType>
                                  <xs:restriction base="xs:unsignedByte">
                                    <xs:enumeration value="0"/>
                                    <xs:enumeration value="2"/>
                                    <xs:enumeration value="3"/>
                                    <xs:enumeration value="4"/>
                                    <xs:enumeration value="5"/>
                                  </xs:restriction>
                                </xs:simpleType>
                              </xs:element>
                              <xs:element name="AttendeeType" minOccurs="0">
                                <xs:simpleType>
                                  <xs:restriction base="xs:unsignedByte">
                                    <xs:enumeration value="1"/>
                                    <xs:enumeration value="2"/>
                                    <xs:enumeration value="3"/>
                                  </xs:restriction>
                                </xs:simpleType>
                              </xs:element>
                          </xs:all>
                      </xs:complexType>
                  </xs:element>
              </xs:sequence>
          </xs:complexType>
      </xs:element>
    </xs:all>
  </xs:complexType>
</xs:element>
```


Copyright © 2008 Microsoft Corporation.

```
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="EmptyTag"/>
  <xs:group name="GhostingProps">
    <xs:sequence>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="TimeZone" type="calendar:EmptyTag" />
        <xs:element name="AllDayEvent" type="calendar:EmptyTag" />
        <xs:element name="BusyStatus" type="calendar:EmptyTag" />
        <xs:element name="OrganizerName" type="calendar:EmptyTag" />
        <xs:element name="OrganizerEmail" type="calendar:EmptyTag" />
        <xs:element name="DtStamp" type="calendar:EmptyTag" />
        <xs:element name="EndTime" type="calendar:EmptyTag" />
        <xs:element name="Location" type="calendar:EmptyTag" />
        <xs:element name="Reminder" type="calendar:EmptyTag" />
        <xs:element name="Sensitivity" type="calendar:EmptyTag" />
        <xs:element name="Subject" type="calendar:EmptyTag" />
        <xs:element name="StartTime" type="calendar:EmptyTag" />
        <xs:element name="UID" type="calendar:EmptyTag" />
        <xs:element name="MeetingStatus" type="calendar:EmptyTag" />
        <xs:element name="Attendees" type="calendar:EmptyTag" />
        <xs:element name="Categories" type="calendar:EmptyTag" />
        <xs:element name="Recurrence" type="calendar:EmptyTag" />
        <xs:element name="Exceptions" type="calendar:EmptyTag" />
        <xs:element name="DisallowNewTimeProposal" type="calendar:EmptyTag" />
        <xs:element name="ResponseRequested" type="calendar:EmptyTag" />
      </xs:choice>
    </xs:sequence>
  </xs:group>
  <xs:group name="TopLevelSchemaProps">
    <xs:sequence>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="TimeZone" type="calendar:EmptyTag" />
        <xs:element name="StartTime" type="calendar:EmptyTag" />
        <xs:element name="EndTime" type="calendar:EmptyTag" />
        <xs:element name="Subject" type="calendar:EmptyTag" />
        <xs:element name="Location" type="calendar:EmptyTag" />
        <xs:element name="Reminder" type="calendar:EmptyTag" />
        <xs:element name="AllDayEvent" type="calendar:EmptyTag" />
        <xs:element name="BusyStatus" type="calendar:EmptyTag" />
        <xs:element name="Recurrence" type="calendar:EmptyTag" />
        <xs:element name="Sensitivity" type="calendar:EmptyTag" />
        <xs:element name="DtStamp" type="calendar:EmptyTag" />
        <xs:element name="Attendees" type="calendar:EmptyTag" />
        <xs:element name="Categories" type="calendar:EmptyTag" />
        <xs:element name="MeetingStatus" type="calendar:EmptyTag" />
        <xs:element name="OrganizerName" type="calendar:EmptyTag" />
        <xs:element name="OrganizerEmail" type="calendar:EmptyTag" />
        <xs:element name="UID" type="calendar:EmptyTag" />
        <xs:element name="DisallowNewTimeProposal" type="calendar:EmptyTag" />
        <xs:element name="ResponseRequested" type="calendar:EmptyTag" />
        <xs:element name="Exceptions" type="calendar:EmptyTag" />
      </xs:choice>
    </xs:sequence>
  </xs:group>
</xs:schema>
```


Copyright © 2008 Microsoft Corporation.

2.2.1.19.3.1.2 Sync Command Response for Calendar Items

For the complete **Sync** command response, see section 2.2.1.19.2.

2.2.1.19.3.2 Sync Command for Contacts Folder

2.2.1.19.3.2.1 Sync Command Request for Contacts

```
<?xml version="1.0" ?>
<xs:schema xmlns:contacts="Contacts:" attributeFormDefault="unqualified"
elementFormDefault="qualified"
targetNamespace="Contacts:" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Anniversary" type="xs:dateTime" />
  <xs:element name="AssistantName" type="xs:string" />
  <xs:element name="AssistantPhoneNumber" type="xs:string" />
  <xs:element name="AssistnamePhoneNumber" type="xs:string" />
  <xs:element name="Birthday" type="xs:dateTime" />
  <xs:element name="Business2PhoneNumber" type="xs:string" />
  <xs:element name="BusinessAddressCity" type="xs:string" />
  <xs:element name="BusinessPhoneNumber" type="xs:string" />
  <xs:element name="WebPage" type="xs:string" />
  <xs:element name="BusinessAddressCountry" type="xs:string" />
  <xs:element name="Department" type="xs:string" />
  <xs:element name="Email1Address" type="xs:string" />
  <xs:element name="Email2Address" type="xs:string" />
  <xs:element name="Email3Address" type="xs:string" />
  <xs:element name="BusinessFaxNumber" type="xs:string" />
  <xs:element name="FileAs" type="xs:string" />
  <xs:element name="FirstName" type="xs:string" />
  <xs:element name="MiddleName" type="xs:string" />
  <xs:element name="HomeAddressCity" type="xs:string" />
  <xs:element name="HomeAddressCountry" type="xs:string" />
  <xs:element name="HomeFaxNumber" type="xs:string" />
  <xs:element name="HomePhoneNumber" type="xs:string" />
  <xs:element name="Home2PhoneNumber" type="xs:string" />
  <xs:element name="HomeAddressPostalCode" type="xs:string" />
  <xs:element name="HomeAddressState" type="xs:string" />
  <xs:element name="HomeAddressStreet" type="xs:string" />
  <xs:element name="MobilePhoneNumber" type="xs:string" />
  <xs:element name="Suffix" type="xs:string" />
  <xs:element name="CompanyName" type="xs:string" />
  <xs:element name="OtherAddressCity" type="xs:string" />
  <xs:element name="OtherAddressCountry" type="xs:string" />
  <xs:element name="CarPhoneNumber" type="xs:string" />
  <xs:element name="OtherAddressPostalCode" type="xs:string" />
  <xs:element name="OtherAddressState" type="xs:string" />
  <xs:element name="OtherAddressStreet" type="xs:string" />
  <xs:element name="PagerNumber" type="xs:string" />
  <xs:element name="Title" type="xs:string" />
  <xs:element name="BusinessAddressPostalCode" type="xs:string" />
  <xs:element name="LastName" type="xs:string" />
  <xs:element name="Spouse" type="xs:string" />
  <xs:element name="BusinessAddressState" type="xs:string" />
  <xs:element name="BusinessAddressStreet" type="xs:string" />
  <xs:element name="JobTitle" type="xs:string" />
  <xs:element name="YomiFirstName" type="xs:string" />
  <xs:element name="YomiLastName" type="xs:string" />
  <xs:element name="YomiCompanyName" type="xs:string" />
```


Copyright © 2008 Microsoft Corporation.

```
<xs:element name="OfficeLocation" type="xs:string" />
<xs:element name="RadioPhoneNumber" type="xs:string" />
<xs:element name="Picture" type="xs:string" />
<xs:element name="Categories">
  <xs:complexType>
    <xs:sequence minOccurs="0">
      <xs:element maxOccurs="300" name="Category" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Children">
  <xs:complexType>
    <xs:sequence minOccurs="0">
      <xs:element maxOccurs="300" name="Child" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType name="EmptyTag"/>
<xs:group name="GhostingProps">
  <xs:sequence>
    <xs:choice maxOccurs="unbounded">
      <xs:element name="Anniversary" type="contacts:EmptyTag"/>
      <xs:element name="Birthday" type="contacts:EmptyTag"/>
      <xs:element name="WebPage" type="contacts:EmptyTag"/>
      <xs:element name="Children" type="contacts:EmptyTag"/>
      <xs:element name="BusinessAddressCountry" type="contacts:EmptyTag"/>
      <xs:element name="Department" type="contacts:EmptyTag"/>
      <xs:element name="Email1Address" type="contacts:EmptyTag"/>
      <xs:element name="Email2Address" type="contacts:EmptyTag"/>
      <xs:element name="Email3Address" type="contacts:EmptyTag"/>
      <xs:element name="BusinessFaxNumber" type="contacts:EmptyTag"/>
      <xs:element name="FileAs" type="contacts:EmptyTag"/>
      <xs:element name="FirstName" type="contacts:EmptyTag"/>
      <xs:element name="HomeAddressCity" type="contacts:EmptyTag"/>
      <xs:element name="HomeAddressCountry" type="contacts:EmptyTag"/>
      <xs:element name="HomeFaxNumber" type="contacts:EmptyTag"/>
      <xs:element name="HomePhoneNumber" type="contacts:EmptyTag"/>
      <xs:element name="Home2PhoneNumber" type="contacts:EmptyTag"/>
      <xs:element name="HomeAddressPostalCode" type="contacts:EmptyTag"/>
      <xs:element name="HomeAddressState" type="contacts:EmptyTag"/>
      <xs:element name="HomeAddressStreet" type="contacts:EmptyTag"/>
      <xs:element name="BusinessAddressCity" type="contacts:EmptyTag"/>
      <xs:element name="MiddleName" type="contacts:EmptyTag"/>
      <xs:element name="MobilePhoneNumber" type="contacts:EmptyTag"/>
      <xs:element name="Suffix" type="contacts:EmptyTag"/>
      <xs:element name="CompanyName" type="contacts:EmptyTag"/>
      <xs:element name="OtherAddressCity" type="contacts:EmptyTag"/>
      <xs:element name="OtherAddressCountry" type="contacts:EmptyTag"/>
      <xs:element name="CarPhoneNumber" type="contacts:EmptyTag"/>
      <xs:element name="OtherAddressPostalCode" type="contacts:EmptyTag"/>
      <xs:element name="OtherAddressState" type="contacts:EmptyTag"/>
      <xs:element name="OtherAddressStreet" type="contacts:EmptyTag"/>
      <xs:element name="PagerNumber" type="contacts:EmptyTag"/>
      <xs:element name="Title" type="contacts:EmptyTag"/>
      <xs:element name="BusinessAddressPostalCode" type="contacts:EmptyTag"/>
      <xs:element name="AssistantName" type="contacts:EmptyTag"/>
      <xs:element name="AssistantPhoneNumber" type="contacts:EmptyTag"/>
      <xs:element name="AssistnamePhoneNumber" type="contacts:EmptyTag"/>
      <xs:element name="LastName" type="contacts:EmptyTag"/>
```


Copyright © 2008 Microsoft Corporation.

```
        <xs:element name="Spouse" type="contacts:EmptyTag"/>
        <xs:element name="BusinessAddressState" type="contacts:EmptyTag"/>
        <xs:element name="BusinessAddressStreet" type="contacts:EmptyTag"/>
        <xs:element name="BusinessPhoneNumber" type="contacts:EmptyTag"/>
        <xs:element name="Business2PhoneNumber" type="contacts:EmptyTag"/>
        <xs:element name="JobTitle" type="contacts:EmptyTag"/>
        <xs:element name="YomiFirstName" type="contacts:EmptyTag"/>
        <xs:element name="YomiLastName" type="contacts:EmptyTag"/>
        <xs:element name="YomiCompanyName" type="contacts:EmptyTag"/>
        <xs:element name="OfficeLocation" type="contacts:EmptyTag"/>
        <xs:element name="RadioPhoneNumber" type="contacts:EmptyTag"/>
        <xs:element name="Picture" type="contacts:EmptyTag"/>
        <xs:element name="Categories" type="contacts:EmptyTag"/>
      </xs:choice>
    </xs:sequence>
  </xs:group>
  <xs:group name="TopLevelSchemaProps">
    <xs:sequence>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="Anniversary" type="contacts:EmptyTag"/>
        <xs:element name="Birthday" type="contacts:EmptyTag"/>
        <xs:element name="Webpage" type="contacts:EmptyTag"/>
        <xs:element name="Children" type="contacts:EmptyTag"/>
        <xs:element name="BusinessAddressCountry" type="contacts:EmptyTag"/>
        <xs:element name="Department" type="contacts:EmptyTag"/>
        <xs:element name="Email1Address" type="contacts:EmptyTag"/>
        <xs:element name="Email2Address" type="contacts:EmptyTag"/>
        <xs:element name="Email3Address" type="contacts:EmptyTag"/>
        <xs:element name="BusinessFaxNumber" type="contacts:EmptyTag"/>
        <xs:element name="FileAs" type="contacts:EmptyTag"/>
        <xs:element name="FirstName" type="contacts:EmptyTag"/>
        <xs:element name="HomeAddressCity" type="contacts:EmptyTag"/>
        <xs:element name="HomeAddressCountry" type="contacts:EmptyTag"/>
        <xs:element name="HomeFaxNumber" type="contacts:EmptyTag"/>
        <xs:element name="HomeTelephoneNumber" type="contacts:EmptyTag"/>
        <xs:element name="Home2TelephoneNumber" type="contacts:EmptyTag"/>
        <xs:element name="HomeAddressPostalCode" type="contacts:EmptyTag"/>
        <xs:element name="HomeAddressState" type="contacts:EmptyTag"/>
        <xs:element name="HomeAddressStreet" type="contacts:EmptyTag"/>
        <xs:element name="BusinessAddressCity" type="contacts:EmptyTag"/>
        <xs:element name="MiddleName" type="contacts:EmptyTag"/>
        <xs:element name="MobileTelephoneNumber" type="contacts:EmptyTag"/>
        <xs:element name="Suffix" type="contacts:EmptyTag"/>
        <xs:element name="CompanyName" type="contacts:EmptyTag"/>
        <xs:element name="OtherAddressCity" type="contacts:EmptyTag"/>
        <xs:element name="OtherAddressCountry" type="contacts:EmptyTag"/>
        <xs:element name="CarTelephoneNumber" type="contacts:EmptyTag"/>
        <xs:element name="OtherAddressPostalCode" type="contacts:EmptyTag"/>
        <xs:element name="OtherAddressState" type="contacts:EmptyTag"/>
        <xs:element name="OtherAddressStreet" type="contacts:EmptyTag"/>
        <xs:element name="PagerNumber" type="contacts:EmptyTag"/>
        <xs:element name="Title" type="contacts:EmptyTag"/>
        <xs:element name="BusinessAddressPostalCode" type="contacts:EmptyTag"/>
        <xs:element name="AssistantName" type="contacts:EmptyTag"/>
        <xs:element name="AssistantTelephoneNumber" type="contacts:EmptyTag"/>
        <xs:element name="LastName" type="contacts:EmptyTag"/>
        <xs:element name="Spouse" type="contacts:EmptyTag"/>
        <xs:element name="BusinessAddressState" type="contacts:EmptyTag"/>
        <xs:element name="BusinessAddressStreet" type="contacts:EmptyTag"/>
```


Copyright © 2008 Microsoft Corporation.

```
        <xs:element name="BusinessTelephoneNumber" type="contacts:EmptyTag"/>
        <xs:element name="Business2TelephoneNumber" type="contacts:EmptyTag"/>
        <xs:element name="JobTitle" type="contacts:EmptyTag"/>
        <xs:element name="YomiFirstName" type="contacts:EmptyTag"/>
        <xs:element name="YomiLastName" type="contacts:EmptyTag"/>
        <xs:element name="YomiCompanyName" type="contacts:EmptyTag"/>
        <xs:element name="OfficeLocation" type="contacts:EmptyTag"/>
        <xs:element name="RadioTelephoneNumber" type="contacts:EmptyTag"/>
        <xs:element name="Categories" type="contacts:EmptyTag"/>
        <xs:element name="Picture" type="contacts:EmptyTag"/>
      </xs:choice>
    </xs:sequence>
  </xs:group>
</xs:schema>
```

2.2.1.19.3.2.2 Sync Command Response for Contacts

For the complete **Sync** command response, see section 2.2.1.19.2.

2.2.1.19.3.3 Sync Command for Contacts2 Folder

2.2.1.19.3.3.1 Sync Command Request for Contacts2

```
<?xml version="1.0" ?>
<xs:schema xmlns:contacts2="Contacts2:" attributeFormDefault="unqualified"
elementFormDefault="qualified"
targetNamespace="Contacts2:" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="CustomerId" type="xs:string" />
  <xs:element name="GovernmentId" type="xs:string" />
  <xs:element name="IMAddress" type="xs:string" />
  <xs:element name="IMAddress2" type="xs:string" />
  <xs:element name="IMAddress3" type="xs:string" />
  <xs:element name="ManagerName" type="xs:string" />
  <xs:element name="CompanyMainPhone" type="xs:string" />
  <xs:element name="AccountName" type="xs:string" />
  <xs:element name="NickName" type="xs:string" />
  <xs:element name="MMS" type="xs:string" />
  <xs:complexType name="EmptyTag"/>
  <xs:group name="GhostingProps">
    <xs:sequence>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="CustomerId" type="contacts2:EmptyTag"/>
        <xs:element name="GovernmentId" type="contacts2:EmptyTag"/>
        <xs:element name="IMAddress" type="contacts2:EmptyTag"/>
        <xs:element name="IMAddress2" type="contacts2:EmptyTag"/>
        <xs:element name="IMAddress3" type="contacts2:EmptyTag"/>
        <xs:element name="ManagerName" type="contacts2:EmptyTag"/>
        <xs:element name="CompanyMainPhone" type="contacts2:EmptyTag"/>
        <xs:element name="AccountName" type="contacts2:EmptyTag"/>
        <xs:element name="NickName" type="contacts2:EmptyTag"/>
        <xs:element name="MMS" type="contacts2:EmptyTag"/>
      </xs:choice>
    </xs:sequence>
  </xs:group>
  <xs:group name="TopLevelSchemaProps">
    <xs:sequence>
      <xs:choice maxOccurs="unbounded">
```


Copyright © 2008 Microsoft Corporation.

```
        <xs:element name="CustomerId" type="contacts2:EmptyTag"/>
        <xs:element name="GovernmentId" type="contacts2:EmptyTag"/>
        <xs:element name="IMAddress" type="contacts2:EmptyTag"/>
        <xs:element name="IMAddress2" type="contacts2:EmptyTag"/>
        <xs:element name="IMAddress3" type="contacts2:EmptyTag"/>
        <xs:element name="ManagerName" type="contacts2:EmptyTag"/>
        <xs:element name="CompanyMainPhone" type="contacts2:EmptyTag"/>
        <xs:element name="AccountName" type="contacts2:EmptyTag"/>
        <xs:element name="NickName" type="contacts2:EmptyTag"/>
        <xs:element name="MMS" type="contacts2:EmptyTag"/>
      </xs:choice>
    </xs:sequence>
  </xs:group>
</xs:schema>
```

2.2.1.19.3.3.2 Sync Command Response for Contacts2

For the complete **Sync** command response, see section 2.2.1.19.2.

2.2.1.19.3.4 Sync Command for E-Mail Folder

2.2.1.19.3.4.1 Sync Command Request for E-Mail

```
<?xml version="1.0" ?>
<xs:schema xmlns:email="Email:" attributeFormDefault="unqualified"
elementFormDefault="qualified"
targetNamespace="Email:" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:tasks="Tasks:">
  <xs:import namespace="Tasks:"/>
  <xs:element name="Read" type="xs:unsignedByte" />
  <xs:element name="DateReceived">
      <xs:simpleType>
          <xs:restriction base="xs:string">
              <xs:maxLength value="1024"/>
          </xs:restriction>
      </xs:simpleType>
  </xs:element>
  <xs:element name="Flag">
    <xs:complexType>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
          <xs:element name="Status" maxOccurs="1" type="xs:unsignedByte" />
          <xs:element name="FlagType" maxOccurs="1" type="xs:string" />
                      <xs:element name="CompleteTime" maxOccurs="1" type="xs:string" />
          <xs:element ref="tasks:StartDate" maxOccurs="1"/>
          <xs:element ref="tasks:UtcStartDate" maxOccurs="1"/>
          <xs:element ref="tasks:DueDate" maxOccurs="1"/>
          <xs:element ref="tasks:UtcDueDate" maxOccurs="1"/>
                      <xs:element ref="tasks:DateCompleted" maxOccurs="1"/>
                      <xs:element ref="tasks:ReminderSet" maxOccurs="1"/>
                      <xs:element ref="tasks:ReminderTime" maxOccurs="1"/>
          <xs:element ref="tasks:Subject" maxOccurs="1"/>
          <xs:element ref="tasks:OrdinalDate" maxOccurs="1"/>
          <xs:element ref="tasks:SubOrdinalDate" maxOccurs="1"/>
      </xs:choice>
    </xs:complexType>
  </xs:element>
  <xs:element name="To">
      <xs:simpleType>
```


Copyright © 2008 Microsoft Corporation.

```
        <xs:restriction base="xs:string">
          <xs:maxLength value="1024"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="From">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="1024"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="InternetCPID" type="xs:integer" />
    <xs:element name="Importance" type="xs:unsignedByte" />
    <xs:element name="Categories">
      <xs:complexType>
        <xs:sequence minOccurs="0">
            <xs:element maxOccurs="300" name="Category" type="xs:string" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:complexType name="EmptyTag"/>
    <xs:group name="GhostingProps">
      <xs:sequence>
        <xs:choice maxOccurs="unbounded">
            <xs:element name="To" type="email:EmptyTag" />
            <xs:element name="From" type="email:EmptyTag" />
            <xs:element name="DateReceived" type="email:EmptyTag" />
            <xs:element name="Body" type="email:EmptyTag" />
            <xs:element name="InternetCPID" type="email:EmptyTag" />
            <xs:element name="Read" type="email:EmptyTag" />
            <xs:element name="Flag" type="email:EmptyTag" />
            <xs:element name="Importance" type="email:EmptyTag" />
        </xs:choice>
      </xs:sequence>
    </xs:group>
    <xs:group name="TopLevelSchemaProps">
      <xs:sequence>
        <xs:choice maxOccurs="unbounded">
          <xs:element name="To" type="email:EmptyTag"/>
          <xs:element name="CC" type="email:EmptyTag"/>
          <xs:element name="From" type="email:EmptyTag"/>
          <xs:element name="ReplyTo" type="email:EmptyTag"/>
          <xs:element name="DateReceived" type="email:EmptyTag"/>
          <xs:element name="Subject" type="email:EmptyTag"/>
          <xs:element name="DisplayTo" type="email:EmptyTag"/>
          <xs:element name="Importance" type="email:EmptyTag"/>
          <xs:element name="Read" type="email:EmptyTag"/>
          <xs:element name="MessageClass" type="email:EmptyTag"/>
          <xs:element name="MeetingRequest" type="email:EmptyTag"/>
          <xs:element name="ThreadTopic" type="email:EmptyTag"/>
          <xs:element name="InternetCPID" type="email:EmptyTag"/>
        </xs:choice>
      </xs:sequence>
    </xs:group>
  </xs:schema>
```


Copyright © 2008 Microsoft Corporation.

2.2.1.19.3.4.2 Sync Command Response for E-Mail

For the complete **Sync** command response, see section 2.2.1.19.2.

2.2.1.19.3.5 Sync Command for Tasks Folder

2.2.1.19.3.5.1 Sync Command Request for Tasks

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
        xmlns:tns="Tasks:"
        attributeFormDefault="unqualified"
        elementFormDefault="qualified"
        targetNamespace="Tasks:"
        xmlns:xs="http://www.w3.org/2001/XMLSchema"
        xmlns="Tasks:">

    <xs:element name="Subject" type="xs:string"/>
    <xs:element name="Categories">
        <xs:complexType>
            <xs:sequence minOccurs="0">
                <xs:element maxOccurs="300" name="Category" type="xs:string" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="Complete" type="xs:unsignedByte"/>
    <xs:element name="DateCompleted" type="xs:string"/>
    <xs:element name="DueDate" type="xs:string"/>
    <xs:element name="UtcDueDate" type="xs:string"/>
    <xs:element name="Importance" type="xs:unsignedByte"/>
    <xs:element name="Recurrence">
        <xs:complexType>
            <xs:all minOccurs="0">
                <xs:element minOccurs="1" name="Type">
                    <xs:simpleType>
                        <xs:restriction base="xs:unsignedByte">
                            <xs:minInclusive value="0"/>
                            <xs:maxInclusive value="6"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element minOccurs="0" name="Start" type="xs:string" />
                <xs:element minOccurs="0" name="Until" type="xs:string" />
                <xs:element minOccurs="0" name="Occurrences" type="xs:unsignedShort" />
                <xs:element minOccurs="0" name="Interval">
                    <xs:simpleType>
                        <xs:restriction base="xs:unsignedShort">
                            <xs:minInclusive value="0"/>
                            <xs:maxInclusive value="999"/>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element minOccurs="0" name="DayOfWeek">
                    <xs:simpleType>
                        <xs:restriction base="xs:unsignedShort">
                            <xs:minInclusive value="0"/>
                            <xs:maxInclusive value="127"/>
                        </xs:restriction>
```


Copyright © 2008 Microsoft Corporation.

```
                </xs:simpleType>
            </xs:element>
            <xs:element minOccurs="0" name="DayOfMonth">
                <xs:simpleType>
                    <xs:restriction base="xs:unsignedByte">
                        <xs:minInclusive value="1"/>
                        <xs:maxInclusive value="127"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element minOccurs="0" name="WeekOfMonth">
                <xs:simpleType>
                    <xs:restriction base="xs:unsignedByte">
                        <xs:minInclusive value="1"/>
                        <xs:maxInclusive value="5"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element minOccurs="0" name="MonthOfYear">
                <xs:simpleType>
                    <xs:restriction base="xs:unsignedByte">
                        <xs:minInclusive value="1"/>
                        <xs:maxInclusive value="12"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element minOccurs="0" name="Regenerate" type="xs:unsignedByte"/>
            <xs:element minOccurs="0" name="DeadOccur" type="xs:unsignedByte"/>
            <xs:element minOccurs="0" name="CalendarType">
                <xs:simpleType>
                    <xs:restriction base="xs:unsignedByte">
                        <xs:minInclusive value="0"/>
                        <xs:maxInclusive value="23"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element minOccurs="0" name="IsLeapMonth">
                <xs:simpleType>
                    <xs:restriction base="xs:unsignedByte">
                        <xs:minInclusive value="0"/>
                        <xs:maxInclusive value="1"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
        </xs:all>
    </xs:complexType>
</xs:element>
<xs:element name="ReminderSet" type="xs:unsignedByte"/>
<xs:element name="ReminderTime" type="xs:string"/>
<xs:element name="Sensitivity">
    <xs:simpleType>
        <xs:restriction base="xs:unsignedByte">
            <xs:minInclusive value="0"/>
            <xs:maxInclusive value="3"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="StartDate" type="xs:string"/>
<xs:element name="UtcStartDate" type="xs:string"/>
```


Copyright © 2008 Microsoft Corporation.

```
    <xs:element name="OrdinalDate" type="xs:string"/>
    <xs:element name="SubOrdinalDate" type="xs:string"/>
</xs:schema>
```

2.2.1.19.3.5.2 Sync Command Response for Tasks

For the complete **Sync** command response, see section 2.2.1.19.2.

2.2.1.20 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 **certificate authority**. Certificate validation is particularly important for verifying signatures (for example, on S/MIME signed mail).

2.2.1.20.1 Request

The following code shows the XSD for the **ValidateCert** command response.

```
<?xml version="1.0" ?>
<xs:schema xmlns:tns="ValidateCert:" attributeFormDefault="unqualified"
elementFormDefault="qualified"
targetNamespace="ValidateCert:" xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="ValidateCert">
      <xs:complexType>
         <xs:all minOccurs="0" maxOccurs="1">
            <xs:element name="CertificateChain" minOccurs="0" maxOccurs="1">
               <xs:complexType>
                  <xs:choice maxOccurs="unbounded">
                     <xs:element name="Certificate" minOccurs="1" 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" minOccurs="1" maxOccurs="1">
               <xs:complexType>
                  <xs:choice maxOccurs="unbounded">
                     <xs:element name="Certificate" minOccurs="1" maxOccurs="unbounded">
                        <xs:simpleType>
                           <xs:restriction base="xs:base64Binary">
                              <xs:minLength value="4"/>
                           </xs:restriction>
                        </xs:simpleType>
                     </xs:element>
                  </xs:choice>
               </xs:complexType>
```


Copyright © 2008 Microsoft Corporation.

```
            </xs:element>
            <xs:element name="CheckCrl" minOccurs="0" maxOccurs="1">
               <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>
```

2.2.1.20.1.1 ValidateCert

The <ValidateCert> element is the top-level element in the XML document. It identifies the body of the HTTP **POST** as containing a **ValidateCert** command.

Parent elements	Child elements	Data type	Number allowed
None	<Certificates> (request only) <CertificateChain> (request only) <CheckCRL> (request only) <Status> (response only)	**Container**	1 (required)

2.2.1.20.1.2 Certificate

The <Certificate> element contains the Base64-encoded X509 certificate BLOB.

Parent elements	Child elements	Data type	Number allowed
<Certificates> <CertificateChain>	<Status> (response only)	**String** (Base64-encoded)	1…N

2.2.1.20.1.3 CertificateChain

The <CertificateChain> element contains the list of certificates to be validated.

Parent elements	Child elements	Data type	Number allowed
<ValidateCert> (request only)	<Certificate>	**Container**	0…1 (optional)

2.2.1.20.1.4 Certificates

The <Certificates> element contains the list of certificates to be validated.

Parent elements	Child elements	Data type	Number allowed
<ValidateCert> (request only)	<Certificate>	**Container**	1 (required)

Copyright © 2008 Microsoft Corporation.

2.2.1.20.1.5 CheckCRL

The <CheckCRL> element specifies whether the server SHOULD ignore an unverifiable revocation status.

Parent elements	Child elements	Data type	Number allowed
<ValidateCert>	None	**Integer**	0...1 (optional)

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

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

2.2.1.20.2 Response

The following code shows the XSD for the <ValidateCert> command response.

```
<?xml version="1.0" ?>
<xs:schema xmlns:tns="ValidateCert:" attributeFormDefault="unqualified"
elementFormDefault="qualified"
targetNamespace="ValidateCert:" xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="ValidateCert">
   <xs:complexType>
      <xs:sequence>
         <xs:element minOccurs="1" maxOccurs="1" name="Status" type="xs:unsignedByte" />
<xs:element minOccurs="0" maxOccurs="unbounded" name="Certificate">
   <xs:element minOccurs="1" maxOccurs="1" name="Status" type="xs:unsignedByte" />
</xs:element>
      </xs:sequence>
   </xs:complexType>
</xs:element>
```

2.2.1.20.2.1 ValidateCert

The <ValidateCert> element is the top-level element in the XML document. It identifies the body of the HTTP **POST** as containing a **ValidateCert** command.

Parent elements	Child elements	Data type	Number allowed
None	<Certificates> (request only) <CertificateChain> (request only) <CheckCRL> (request only) <Status> (response only)	**Container**	1 (required)

2.2.1.20.2.2 Certificate

The <Certificate> element contains the Base64-encoded **X509** certificate BLOB.

Copyright © 2008 Microsoft Corporation.

Parent elements	Child elements	Data type	Number allowed
<ValidateCert>	<Status> (response only)	**String** (Base64-encoded)	0...N

2.2.1.20.2.3 Status

The <Status> element indicates whether one or more certificates were successfully validated.

Parent elements	Child elements	Data type	Number allowed
<ValidateCert> (response only) <Certificate> (response only)	None	**Integer**	1...N (required)

The following table shows valid values for the element.

Value	Meaning
1	Successful validation.
2	Protocol error.
3	The signature in the digital ID can't be validated.
4	The digital ID was issued by an untrusted source.
5	The certificate chain that contains the digital ID was not created properly.
6	The digital ID is not valid for signing e-mail messages.
7	The digital ID used to sign the message has expired or is not yet valid.
8	The time periods during which the digital IDs in the certificate chain are not consistent.
9	A certificate is being used for a purpose other than what it was specified for.
10	Information associated with the digital ID is missing or incorrect.
11	A certificate that can only be used as an end-entity is being used as a certification authority (CA), or a CA that can only be used as an end-entity is being used as a certificate.
12	The digital ID doesn't match the recipient's e-mail address.
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.
14	The validity of the digital ID can't be determined because the server that provides this information can't be contacted.
15	A digital ID in the chain has been revoked by the authority that issued it.
16	The digital ID can't be validated because its revocation status can't be determined.
17	An unknown server error has occurred.

Copyright © 2008 Microsoft Corporation.

2.2.2 Status Codes

This section specifies the codes that the server returns in the <Status> element of each command response. For details about processing the return codes, see section 3.1.5.5.

For a complete list of status codes for all commands, see section 2.2.2.14.

Each status code has a scope assigned to it. The following table lists the scope values.

Scope Value	Description
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.

2.2.2.1 FolderCreate Status Codes

The following table lists the status codes for the **FolderCreate** command.

Code	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.
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 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).
8	The request timed out.	The server took too long to respond to the request.	Global	Retry.
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 that contains a semantic error.	Global	Fix bug in client code. Double-check the request for accuracy.
12	Code unknown.	Unusual back-end issue.	Global	No solution.

Copyright © 2008 Microsoft Corporation.

2.2.2.2 FolderDelete Status Codes

The following **table** lists the status codes for the **FolderDelete** command.

Code	Meaning	Cause	Scope	Resolution
1	Success.	Server successfully completed command.	Global	None.
3	The specified folder is a **special folder**; for example, Inbox, Outbox, contacts, and so on.	The client specified a special folder in a **FolderDelete** command request. 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 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).
8	The request timed out.	The server took too long to respond to the request.	Global	Retry.
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	Malformed request.	The client sent a **FolderCreate** command request that contains a semantic or syntactic error.	Global	Fix bug in client code. Double-check the request for accuracy.
11	Code Unknown.	Unusual back-end issue.	Global	No solution.

2.2.2.3 FolderSync Status Codes

The following table lists the status codes for the **FolderSync** command.

Code	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).

Copyright © 2008 Microsoft Corporation.

Code	Meaning	Cause	Scope	Resolution
8	The request timed out.	The server took too long to respond to the request.	Global	Retry.
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	Malformed request.	The client sent a **FolderSync** command request that contains a semantic or syntactic error.	Global	Fix bug in client code. 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 request. If continued attempts to synchronization fail, consider returning to synchronization key zero (0).
12	Code Unknown.	Unusual back-end issue.	Global	No solution.

2.2.2.4 FolderUpdate Status Codes

The following table lists the status codes for the **FolderUpdate** command.

Code	Meaning	Cause	Scope	Resolution
1	Success.	Server successfully completed command.	Global	None.
3	The specified folder is a special folder; for example, Inbox, Outbox, Contacts, and so on.	Client specified a special folder in a **FolderUpdate** command request. Special folders cannot be deleted.	Item	None.
4	The specified folder does not exist.	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 request. If continued attempts to synchronization fail, consider returning to synchronization key 0.
8	The request timed out.	The server took too long to respond to the request.	Global	Retry.
9	Synchronization key mismatch or invalid synchronization key.	The client sent a malformed or mismatched synchronization key, or the	Global	Issue a **FolderSync** command request with a synchronization key of 0.

Copyright © 2008 Microsoft Corporation.

Code	Meaning	Cause	Scope	Resolution
		synchronization state is corrupted on the server.		
10	Malformed request.	The client sent a **FolderCreate** command request that contains a semantic error.	Global	Fix bug in client code. Double-check the request for accuracy.
11	Code unknown.	Unusual back-end issue.	Global	No solution.

2.2.2.5 GetItemEstimate Status Codes

The following table lists the status codes for the **GetItemEstimate** command.

Code	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** 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 with synchronization key zero (0).	Item	Issue a **Sync** command with synchronization key of zero (0) before issuing **GetItemEstimate** again.
4	Invalid synchronization key	Malformed or mismatched synchronization key. —or— The synchronization state is corrupted on the server.	Global	Delete any items added since the last successful synchronization and return to synchronization key zero (0). The <ItemID> value is the new item ID sent back by Sync during synchronization zero (0). These <ItemID> values are different than the current item IDs. Client code MUST check for duplicate items.

2.2.2.6 MeetingResponse Status Codes

The following table lists the status codes for the **MeetingResponse** command.

Code	Meaning	Cause	Scope	Resolution
1	Success.	Server successfully completed command.	Global	None.
2	Invalid meeting	The client has sent a malformed or invalid item.	Item	Stop sending the item. This is not a transient condition.

Copyright © 2008 Microsoft Corporation.

Code	Meaning	Cause	Scope	Resolution
	request.			
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**. If continued attempts fail, synchronize the folder again, and then attempt the MeetingResponse command again. If it still continues to fail, make no changes.
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**. If continued attempts fail, synchronize the folder again, and then attempt the MeetingResponse command again. If it still continues to fail, make no changes.

2.2.2.7 MoveItems Status Codes

The following <table> lists the status codes for the **MoveItems** command.

Code	Meaning	Cause	Scope	Resolution
1	Invalid collectionId for source.	The source folder collectionId 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 collectionID.
2	Invalid collectionId for destination.	The destination folder collectionId is not recognized by the server, possibly because the source folder has been deleted.	Item	Issue a **FolderSync** to get the new hierarchy. Then, use a valid collectionID.
3	Success.	Server successfully completed command.	Global	None.
4	Source and destination collectionIds are the same.	The client supplied a destination folder that is the same as the source.	Item	Only send requests where the collectionIds for the source and destination differ.
5	An error occurred on the server.	Server misconfiguration, temporary system issue, or bad item. This is frequently a transient condition.	Global	Retry the **MoveItems**. If continued attempts fail, give up on the item.
7	Source or destination item was locked.	Transient server condition.	Item	Retry.

2.2.2.8 Ping Status Codes

The following table lists the status codes for the **Ping** command.

Code	Meaning	Cause	Scope	Resolution
1	The heartbeat interval expired before any changes occurred in the folders being		Global	Reissue the **Ping** command request.

Copyright © 2008 Microsoft Corporation.

Code	Meaning	Cause	Scope	Resolution
	monitored.			
2	Changes occurred in at least one of the monitored folders. The response specifies the changed folders.		Global	Issue a **Sync** request for each folder that was specified in the **Ping** command response.
3	The **Ping** command request omitted required parameters.	The **Ping** command request omitted one or both of the folder list or interval.	Global	Reissue the **Ping** command request with the entire XML body.
4	Syntax error in **Ping** command request.	Frequently caused by poorly formatted WBXML.	Global	Fix bug in client code.
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.	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 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.

2.2.2.9 Provision Status Codes

The following table lists the status codes for the **Provision** command.

Code	Meaning	Cause	Scope	Resolution
1	Success.	The requested policy data is included in	Policy	Apply the policy.

Copyright © 2008 Microsoft Corporation.

Code	Meaning	Cause	Scope	Resolution
		the response.		
2	Protocol error.	Syntax error in the **Provision** command request.	Global	Fix bug in client code.
2	Policy not defined.	There is no policy of the requested type that is defined on the server.	Policy	Stop sending policy information. No policy is implemented.
3	The policy type is unknown.	The client sent a policy that the server does not recognize.	Policy	Issue a request by using MS-EAS-Provisioning-WBXML.
3	An error occurred on the server.	Server misconfiguration, temporary system issue, or bad item. This is frequently a transient condition.	Global	Retry.
4	The policy data is corrupted.	The policy data on the server is corrupted.	Policy	Direct the user to contact the server administrator.
5	Policy key mismatch.	The client is trying to acknowledge an out-of-date or invalid policy.	Policy	Issue a new Provision request to obtain a valid policy key.

2.2.2.10 ResolveRecipients Status Codes

The following table lists the status codes for the **ResolveRecipients** command.

Code	Meaning	Cause	Scope	Resolution
1	Success.	Server successfully completed command.	Global	None.
2	The recipient provided was ambiguous. The response lists all possible matches. No certificate nodes were returned.	The recipient string supplied by the client matched more than one, but not more than <MaxAmbiguousRecipients>, recipients.	Item	Prompt the user to select the intended recipient from the list returned.
3	The recipient provided was ambiguous. The response lists some possible matches. No certificate nodes were returned.	The recipient string supplied by the client matched more than <MaxAmbiguousRecipients> recipients.	Item	Prompt the user to select the intended recipient from the list returned or to get more recipients.
4	No matching entries were found for the recipient. No certificates were returned.	The recipient does not exist or the supplied recipient string was incorrect.	Item	Inform the user of the error and direct the user to check the spelling.
5	Protocol error.	Syntactic or semantic error in the **ResolveRecipients** request.	Global	Fix bug in client code.
6	An error occurred on	Server misconfiguration, temporary	Global	Retry.

Copyright © 2008 Microsoft Corporation.

Code	Meaning	Cause	Scope	Resolution
	the server.	system issue, or bad item. This is frequently a transient condition.		
7	No certificate found.	No certificate was found but one was requested	Item	Prompt the user.
8	Global limit hit	The global limit of certificates for the request was hit	Item	Retry with fewer recipients if possible, otherwise prompt the user.
9	Certificate enumeration failure.	There was an error enumerating the certificates.	Item	Prompt the user.

2.2.2.11 Search Status Codes

The following table lists the status codes for the **Search** command.

Code	Meaning	Cause	Scope	Resolution
1	Success.	Server successfully completed command.	Global	None.
2	The request was invalid.	One or more of the search parameters was invalid.	Item	Inform the user which parameter was invalid. Describe the valid options and prompt the user to select a valid value.
3	An error occurred on the server.	Server misconfiguration, temporary system issue, or bad item. This is frequently a transient condition.	Global	Retry.
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 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	Folder sync required.	The folder hierarchy is out of date.	Global	Issue a **FolderSync** and try again.
12	End of	The requested range has gone	Local	Prompt the user that there are no

Copyright © 2008 Microsoft Corporation.

Code	Meaning	Cause	Scope	Resolution
	retrievable range warning.	past the end of the range of retrievable results.		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.

2.2.2.12 Sync Status Codes

The following table lists the status codes for the **Sync** command.

Code	Meaning	Cause	Scope	Resolution
1	Success.	Server successfully completed command.	Global	None.
2	Protocol version mismatch.	The client's version string specified version was different than previous **Sync** requests.	Global	Specify the correct version in the version string.
3	Invalid synchronization key.	Malformed or mismatched synchronization key. —or— Synchronization state corrupted on server.	Global	Delete any items that were added since the last successful synchronization and return to synchronization key 0. The ItemID value is the new item ID sent back by **Sync** during synchronization zero (0). These ItemID values are different than the current item IDs. Client code MUST check for duplicate items.
4	Protocol error.	There was a semantic error in the recovery synchronization.	Item or Global	This is caused by a bug in the client. Fix the client. In recovery synchronization, resend all changes from previous synchronization. Wait until all changes are acknowledged before committing the new synchronization key. Return to synchronization key zero (0).
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

Copyright © 2008 Microsoft Corporation.

Code	Meaning	Cause	Scope	Resolution
				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> command that has an ItemID 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 try the **Sync** command again.
10	The <NotifyGUID> element caused an error.	An error occurred while setting the <NotifyGUID> element.	Item	This error SHOULD be ignored, as the <NotifyGUID> element is no longer used.
11	The **Sync** command cannot be completed.	The device has not been provisioned for notifications yet.	Item	Use the **Provision** command to request policy settings from the server then retry the **Sync** command.
12	The folder hierarchy has changed.	Mailbox folders are not synchronized.	Item	Perform a **FolderSync** command 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> value.	The <Wait> element value is too large or too small.	Item	Update the <Wait> element value according to the <Limit> element and then resend the <Sync> command request.
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.

2.2.2.13 ValidateCert Status Codes

The following table lists the status codes for the **ValidateCert** command.

Copyright © 2008 Microsoft Corporation.

Code	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.
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 e-mail messages.	The supplied certificate is not meant to be used for signing e-mail.	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 in the chain is being used for a purpose other than what it was intended for.	Item	Obtain the correct certificate chain.
12	The digital ID does not match the recipient's e-mail 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	The certificate has been revoked by the certification authority that issued	Item	Obtain a new certificate.

Copyright © 2008 Microsoft Corporation.

Code	Meaning	Cause	Scope	Resolution
	digital ID was reported stolen, or the digital ID was compromised.	it.		
14	The validity of the digital ID cannot be determined because the server that provides this information cannot be contacted.	The certificate revocation server is offline.	Item	Retry request after some time.
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.

2.2.2.14 Common Status Codes

The following table lists all common status codes that MAY be returned within any of the commands. They are used in addition to the specific error codes described with the previous sections of this document.

Name	Description	Value
InvalidContent	The body of the HTTP request sent by the client is invalid.<56> 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** with a text/plain body, or **SendMail** with version 12.0 and a WBXML body.	101
InvalidWBXML	The request contains WBXML but it could not be decoded into XML.	102
InvalidXML	The XML provided in the request did not follow the [MS-ASCMD] protocol.	103
InvalidDateTime	The request contains a timestamp that could not be parsed into a valid date and time.	104
InvalidCombinationOfIDs	The request contained a combination of parameters that is invalid.	105
InvalidIDs	The request contains one or more IDs that could not be parsed into valid	106

Copyright © 2008 Microsoft Corporation.

Name	Description	Value
	values. That is different from specifying an ID in the proper format but does not resolve to an existing item.<57>	
InvalidMIME	The request contains a MIME that could not be parsed.	107
DeviceIdMissingOrInvalid	The DeviceID is either missing or has an invalid format	108
DeviceTypeMissingOrInvalid	The DeviceType is either missing or has an invalid format	109
ServerError	The server encountered an unknown server.<58>	110
ServerErrorRetryLater	The server encountered an unknown error, the device should retry later.<59>	111
ActiveDirectoryAccessDenied	The server does not have access to read/write to an object in the ActiveDirectory<60>	112
MailboxQuotaExceeded	The mailbox has reached its size quota.<61>	113
MailboxServerOffline	The mailbox server is offline.	114
SendQuotaExceeded	The request would exceed the "send" quota.	115
MessageRecipientUnresolved	One of the recipient could not be resolved to an e-mail address.	116
MessageReplyNotAllowed	The mailbox server will not allow a reply of this message.	117
Message PreviouslySent	The message was already sent in a previous request. The server determined this by remembering the <ClientId> values of the last few sent messages. This request contains a <ClientId> that was already used in a recent message.	118
MessageHasNoRecipient	The message being sent contains no recipient.	119
MailSubmissionFailed	The server failed to submit the message for delivery.	120
MessageReplyFailed	The server failed to create a reply message.	121
AttachmentIsTooLarge	The attachment is too large to be processed by this request.	122

Copyright © 2008 Microsoft Corporation.

Name	Description	Value
UserHasNoMailbox	The user does not appear to have a mailbox (according to the **Active Directory**).	123
UserCannotBeAnonymous	The request was sent without credentials. Anonymous requests are not allowed.	124
UserPrincipalCouldNotBeFound	The user was not found in the Active Directory.	125
UserDisabledForSync	The user object in the Active Directory indicates that this user is not allowed to use ActiveSync.	126
UserOnNewMailboxCannotSync	The server is configured to prevent users from syncing.	127
UserOnLegacyMailboxCannotSync	The server is configured to prevent users on legacy servers from syncing.	128
DeviceIsBlockedForThisUser	The user is configured to allow only some devices to sync. This device is not the allowed device.	129
AccessDenied	The user is not allowed to perform that request.	130
AccountDisabled	The user's account is disabled.	131
SyncStateNotFound	The sync state for the device 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.<62>	132
SyncStateLocked	The sync state is locked. Possibly because the mailbox is being moved or was recently moved.	133
SyncStateCorrupt	The sync state appears to be corrupt.	134
SyncStateAlreadyExists	The sync state for this device already exists. This can happen with two initial syncs are executed concurrently.	135
SyncStateVersionInvalid	The sync state version is invalid.	136
CommandNotSupported	The command is not supported by this server.<63>	137
VersionNotSupported	The command is not supported in the protocol version specified.<64>	138
DeviceNotFullyProvisionable	The device uses a protocol version that cannot send all the policy settings the admin enabled.	139

Copyright © 2008 Microsoft Corporation.

Name	Description	Value
RemoteWipeRequested	A remote wipe was requested. The device should provision to get the request and then do another provision to acknowledge it.<65>	140
LegacyDeviceOnStrictPolicy	A policy is in place but the device is not provisionable.	141
DeviceNotProvisioned	There is a policy in place, the device needs to provision.<66>	142
PolicyRefresh	The policy is configured to be refreshed every few hours. The device needs to re-provision.	143
InvalidPolicyKey	The device's policy key is invalid. The policy has probably changed on the server. The device needs to re-provision.	144
ExternallyManagedDevicesNotAllowed	The device claimed to be externally managed, but the server doesn't allow externally managed devices to sync.	145
NoRecurrenceInCalendar	The request tried to forward an occurrence of a meeting that has no recurrence.	146
UnexpectedItemClass<67>	The request tried to operate on a type of items unknown to the server.	147
RemoteServerHasNoSSL	The request needs to be proxied to another server but that server doesn't have Secure Socket Layers (SSL) enabled. This server is configured to only **proxy request**s to servers with SSL enabled.	148
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 impossible. The device needs to send the full request again.	149
ItemNotFound	The item specified in the request could not be found in the mailbox.	150
TooManyFolders	The mailbox contains too many folders.	151
NoFoldersFound	The mailbox contains no folders.	152
ItemsLostAfterMove	After moving items to the destination folder, some of those items could not be found.	153
FailureInMoveOperation	The mailbox server returned an	154

Copyright © 2008 Microsoft Corporation.

Name	Description	Value
	unknown error while moving items.	
MoveCommandDisallowedForNonPersistentMoveAction	Only the MoveAlways action is supported.	155
MoveCommandInvalidDestinationFolder	The destination folder for the move is invalid.	156

Copyright © 2008 Microsoft Corporation.

3 Protocol Details

3.1 Common Details

3.1.1 Abstract Data Model

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

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 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 describes how the client device can download policy settings from the server by using the **Provision** command.

The first command the client issues to the server MUST be a **Provision** command. This initial request for policy settings contains the <PolicyType> element, which specifies the format in which the policy settings are provided. The server then responds with the <PolicyType>, <PolicyKey>, and <Data> elements. The policy key 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 <PolicyType> element, are contained in the <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 <PolicyType>, <PolicyKey>, and <Status> elements. The <Status> element indicates whether the policy settings were successfully applied by the client. The response from the server contains <PolicyType>, <PolicyKey>, and <Status> elements. The <Status> element indicates whether the server successfully recorded the client's acknowledgement.

The following figure shows the process for downloading policy settings.

Copyright © 2008 Microsoft Corporation.

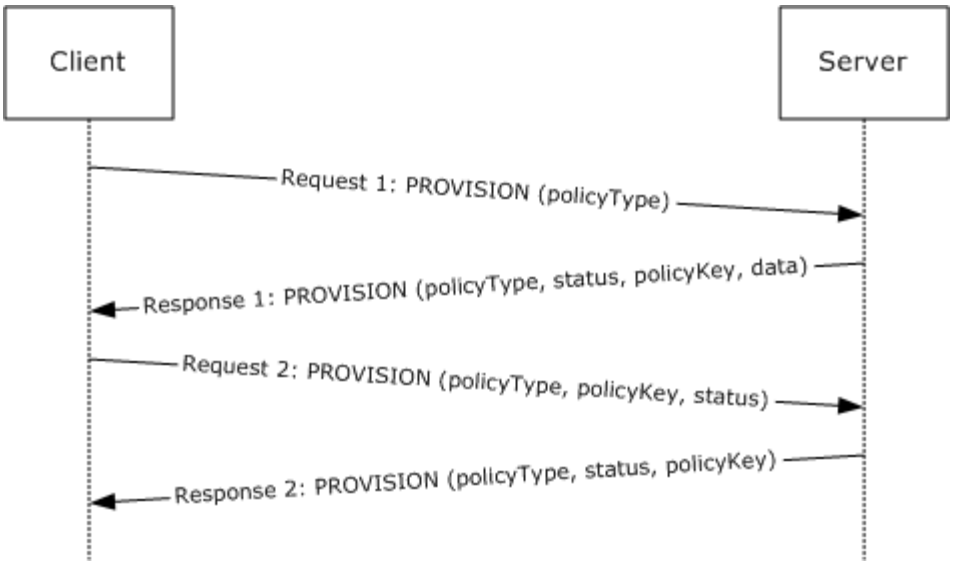

Figure 4: Downloading policy settings

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 response contains 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 response contains the policy type, policy key, and status code to indicate that the server recorded the client's acknowledgement.

3.1.5.2 Synchronizing a Folder Hierarchy

This section describes how to use the **FolderSync** command 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 <SyncKey> key of zero (0). The server responds with a new <SyncKey> value and provides a list of all the folders in the user's mailbox. The folders are identified by a <ServerId>, which can then be used in a **Sync** command to synchronize the items in those folders.

Additional folder synchronizations can be performed by using the <SyncKey> 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> of zero (0), and resynchronizing the entire hierarchy. Existing <ServerId> values do not change when the client resynchronizes.

The client can use the **GetItemEstimate** command 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 <WindowSize> element, which specifies the maximum number of items to synchronize at one time. If the number of items returned is larger

Copyright © 2008 Microsoft Corporation.

than the <WindowSize>, the <MoreAvailable> element is returned in the **Sync** command response. The client then continues to call the **Sync** command until no more items are available. The following figure shows the process for synchronizing multiple folders.

The following table lists the command sequence for folder hierarchy synchronization.

The asterisk (*) in the Order column means that a step can be repeated multiple times. [n] means that a step is optional.

Order	Client action	Server action
1	The client sends the **FolderSync** command with the <SyncKey> element set to zero (0) to get the folder hierarchy and the <ServerId> values of all the folders.	The server response contains the folder hierarchy and a new <SyncKey>. The client stores the names and <ServerId> values of all folders that can be synchronized.
2*	The client sends the **FolderSync** command with the new <SyncKey> value to update the folder heirarchy.	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 by synchronized.

3.1.5.3 Synchronizing Inbox, Calendar, Contacts, and Tasks Folders

The client synchronizes the contents of individual folders by using the **Sync** command. The client can synchronize the Inbox, Calendar, or Contacts folder, or any folder within the mailbox after the folder hierarchy has been populated by the **FolderSync** command, as described in section 3.1.5.2.

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 <SyncKey> value of zero (0) and the <CollectionId> for the appropriate folder. The **Sync** command response includes a new <SyncKey> value, which is generated by the server for each transaction.

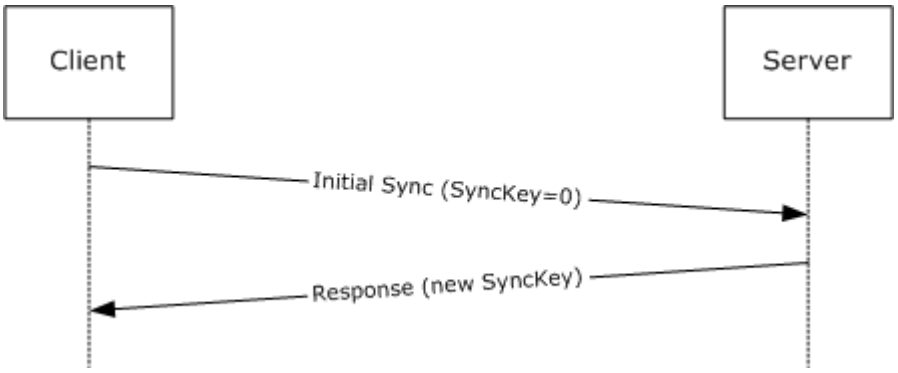

Figure 5: Retrieval of SyncKey value

The <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 <SyncKey> element as well as the <GetChanges> element.

Copyright © 2008 Microsoft Corporation.

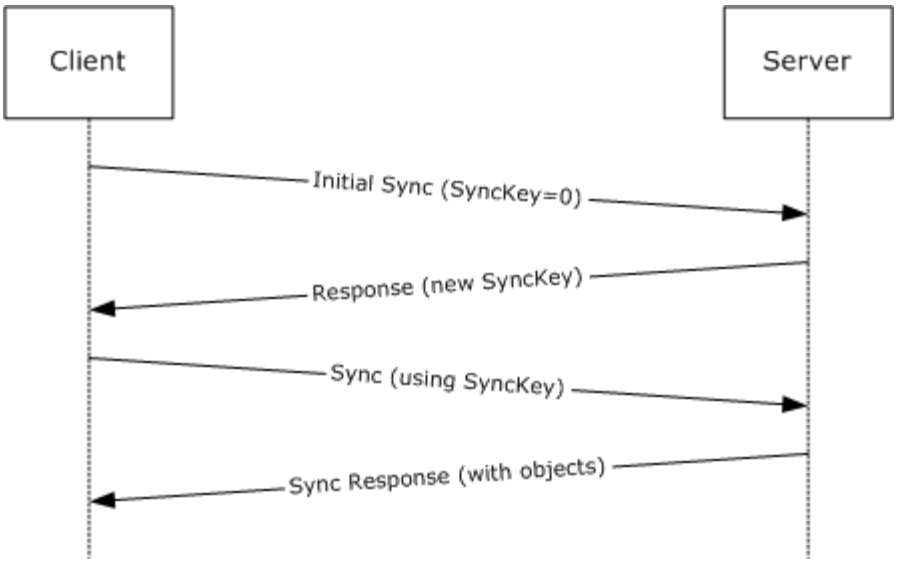

Figure 6: Retrieval of folder content

The server responds by adding all the items in the collection to the client and returning a new <SyncKey>, which can be used in successive synchronizations. 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 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 (UI) displays a progress bar while getting items from the server. In some cases, the client could have to submit a <WindowSize> element that specifies the number of items to be synchronized at a time.

If more items remain to be synchronized, the <MoreAvailable> element 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 <WindowSize> element of the **Sync** command, see section 2.2.1.19.1.5. The following figure shows the folder synchronization process.

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 a 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 **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.	The server response contains the synchronization key 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	The server responds with new synchronization keys for each collection.

Copyright © 2008 Microsoft Corporation.

Order	Client action	Server action
	to be synchronized.	
[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 response indicates how many items will be added for each collection.
4*	The client sends the **Sync** command with the <GetChanges> element for a collection. The command SHOULD include the <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 response contains <Add> elements for items in the collection. If the response contains the <MoreAvailable> element, this step is repeated.

The client SHOULD use the <WindowSize> element to break the server <Add> elements into sets of multiple items. The recommended window size is 100. For more details about the <WindowSize> element used by the **Sync** command, see section 2.2.1.19.1.5.

3.1.5.4 Receiving and Accepting Meeting Requests

This section describes how to retrieve items from the Inbox folder by using the **Sync** command, to respond to a meeting request item by using the **MeetingResponse** command, and to synchronize 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 e-mail message that has an embedded calendar item. The message contains a <MessageClass> element that has a value of IPM.Schedule.Meeting.Request, and its <ApplicationData> element contains a <MeetingRequest> element. When the client displays the meeting request message, 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.

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 <CalendarId> element of the response. If the response to the meeting is declined, the response will not contain a <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.

Copyright © 2008 Microsoft Corporation.

Order	Client action	Server action
1	The client sends the **Sync** command for the Inbox collection with the value of the <SyncKey> element set to zero (0).	The server response contains the <SyncKey> for the collection, to be used in successive synchronizations.
2*	The client sends a **Sync** command, specifying the **GetChanges** element and the <SyncKey> for the Inbox folder. The command SHOULD include the <WindowSize> element, the recommended value for which is 100.	The server response contains <Add> elements for items in the Inbox collection, including a meeting request item. If the response contains the <MoreAvailable> element, 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 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.
5	If a response was requested by the organizer, the client should use a **SendMail** command to send an appropriately formatted meeting response	The server acknowledges the sending of the e-mail.
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 Handling Status Errors

Client software MUST handle errors that occur during synchronization sessions. Errors fall into two categories: HTTP errors and synchronization 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. Synchronization errors result from a problem during the synchronization process. Synchronization errors are indicated by codes that are returned in the <Status> element of a command response. For more details about the status codes, see section 2.2.2.

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.

In addition to synchronization 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, either immediately or later. 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 server response.

Copyright © 2008 Microsoft Corporation.

- 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 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.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

Copyright © 2008 Microsoft Corporation.

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, see [MS-ASPROV] section 4.

4.2 Discovering Account Settings by Using the AutoDiscover Command

The **Autodiscover** command 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 e-mail address and domain credentials, for example, kim@contoso.com.

2. The client uses the domain information in the user's e-mail 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

   https://autodiscover.<*SMTP-address-domain*>/autodiscover/autodiscover.xml

   In this example, these URLs map to https://contoso/autodiscover/autodiscover.xml and 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.

4. If redirection to another Autodiscover service is required, then the <Redirect> element 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 initiatesynchronization. For more details about performing the DNS SRV lookup, see [DNS-SRV].

The following sections show success and error response messages.

Account autodiscovery uses an e-mail address to look up information that is required to configure software. Given an e-mail name (such as EduardDell@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:

Copyright © 2008 Microsoft Corporation.

- autodiscover.woodgrovebank.com
- woodgrovebank.com

Each server is sent an HTTP **POST** command. The post data is an XML request for a certain type of information. E-mail account configuration is the first use. The XML contains information that helps execute the request. For mail, the information includes the e-mail 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

```
<Autodiscover
xmlns="http://schemas.microsoft.com/exchange/autodiscover/mobilesync/requestschema/2006">
   <Request>
      <EMailAddress>eduarddell@woodgrovebank.com</EMailAddress>
      <AcceptableResponseSchema>
      http://schemas.microsoft.com/exchange/autodiscover/mobilesync/
      responseschema/2006
      </AcceptableResponseSchema>
   </Request>
</Autodiscover>
```

4.2.2 Response - Case Error

```
<Autodiscover
xmlns:A="http://schemas.microsoft.com/exchange/autodiscover/mobilesync/responseschema/2006">
    <A:Response>
        <A:Culture>en:en</A:Culture>
        <A:User>
           <A:EMailAddress>eduarddell@woodgrovebank.com</A:EMailAddress>
       </A:User>
       <A:Action>
           <A:Error>
               <Status>1</Status>
               <Message>The directory service could not be reached</Message>
               <DebugData>MailUser</DebugData>
           </A:Error>
       </A:Action>
    </A:Response>
</Autodiscover>
```

4.2.3 Response - Case Redirect

In the following redirect example, assume that the **Autodiscover** command request was sent to autodiscover.woodgrovebank.com. The redirect node indicates to the client to try autodiscover.loandept.woodgrovebank.com.

```
<Autodiscover
xmlns:A="http://schemas.microsoft.com/exchange/autodiscover/mobilesync/responseschema/2006">
    <A:Response>
        <A:Culture>en:en</A:Culture>
```


Copyright © 2008 Microsoft Corporation.

```
        <A:User>
           <A:DisplayName>Eduard Dell</A:DisplayName>
           <A:EMailAddress>eduarddell@woodgrovebank.com</A:EMailAddress>
        </A:User>
        <A:Action>
           <A:Redirect>eduarddell@loandept.woodgrovebank.com </A:Redirect>
        </A:Action>
    </A:Response>
</Autodiscover>
```

4.2.4 Response - Case Server Settings

In the following success response, the Autodiscover service has provided server URL information for two services: MobileSync and CertEnroll. The client can use the MobileSync URL to configure the settings for the [MS-ASCMD]. The client can also optionally use the CertEnroll information to obtain a client certificate for SSL negotiation.<68>

```
<Autodiscover
xmlns:A="http://schemas.microsoft.com/exchange/autodiscover/mobilesync/responseschema/2006">
    <A:Response>
        <A:Culture>en:en</A:Culture>
        <A:User>
            <A:DisplayName>Eduard Dell</A:DisplayName>
            <A:EMailAddress>eduarddell@woodgrovebank.com</A:EMailAddress>
        </A:User>
        <A:Action>
            <A:Settings>
                <A:Server>
                    <A:Type>MobileSync</A:Type>
                    <A:Url>
                        https://loandept.woodgrovebank.com/Microsoft-Server-ActiveSync
                    </A:Url>
                    <A:Name>
                 https://loandept.woodgrovebank.com/Microsoft-Server-ActiveSync
             </A:Name>
                </A:Server>
                <A:Server>
                    <A:Type>CertEnroll</A:Type>
                    <A:Url>https://cert.woodgrovebank.com/CertEnroll</A:Url>
                    <A:Name />
                   <A:ServerData>CertEnrollTemplate</A:ServerData>
                </A:Server>
            </A:Settings>
        </A:Action>
    </A:Response>
</A:Autodiscover>
```

4.2.5 Response - Case Framework Error

If the provider cannot be found, or if the <AcceptableResponseSchema> cannot be matched, then the following XML fragment is returned to the client.

The error code 600 means an invalid request was sent to the service, and 601 means that a provider could not be found to handle the <AcceptableResponseSchema> that was specified.

Copyright © 2008 Microsoft Corporation.

```
<Autodiscover
xmlns:A="http://schemas.microsoft.com/exchange/autodiscover/mobilesync/responseschema/2006">
   <A:Response>
      <A:Error Time="16:56:32.6164027" Id="1054084152">
          <A:ErrorCode>600</ErrorCode>
          <A:Message>Invalid Request</Message>
          <A:DebugData />
      </A:Error>
   </A: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.<69>

```
<Autodiscover>
  <Account>
    <AccountType>default e-mail</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 Synchronizing Data by Using the Sync Command

This section provides sample messages related to the **Sync** command.

Copyright © 2008 Microsoft Corporation.

4.3.1 Synchronizing Folders

The following example shows a request that is sent to the server to synchronize an e-mail 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 specified in the response. The server response contains the new synchronization key and the items to be added, deleted, and changed on the client.

4.3.1.1 Request

```
<Collections>
    <Collection>
        <Class>Email</Class>
        <SyncKey>6</SyncKey>
        <CollectionId>1</CollectionId>
        <DeletesAsMoves/>
        <GetChanges/>
        <Options> ... </Options>
    </Collection>
</Collections>
```

4.3.1.2 Response

```
<Collections>
    <Collection>
        <Class>Email</Class>
        <SyncKey>7</SyncKey>
        <CollectionId>1</CollectionId>
        <Status>1</Status>
        <Commands>
            <Add>...</Add>
            <Delete>...</Delete>
            <Change>...</Change>
            <Fetch>...</Fetch>
        </Commands>
    </Collection>
</Collections>
```

4.3.2 Fetching an E-Mail by Using the ServerId

The following example shows a request that is sent to the server to fetch an item from the server by its <ServerId>.

4.3.2.1 Request

```
<Commands>
    <Fetch >
        <ServerId>1:14</ServerId>
    </Fetch >
</Commands>
```

A response from the server contains the server ID, status, and application data of the requested item.

Copyright © 2008 Microsoft Corporation.

4.3.2.2 Response

```
<Responses>
    <Fetch>
        <ServerId>1:14</ServerId>
        <Status>1</Status>
        <ApplicationData>...</ApplicationData>
    </Fetch>
</Responses>
```

4.3.3 Uploading New ApplicationData to the Server

This example shows a **Sync** command request that is sent to the server to add a contact. The response from the server shows that the synchronization was successful and that the new item from the client, identified by the <ClientId> element, was added to the collection on the server. The server also assigns a permanent ID for the newly added item in the <ServerId> element. After the client receives a successful response, the client uses this server ID for any future <Change> or <Delete> commands for this item.

4.3.3.1 Request

```
<Commands>
    <Add>
        <ClientId>123</ClientId>
        <ApplicationData>
            <A:Email1Address>schai@fourthcoffee.com</A:Email1Address>
            <A:FirstName>Sean</A:FirstName>
            <A:MiddleName>W</A:MiddleName>
            <A:LastName>Chai</A:LastName>
            <A:Title>Sr Marketing Manager</A:Title>
        </ApplicationData>
    </Add>
</Commands>
```

4.3.3.2 Response

```
<Responses>
    <Add>
        <ClientId>123</ClientId>
        <ServerId>4:1</ServerId>
        <Status>1</Status>
    </Add>
</Responses>
```

4.3.4 Updating ApplicationData on the Server

The following example shows a **Sync** command request from the client. The request modifies a contact, which is identified by the server ID, on the server. The response from the server shows that the change that is specified by the **Sync** request of the client succeeded and supplies the synchronization key and collection ID of the changed item.

Copyright © 2008 Microsoft Corporation.

4.3.4.1 Request

```
<Commands>
    <Change>
        <ServerId>3:1</ServerId>
        <ApplicationData>
            <A:Email1Address>jsmith@fourthcoffee.com</A:Email1Address>
            <A:FirstName>Jeff</A:FirstName>
        </ApplicationData>
    </Change>
</Commands>
```

4.3.4.2 Response

```
<Collections>
    <Collection>
      <Class>Contacts</Class>
      <SyncKey>4</SyncKey>
      <CollectionId>1</CollectionId>
      <Status>1</Status>
    </Collection>
</Collections>
```

4.3.5 Downloading Current Information from the Server

The following example shows a request that is sent to the server to synchronize an e-mail 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 specified in the response. The server response contains the new synchronization key and the items to be added, deleted, and changed on the client.

4.3.5.1 Request

```
<Collections>
<Collection>
        <Class>Email</Class>
        <SyncKey>6</SyncKey>
        <CollectionId>1</CollectionId>
        <DeletesAsMoves/>
        <GetChanges/>
        <Options> ... </Options>
</Collection>
</Collections>
```

4.3.5.2 Response

```
<Collections>
<Collection>
        <Class>Email</Class>
        <SyncKey>7</SyncKey>
        <CollectionId>1</CollectionId>
        <Status>1</Status>
        <Commands>
            <Add>...</Add>
            <Delete>...</Delete>
```


Copyright © 2008 Microsoft Corporation.

```
            <Change>...</Change>
            <Fetch>...</Fetch>
        </Commands>
</Collection>
</Collections>
```

4.3.6 Identifying Acceptance of Partial Collections

The following example shows a **Sync** request that includes the <Partial> element.

```
<Sync xmlns="AirSync:">
    <Collections>
        <Collection>
            <SyncKey>1723058747</SyncKey>
            <CollectionId>10</CollectionId>
        </Collection>
    </Collections>
    <Wait>8</Wait>
    <Partial/>
</Sync>
```

4.3.7 Identifying Acceptance of MIME Content

This example uses the <MIMETruncated>, <MIMETruncation>, and <MIMEData> elements to identify acceptance of MIME content on the client.

4.3.7.1 Sync Request With Support for MIME Content

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:">
  <Collections>
    <Collection>
      <Class>Email</Class>
      <SyncKey>2</SyncKey>
      <CollectionId>1</CollectionId>
      <DeletesAsMoves/>
      <GetChanges/>
      <Options>
        <MIMETruncation>1</MIMETruncation>
        <MIMESupport>1</MIMESupport>
      </Options>
    </Collection>
  </Collections>
</Sync>
```

4.3.7.2 Sync Response With MIME Content

```
<Add>
   <ServerId>1:3</ServerId>
   <ApplicationData>
      <A:To>"James Smith" &lt;jsmith@contoso.com&gt;</A:To>
      <A:From>"Jyothi Pai" &lt;jpai@contoso.com&gt;</A:From>
      <A:Subject>RE: Presentation</A:Subject>
      <A:DateReceived>2004-11-12T00:45:06.000Z</A:DateReceived>
```


Copyright © 2008 Microsoft Corporation.

```
      <A:DisplayTo>James Smith</A:DisplayTo>
      <A:Importance>1</A:Importance>
      <A:Read>0</A:Read>
      <A:MIMETruncated>0</A:MIMETruncated>
      <A:MIMEData>"Received: from server1.contoso.com
      ([192.168.0.20]) by server2.contoso.com with Microsoft
      SMTPSVC(5.0.2195.6624); ... </A:MIMEData>
      <A:Importance>1</A:Importance>
      <A:Read>0</A:Read>
      <A:MessageClass>IPM.Note.SMIME.MultipartSigned</A:MessageClass>
   </ApplicationData>
</Add>
```

4.3.7.3 Sync Request With BodyPreference and MIME Support

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:" xmlns:A="AirSyncBase:">
  <Collections>
    <Collection>
      <Class>Email</Class>
      <SyncKey>2</SyncKey>
      <CollectionId>17</CollectionId>
      <DeletesAsMoves/>
      <GetChanges/>
      <Options>
        <A:BodyPreference>
          <A:Type>4</A:Type>
          <A:TruncationSize>512</A:TruncationSize>
        </A:BodyPreference>
        <MIMESupport>1</MIMESupport>
      </Options>
    </Collection>
  </Collections>
</Sync>
```

4.3.7.4 Sync Response with MIME Support

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:" xmlns:A="POOMMAIL:" xmlns:B="AirSyncBase:">
  <Collections>
    <Collection>
      <Class>Email</Class>
      <SyncKey>3</SyncKey>
      <CollectionId>17</CollectionId>
      <Status>1</Status>
      <Commands>
        <Change>
          <ServerId>17:11</ServerId>
          <ApplicationData>
            <A:To>"Mike Phipps" &lt;mike@contoso.com&gt;</A:To>
            <A:From>"Arlene Huff" &lt;arlene@contoso.com&gt;</A:From>
            <A:Subject>opaque s + e </A:Subject>
            <A:DateReceived>2007-02-01T06:42:46.015Z</A:DateReceived>
            <A:DisplayTo>Mike Phipps</A:DisplayTo>
            <A:ThreadTopic>opaque s + e</A:ThreadTopic>
            <A:Importance>1</A:Importance>
```


Copyright © 2008 Microsoft Corporation.

```
            <A:Read>1</A:Read>
            <B:Attachments>
              <B:Attachment>
                <B:DisplayName>smime.p7m</B:DisplayName>
                <B:FileReference>17%3a11%3a0</B:FileReference>
                <B:Method>1</B:Method>
                <B:EstimatedDataSize>9340</B:EstimatedDataSize>
              </B:Attachment>
            </B:Attachments>
            <B:Body>
              <B:Type>4</B:Type>
              <B:EstimatedDataSize>13814</B:EstimatedDataSize>
              <B:Truncated>1</B:Truncated>
              <B: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: Arlene Huff &lt;arlene@contoso.com&gt;
              To: Mike &lt;mike@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</B:Data>
            </B:Body>
            <A:MessageClass>IPM.Note.SMIME</A:MessageClass>
            <A:InternetCPID>20127</A:InternetCPID>
            <A:Flag/>
            <A:ContentClass>urn:content-classes:message</A:ContentClass>
            <B:NativeBodyType>1</B:NativeBodyType>
          </ApplicationData>
        </Change>
      </Commands>
    </Collection>
  </Collections>
</Sync>
```

4.3.8 Identifying That More Content is Ready for Download

The following example is a response message indicating that more content is available for download from the server. The content exceeded the <WindowSize> value.

```
<Collection>
      <Class>Email</Class>
      <SyncKey>2</SyncKey>
      <CollectionId>1</CollectionId>
      <Status>1</Status>
      <Commands>
            ...
      </Commands>
      <MoreAvailable/>
</Collection>
```


Copyright © 2008 Microsoft Corporation.

4.3.9 Synchronizing the Calendar Folder

The following example shows the initial synchronization of the Calendar folder with a synchronization key of 0.

4.3.9.1 Request

```
<Collection>
    <Class>Calendar</Class>
    <SyncKey>0</SyncKey>
    <DeletesAsMoves/>
    <GetChanges/>
</Collection>
```

The following example shows the synchronization of the calendar with a synchronization key that was obtained from a previous synchronization.

4.3.9.2 Request

```
<Collection>
    <Class>Calendar</Class>
    <SyncKey>9</SyncKey>
    <DeletesAsMoves/>
    <GetChanges/>
</Collection>
```

4.3.10 Empty Sync Request and Response

This section demonstrates scenario in which an empty **Sync** response and empty **Sync** request are exchanged between the client and server. For more details about empty **Sync** requests, see section 2.2.1.19.1.1.1. For more details about empty **Sync** responses, see section 2.2.1.19.2.17.1.

The scenario begins when **Sync** request is issued by the client and indicates that there are no pending changes to resport to the server. The **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

<?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>
```


Copyright © 2008 Microsoft Corporation.

```
    <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.

4.4 Pinging the Server for Updates by Using the Ping Command

This section provides sample messages related to **Ping**.

4.4.1 Ping Command Request

The following is an example of the **Ping** element in a **Ping** command request.

```
<?xml version="1.0" encoding="utf-8"?>
<Ping xmlns="Ping:">
  <Folders>
  ...
  </Folders>
</Ping>
```

4.4.2 Ping Command Response

4.4.2.1 Typical Response

The following example shows a typical response to a **Ping** command request, when the heartbeat interval that was specified by the client has expired and there were no changes in any of the specified folders.

```
<Ping xmlns="Ping:">
  <Status>1</Status>
```


Copyright © 2008 Microsoft Corporation.

```
</Ping>
```

4.4.2.2 Response – Changes Found

The following response message shows that changes have occurred in two folders that were being monitored. The client then synchronizes the specified folders. Do not reissue the next **Ping** command until the folders have been synchronized.

```
<Ping xmlns="Ping:">
  <Status>2</Status>
  <Folders>
    <Folder>1234</Folder>
    <Folder>5678</Folder>
  </Folders>
</Ping>
```

4.4.2.3 Response – HeartbeatInterval Error

The following example shows a response to a **Ping** command request that specified a heartbeat interval outside the acceptable range. The returned heartbeat interval is either the minimum or maximum allowed value. The client compares the requested interval with the returned interval and determine whether the requested heartbeat interval was either too great or too small.

```
<Ping xmlns="Ping:">
  <Status>5</Status>
  <HeartbeatInterval>60</HeartbeatInterval>
</Ping>
```

4.4.2.4 Response – Folder Error

The following example shows a response to a **Ping** command request where the number of folders that was specified 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 <MaxFolders> element.

```
<Ping xmlns="Ping:">
  <Status>6</Status>
  <MaxFolders>200</MaxFolders>
</Ping>
```

4.5 Fetching E-Mail and Attachments by Using the ItemOperations Command

The **ItemOperations** command enables the client to retrieve PIM items and attachments (in addition to document library items and search results) outside the **Sync** command context.

These examples 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 e-mail and attachments.

Copyright © 2008 Microsoft Corporation.

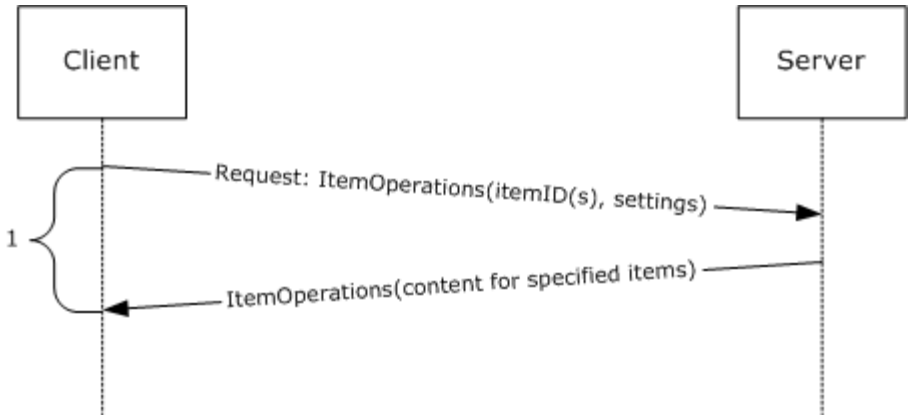

Figure 7: Fetching E-mail

4.5.1 Fetching an E-Mail Item

The following example shows the client retrieving an e-mail message by using the **ItemOperations** command.

4.5.1.1 Request

```
POST /Microsoft-Server-
ActiveSync?Cmd=ItemOperations&User=deviceuser&DeviceId=device1&
DeviceType=PocketPC HTTP/1.1
Content-Type: application/vnd.ms-sync.wbxml
MS-ASProtocolVersion: 14.0

<?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>
```

4.5.1.2 Response

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Length: 409
Content-Type: application/vnd.ms-sync
Server: Microsoft-IIS/6.0
MS-Server-ActiveSync: 14.0
Date: Tue, 08 May 2007 17:29:52 GMT
```


Copyright © 2008 Microsoft Corporation.

```
<?xml version="1.0" encoding="utf-8"?><ItemOperations
xmlns:airsync="AirSync:" xmlns:email="POOMMAIL:"
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;someone1@example.com&gt;</email:To>
        <email:Cc>"deviceuser3" &lt;someone3@example.com&gt;</email:Cc>
        <email:From>"deviceuser2" &lt;someone2@example.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>
      </Properties>
    </Fetch>
  </Response>
</ItemOperations>
```

4.5.2 Fetching a MIME E-Mail Item

The following example shows the client retrieving a MIME e-mail message by using the <MIMESupport> option.

4.5.2.1 Request

```
<?xml version="1.0" encoding="utf-8"?>
<ItemOperations xmlns="ItemOperations:" xmlns:A="AirSync:" xmlns:B="AirSyncBase:">
   <Fetch>
      <Store>Mailbox</Store>
      <A:CollectionId>17</A:CollectionId>
      <A:ServerId>17:11</A:ServerId>
      <Options>
         <MIMESupport xmlns="AirSync:">1</MIMESupport>
         <B:BodyPreference>
            <B:Type>4</B:Type>
```


Copyright © 2008 Microsoft Corporation.

```
            </B:BodyPreference>
         </Options>
      </Fetch>
   </ItemOperations>
```

4.5.2.2 Response

```
<?xml version="1.0" encoding="utf-8"?>
<ItemOperations xmlns="ItemOperations:" xmlns:A="AirSync:" xmlns:B="POOMMAIL:"
xmlns:C="AirSyncBase:">
   <Status>1</Status>
   <Response>
      <Fetch>
         <Status>1</Status>
         <A:CollectionId>17</A:CollectionId>
         <A:ServerId>17:11</A:ServerId>
         <A:Class>Email</A:Class>
         <Properties>
            <B:To>"Mike Phipps" &lt;mike@contoso.com&gt;</B:To>
            <B:From>"Arlene Huff" &lt;arlene@contoso.com&gt;</B:From>
            <B:Subject>opaque s + e</B:Subject>
            <B:DateReceived>2007-02-01T06:42:46.015Z</B:DateReceived>
            <B:DisplayTo>Mike Phipps</B:DisplayTo>
            <B:ThreadTopic>opaque s + e</B:ThreadTopic>
            <B:Importance>1</B:Importance>
            <B:Read>1</B:Read>
            <C:Attachments>
               <C:Attachment>
                  <C:DisplayName>smime.p7m</C:DisplayName>

<C:FileReference>RgAAAAA4u8%2fWvU8lQ7GtLlC7V9V3BwCdyWYIRkOHRp2ozB%2f0DXQsAHgM%2bwAFAAA6pk60fq
kEQbWH4Wm%2bnjh7AHgNBA%2bgAAAJ%3a0</C:FileReference>
                  <C:Method>1</C:Method>
                  <C:EstimatedDataSize>9340</C:EstimatedDataSize>
               </C:Attachment>
            </C:Attachments>
            <C:Body>
               <C:Type>4</C:Type>
               <C:EstimatedDataSize>13813</C:EstimatedDataSize>
               <C: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: Arlene Huff &lt;arlene@contoso.com&gt;
To: Mike Phipps &lt;mike@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
```


Copyright © 2008 Microsoft Corporation.

```
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
CZImiZPyLGQBGRYNamluZ2h1YWMwMURPTTEPMA0GA1UEAxMGVGVzdENBAgonJIo2AAAAAAAHMA0G
(Large section of sample data removed)
               </C:Data>
            </C:Body>
            <B:MessageClass>IPM.Note.SMIME</B:MessageClass>
            <B:InternetCPID>20127</B:InternetCPID>
            <B:Flag/>
            <B:ContentClass>urn:content-classes:message</B:ContentClass>
            <C:NativeBodyType>1</C:NativeBodyType>
         </Properties>
      </Fetch>
   </Response>
</ItemOperations>
```

4.5.3 Fetching an E-Mail Item with a LongId

The following example shows the client retrieving an e-mail message by using <LongId>. First, use the **Search** command to get the <LongId> of the message, and then use the <Fetch> command with the <LongId> to retrieve the message.

The client sends the **Search** command request message, and it is searching for e-mails containing the text "Sales Totals".

4.5.3.1 Search Request

```
POST /Microsoft-Server-
ActiveSync?Cmd=Search&User=deviceuser&DeviceId=device1&DeviceType=SmartPhone HTTP/1.1
Content-Type: application/vnd.ms-sync
MS-ASProtocolVersion: 14.0
<?xml version="1.0" encoding="utf-8"?>
<Search xmlns="Search:" xmlns:airsync="AirSync:"
xmlns:email="POOMMAIL:">
  <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>
```


Copyright © 2008 Microsoft Corporation.

```
</Search>
```

The server sends the Search command response message includes e-mail data for e-mail that contains the string "Sales Totals". Included with the results is the <LongId> element.

4.5.3.2 Search Response

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Length: 423
Content-Type: application/vnd.ms-sync
Server: Microsoft-IIS/6.0
MS-Server-ActiveSync: 14.0
Date: Tue, 08 May 2007 17:42:07 GMT

<?xml version="1.0" encoding="utf-8"?><Search xmlns:airsync="AirSync:"
xmlns:email="POOMMAIL:" 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;someone1@example.com&gt;
          </email:To>
          <email:From>"deviceuser2" &lt;someone2@example.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: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>
        </Properties>
      </Result>
      <Range>0-0</Range>
      <Total>1</Total>
    </Store>
  </Response>
</Search>
```


Copyright © 2008 Microsoft Corporation.

The <Fetch> command request is now sent by the client, and includes the <LongId> retrieved by the **Search** command.

4.5.3.3 Fetch Request

```
POST /Microsoft-Server-
ActiveSync?Cmd=ItemOperations&User=deviceuser&DeviceId=device1&
DeviceType=PocketPC HTTP/1.1
Content-Type: application/vnd.ms-sync
MS-ASProtocolVersion: 14.0

<?xml version="1.0" encoding="utf-8"?>
<ItemOperations xmlns:airsync="AirSync:"
xmlns:airsyncbase="AirSyncBase:" xmlns="ItemOperations:">
  <Fetch>
    <Store>Mailbox</Store>
    <airsync:LongId>RgAAAACYWCHnyBZ%2fTq8bujFmR1EPBwBzyWfENpc
    EQ7zUNyaWwM4BAAAA8FxEAABzyWfENpcEQ7zUNyaWwM4BAAAA8HA
    CAAAJ</airsync:LongId>
    <Options>
      <airsyncbase:BodyPreference>
        <airsyncbase:Type>1</airsyncbase:Type>
      </airsyncbase:BodyPreference>
    </Options>
  </Fetch>
</ItemOperations>
```

The server sends the <Fetch> command response, which contains the complete e-mail for the specified message.

4.5.3.4 Fetch Response

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Length: 409
Content-Type: application/vnd.ms-sync
Server: Microsoft-IIS/6.0
MS-Server-ActiveSync: 8.1
Date: Tue, 08 May 2007 17:29:52 GMT

<?xml version="1.0" encoding="utf-8"?><ItemOperations
xmlns:airsync="AirSync:" xmlns:email="POOMMAIL:"
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;someone1@example.com&gt;</email:To>
        <email:From>"deviceuser2" &lt;someone2@example.com&gt;
        </email:From>
        <email:Subject>Sales Totals for April</email:Subject>
        <email:DateReceived>2007-05-08T17:29:07.890Z
        </email:DateReceived>
```


Copyright © 2008 Microsoft Corporation.

```
          <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>
        </Properties>
      </Fetch>
    </Response>
  </ItemOperations>
```

4.5.4 Fetching an Attachment

In the following example, the **Sync** command is used to synchronize a new message with an attachment to the client. Then, the **ItemOperations** command is used to retrieve the attachment.

In the XML scenario 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 >.

4.5.4.1 Sync Request

```
POST /Microsoft-Server-
ActiveSync?Cmd=Sync&User=deviceuser&DeviceId=device1&DeviceType=
PocketPC HTTP/1.1
Content-Type: application/vnd.ms-sync
MS-ASProtocolVersion: 14.0
Content-Length: 106

<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:airsyncbase="AirSyncBase:" xmlns="AirSync:">
  <Collections>
    <Collection>
      <Class>Email</Class>
      <SyncKey>1</SyncKey>
      <CollectionId>7</CollectionId>
      <DeletesAsMoves />
      <GetChanges />
    </Collection>
  </Collections>
</Sync>
```


Copyright © 2008 Microsoft Corporation.

4.5.4.2 Sync Response

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Length: 347
Content-Type: application/vnd.ms-sync
Server: Microsoft-IIS/6.0
MS-Server-ActiveSync: 8.1
Date: Tue, 08 May 2007 17:57:32 GMT

<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns:email="POOMMAIL:"
xmlns:airsyncbase="AirSyncBase:" xmlns="AirSync:">
  <Collections>
    <Collection>
      <Class>Email</Class>
      <SyncKey>2</SyncKey>
      <CollectionId>7</CollectionId>
      <Status>1</Status>
      <Commands>
        <Add>
          <ServerId>7:1</ServerId>
          <ApplicationData>
            <email:To>"deviceuser" &lt;someone@example.com&gt;
            </email:To>
            <email:From>"deviceuser2" &lt;someone2@example.com&gt;
            </email:From>
            <email:Subject>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>
```


Copyright © 2008 Microsoft Corporation.

```
          </ApplicationData>
        </Add>
      </Commands>
    </Collection>
  </Collections>
</Sync>
```

4.5.4.3 ItemOperation Request

```
POST /Microsoft-Server-
ActiveSync?Cmd=ItemOperations&User=deviceuser&DeviceId=device1&
DeviceType=PocketPC HTTP/1.1
Content-Type: application/vnd.ms-sync
MS-ASProtocolVersion: 14.0

<?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>
```

4.5.4.4 ItemOperation Response

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Length: 1151
Content-Type: application/vnd.ms-sync
Server: Microsoft-IIS/6.0
MS-Server-ActiveSync: 8.1
Date: Tue, 08 May 2007 17:28:33 GMT

<?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
        IHJ1biBTb3VyY2UgRGVwb3QsIHlvdXIgY29tcHV0ZXIgbXVzdCBtZ
        WV0IGNlcnRhaW4gaGFyZHdhcmUgYW5kIHNvZnR3YXJlIHJlcXVpc
        mVtZW50cy4gVG8gcnVuIHRoZSBTb3VyY2UgRGVwb3QgY2xpZW50L
        CBjaGVjayB0aGUgZm9sbG93aW5nIGxpc3Qgb2YgbWluaW11bSBw
        bGF0Zm9ybS1zcGVjaWZpYyByZXF1aXJlbWVudHMgZm9yIFNvdXJj
        ZSBEZXBvdC4gDQoNClNvZnR3YXJlIFJlcXVpcmVtZW50cw0KV2luZG9
        3cyBYUCBhbmQgV2luZG93cyAyMDAwIJYgSW50ZWwgb3IgQVhQ
        NjQNCk5UNCAtIEludGVsIA0KV2luZG93cyA5eCCWIEludGVsDQ
        pIYXJkd2FyZSBSZXF1aXJlbWVudHMNCkludGVsIENsaWVudA0K
        DQp4ODYgcHJvY2Vzc29yDQpQZW50aXVtIG9yIGJldHRlciANCkV
```


Copyright © 2008 Microsoft Corporation.

```
          ub3VnaCBoYXJkIGRpc2sgc3BhY2UgdG8gaG9sZCB0aGUgZmlsZX
          MNCkFYUDY0IENsaWVudA0KDQpBbHBoYSBDbGllbnQgc3lzdGVtI
          HJ1bm5pbmcgYW4gQVhQNjQgZW11bGF0b3IuIA0KSG93IHRvDQoN
          Ckluc3RhbGwgU291cmNlIERlcG90DQoNCg0KLS0tLS0tLS0tLS0t
          LS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0
          tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0tLS0NCg0KU2VuZCB
          1cyBmZWVkYmFjayBhbmQgc3VnZ2VzdGlvbnMgb24gdGhpcyBIZW
          xwIHRvcGljIG9yIHRoaXMgcHJvZHVjdC4NCg0KQ29udGFjdCBzdX
          Bwb3J0IGZvciBhc3Npc3RhbmNlLg0KDQoNCmMuIDIwMDQuIE1pY3
          Jvc29mdCBDb25maWRlbnRpYWwgDQpMYXN0IHVwZGF0ZWQgMy8xO
          S8yMDA0IA==</Data>
        </Properties>
      </Fetch>
    </Response>
  </ItemOperations>
```


4.6 Retrieving and Changing OOF Settings by Using the Settings Command

This section provides sample messages related to retrieving and changing OOF settings.

4.6.1 Retrieving OOF Settings

The client requests the user's OOF settings by using the <Get> command and specifying the type in which the client wants to have the OOF message formatted.

4.6.1.1 Request

```
POST /Microsoft-Server-
ActiveSync?Cmd=Settings&User=deviceuser&DeviceId=device1&DeviceType=
PocketPC HTTP/1.1
Content-Type: application/vnd.ms-sync
MS-ASProtocolVersion: 14.0

<?xml version="1.0" encoding="utf-8"?>
<Settings xmlns="Settings:">
  <Oof>
    <Get>
      <BodyType>HTML</BodyType>
    </Get>
  </Oof>
</Settings>
```

The client requested the messages to be returned in HTML, so all OOF messages are formatted as such.

4.6.1.2 Response

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Length: 203
Content-Type: application/vnd.ms-sync
Server: Microsoft-IIS/6.0
MS-Server-ActiveSync: 8.1
Date: Tue, 08 May 2007 17:46:07 GMT
```


Copyright © 2008 Microsoft Corporation.

```
<?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>HTML</BodyType>
      </OofMessage>
      <OofMessage>
        <AppliesToExternalKnown />
        <Enabled>1</Enabled>
        <ReplyMessage>External OOF Message</ReplyMessage>
         <BodyType>HTML</BodyType>
       </OofMessage>
       <OofMessage>
       <AppliesToExternalUnknown /><Enabled>0</Enabled>
       <ReplyMessage>External OOF Message</ReplyMessage>
       <BodyType>HTML</BodyType>
     </OofMessage>
    </Get>
  </Oof>
</Settings>
```

4.6.2 Turning On the OOF Message

The client wants to turn on the OOF message. The client has to update the OOF status by using the <Set> command.

4.6.2.1 Request

```
POST /Microsoft-Server-
ActiveSync?Cmd=Settings&User=deviceuser&DeviceId=device1&DeviceType=PocketPC HTTP/1.1
Content-Type: application/vnd.ms-sync
MS-ASProtocolVersion: 14.0

<?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-
```


Copyright © 2008 Microsoft Corporation.

```
        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>
```

4.6.2.2 Response

The server responds with status, to indicate that OOF was successfully enabled.

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Length: 20
Content-Type: application/vnd.ms-sync
Server: Microsoft-IIS/6.0
MS-Server-ActiveSync: 8.1
Date: Tue, 08 May 2007 17:45:09 GMT

<?xml version="1.0" encoding="utf-8"?>
<Settings xmlns="Settings:">
  <Status>1</Status>
  <Oof>
    <Status>1</Status>
  </Oof>
</Settings>
```

4.6.3 Turning Off the OOF Message

The client wants to turn off the OOF message. The client has to update the OOF status by using the <Set> command.

4.6.3.1 Request

```
POST /Microsoft-Server-
ActiveSync?Cmd=Settings&User=deviceuser&DeviceId=device1&DeviceType=PocketPC HTTP/1.1
```


Copyright © 2008 Microsoft Corporation.

```
Content-Type: application/vnd.ms-sync
MS-ASProtocolVersion: 14.0

<?xml version="1.0" encoding="utf-8"?>
<Settings xmlns="Settings:">
  <Oof>
    <Set>
      <OofState>0</OofState>
    </Set>
  </Oof>
</Settings>
```

4.6.3.2 Response

The server responds with status, to indicate that OOF was successfully disabled.

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Length: 20
Content-Type: application/vnd.ms-sync
Server: Microsoft-IIS/6.0
MS-Server-ActiveSync: 8.1
Date: Tue, 08 May 2007 17:45:09 GMT

<?xml version="1.0" encoding="utf-8"?>
<Settings xmlns="Settings:">
  <Status>1</Status>
  <Oof>
    <Status>1</Status>
  </Oof>
</Settings>
```

4.7 Retrieving User Information by Using the Settings Command

The following example shows a user-information request and response.

4.7.1 Request

```
<Settings>
    <UserInformation>
        <Get/>
    </UserInformation>
</Settings>
```

4.7.2 Response

```
<Settings>
    <Status>1</Status>
    <UserInformation>
        <Status>1</Status>
        <Get>
            <EmailAddresses>
                <SMTPAddress>nameA@microsoft.com</SMTPAddress>
                <SMTPAddress>firstB.lastB@microsoft.com</SMTPAddress>
```


Copyright © 2008 Microsoft Corporation.

```
            </EmailAddresses>
        </Get>
    </UserInformation>
</Settings>
```

4.8 Setting Device Information by Using the Settings Command

The following example shows a device-information request and response.

4.8.1 Request

```
<Settings xmlns="Settings:">
    <DeviceInformation>
        <Set>
            <Model>...</Model>
            <IMEI>...</IMEI>
            <FriendlyName>...</FriendlyName>
            <OS>...</OS>
            <OSLanguage>...</OSLanguage>
            <PhoneNumber>...</PhoneNumber>
            <MobileOperator>...</MobileOperator>
            <UserAgent>...</UserAgent>
        </Set>
    </DeviceInformation>
</Settings>
```

4.8.2 Response

```
<Settings xmlns="Settings:">
    <Status>1</Status>
    <DeviceInformation>
        <Set>
            <Status>...</Status>
        </Set>
    </DeviceInformation>
</Settings>
```

4.9 Setting a Device Password by Using the Settings Command

The following example shows a device-password request and response.

4.9.1 Request

```
<Settings>
    <DevicePassword>
        <Set>
            <Password>bar</Password>
        </Set>
    </DevicePassword>
</Settings>
```


Copyright © 2008 Microsoft Corporation.

4.9.2 Response

```
<Settings>
    <Status>1</Status>
    <DevicePassword>
        <Set>
            <Status>...</Status>
        </Set>
    </DevicePassword>
</Settings>
```

4.10 Accessing Documents on File Shares and URIs by Using the Search and ItemOperations Commands

This section shows how to use the following process to retrieve an item from a Windows® SharePoint® Services or UNC site :

Issue a **Search** command, specifying the link to the folder. The server will return folder/item metadata, specifying the ID, file name, size, and so on for the item. For instructions on completing this task, see section 4.10.1.

Issue the **ItemOperations** command, specifying the ID from the item metadata. For instructions on completing this task, see section 4.10.

In issuing request 2, 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.

Copyright © 2008 Microsoft Corporation.

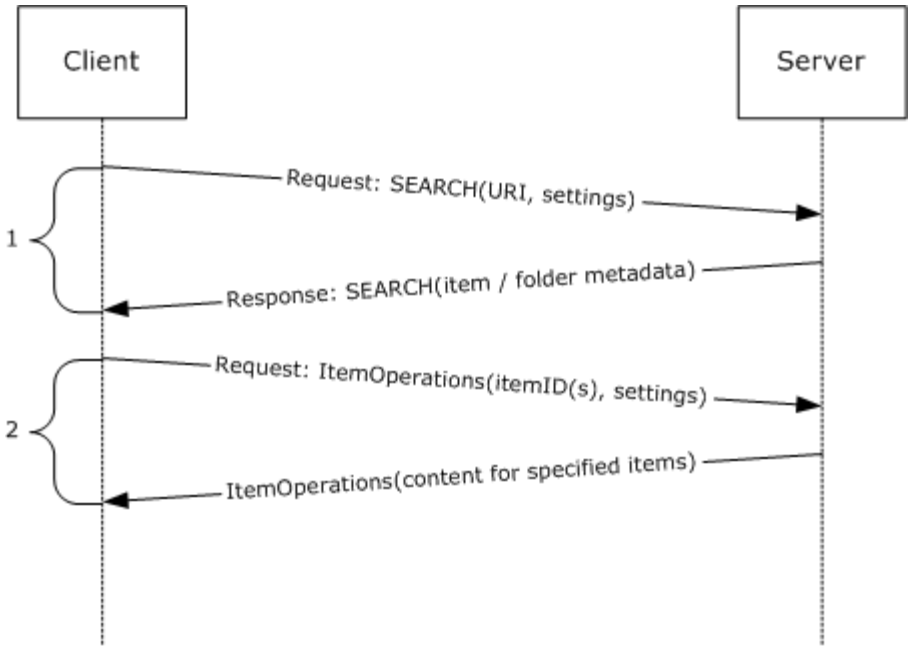

Figure 10: Finding and retrieving an item from a file share or UNC site

4.10.1 Issuing a Search for Item Metadata

As illustrated in the figure, the client first issues a search request 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 <Name> element.
- Specifies the URI as the <Value> in an <EqualTo> query.
- Specifies the range of results that the client wants to have returned in the response.

In this case, the client is attempting to retrieve metadata for the files in a UNC share.

4.10.1.1 Request

```
POST /Microsoft-Server-
ActiveSync?Cmd=Search&User=deviceuser&DeviceId=device1&DeviceType=
SmartPhone HTTP/1.1
Content-Type: application/vnd.ms-sync
MS-ASProtocolVersion: 14.0

<?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>
```


Copyright © 2008 Microsoft Corporation.

```
      </EqualTo>
    </Query>
    <Options>
      <Range>0-999</Range>
    </Options>
  </Store>
</Search>
```

4.10.1.2 Response

The response from the server contains the metadata for the folder and items. The very first node in the response is the top-level node, followed by its children (if any).

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Length: 529
Content-Type: application/vnd.ms-sync
Server: Microsoft-IIS/6.0
MS-Server-ActiveSync: 8.1
Date: Tue, 08 May 2007 17:28:25 GMT
<?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>
```


Copyright © 2008 Microsoft Corporation.

```
          <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>
```

4.10.2 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 and specifying the <LinkId> of the item. In this example, the client also specifies that the client only requires bytes from 10 through 19 of the item returned in this request.

4.10.2.1 Request

```
POST /Microsoft-Server-
ActiveSync?Cmd=ItemOperations&User=deviceuser&DeviceId=device1&
DeviceType=SmartPhone HTTP/1.1
Content-Type: application/vnd.ms-sync
MS-ASProtocolVersion: 14.0

<?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>
```

4.10.2.2 Response

The response from the server contains the requested item. The binary content of the file is Base64-encoded and is included in the <Data> element.

Copyright © 2008 Microsoft Corporation.

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Length: 167
Content-Type: application/vnd.ms-sync
Server: Microsoft-IIS/6.0
MS-Server-ActiveSync: 8.1
Date: Tue, 08 May 2007 17:28:53 GMT

<?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>
```

4.11 Searching for an Item in the Mailbox by Using the Search Command

This section provides sample messages used to perform keyword searches and forward search results for items in the mailbox.

4.11.1 Keyword Search

In the following example, the client is searching the Inbox in the mailbox by using the keyword "Presentation". The client has asked for the first 5 results and specified that it wants text bodies returned for the results. Note that the content of the <FreeText> element is not case-sensitive.

4.11.1.1 Request

```
POST /Microsoft-Server-
ActiveSync?Cmd=Search&User=deviceuser&DeviceId=device1&DeviceType=SmartPhone HTTP/1.1
Content-Type: application/vnd.ms-sync
MS-ASProtocolVersion: 14.0

<?xml version="1.0" encoding="utf-8"?>
<Search xmlns="Search:" xmlns:airsync="AirSync:"
xmlns:email="POOMMAIL:">
<Store>
  <Name>Mailbox</Name>
    <Query>
      <And>
        <airsync:Class>Email</airsync:Class>
        <airsync:CollectionId>7</airsync:CollectionId>
        <FreeText>Presentation</FreeText>
      </And>
```


Copyright © 2008 Microsoft Corporation.

```
      </Query>
      <Options>
        <RebuildResults />
        <Range>0-4</Range>
      </Options>
    </Store>
  </Search>
```

4.11.1.2 Response

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Length: 423
Content-Type: application/vnd.ms-sync
Server: Microsoft-IIS/6.0
MS-Server-ActiveSync: 14.0
Date: Tue, 08 May 2007 17:42:07 GMT

<?xml version="1.0" encoding="utf-8"?><Search xmlns:airsync="AirSync:"
xmlns:email="POOMMAIL:" xmlns:airsyncbase="AirSyncBase:"
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;someone1@example.com&gt;
          </email:To>
          <email:From>"deviceuser2"&lt;someone2@example.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: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>
        </Properties>
      </Result>
      <Range>0-0</Range>
      <Total>1</Total>
    </Store>
```


Copyright © 2008 Microsoft Corporation.

```
    </Response>
  </Search>
```

4.11.2 Forward a Search Result

The client can then take the <LongId> for any given search result and forward the item.

```
POST Microsoft-Server-
ActiveSync?User=rich&DeviceId=6F24CAD599A5BF1A690246B8C68F
AE8D&DeviceType=PocketPC&Cmd=SmartForward
MS-ASProtocolVersion: 14.0
Content-Type: message/rfc822
X-MS-PolicyKey: 3942919513

<?xml version="1.0" encoding="utf-8"?>
<SmartForward xmlns="ComposeMail:">
  <ClientId>40a864cb-79d4-4b44-8f50-0da373ee9377</ClientId>
  <Source>
<LongId>RgAAAACYWCHnyBZ%2fTq8bujFmR1EPBwBzyWfENpcEQ7zUNyaWwM4BAAAA8FxEAABzyWfENpcEQ7zUNyaWwM4
BAAAA8HACAAAJ</LongId>
  </Source>
  <Mime>
MIME-Version: 1.0
content-class:
From:
Subject: FW: rx
Date: Thu, 27 Apr 2006 13:11:01 -0800
Importance: normal
X-Priority: 3
To: <rich@adventure-works.com>
Content-Transfer-Encoding: quoted-printable
Content-Type: text/plain; charset="utf-8"

Please take a look at this presentation…
</Mime>
</SmartForward>


HTTP/1.1 200 OK
Connection: Keep-Alive
Content-Length: 0
Date: Thu, 27 Apr 2006 20:11:11 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
MS-Server-ActiveSync: 14.0
Cache-Control: private
```

4.12 Resolving Recipients and Retrieving Free/Busy Data by Using the ResolveRecipients Command

This section provides sample messages related to the **ResolveRecipients** command.

Copyright © 2008 Microsoft Corporation.

4.12.1 Response for a GAL Entry

The following example shows two recipients that are being returned to the client. In the "Testers" distribution list, there are three recipients but only two have valid certificates.

```
<?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>
    </Recipient>
    <Recipient>
     ...
    </Recipient>
  </Response>
<ResolveRecipients>
```

4.12.2 Response for a Contact Entry

The following example shows a response for a contact entry.

```
<Response>
  <To>Contact</To>
  <Status>1</Status>
  <RecipientCount>1</RecipientCount>
  <Recipient>
    <Type>2</Type>
    <DisplayName>James Smith</DisplayName>
    <EmailAddress>jsmith@example.com</EmailAddress>
  </Recipient>
</Response>
```

4.12.3 Retrieving Free/Busy Data By Using the ResolveRecipients Command

The following examples show a sample request and response in which the free/busy data for two users and two distribution lists are retrieved.

4.12.3.1 Request to Retrieve Free/Busy Data

The following example shows the request used 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"?>
```


Copyright © 2008 Microsoft Corporation.

```
<ResolveRecipients xmlns="ResolveRecipients:">
  <To>all@contoso.com</To>
  <To>ryan@contoso.com</To>
  <To>Tom</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>
```

4.12.3.2 Response with MergedFreeBusy Data

The following example shows the **ResolveRecipients** response issued for the request in section 4.12.3.1.

As shown in the example, the free/busy data for the all@contoso.com distribution list could not be retrieved and <Status> value 162 was returned. The free/busy data for Ryan Calafato was returned successfully. Two ambiguous recipient suggestions were returned for "Tom", neither of which contain the <Availability> element, as it is returned only when an exact match is found. And, the personal distribution list returned a variety of successful and non-successful 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>Ryan Calafato</DisplayName>
      <EmailAddress>ryan@contoso.com</EmailAddress>
      <Availability>
        <Status>1</Status>
        <MergedFreeBusy>0020000000000000000000000010020022000000100000000</MergedFreeBusy>
      </Availability>
    </Recipient>
  </Response>
  <Response>
```


Copyright © 2008 Microsoft Corporation.

```
      <To>tom</To>
      <Status>3</Status>
      <RecipientCount>30</RecipientCount>
      <Recipient>
        <Type>2</Type>
        <DisplayName>Tom Getzinger </DisplayName>
        <EmailAddress>tomget@contoso.com</EmailAddress>
      </Recipient>
      <Recipient>
        <Type>1</Type>
        <DisplayName>Tom Higginbotham (Sr.)</DisplayName>
        <EmailAddress>tomhig@contoso.com</EmailAddress>
      </Recipient>
    </Response>
    <Response>
      <To>myPersonalDistributionList</To>
      <Status>1</Status>
      <RecipientCount>4</RecipientCount>
      <Recipient>
        <Type>2</Type>
        <DisplayName>glen@adventureworks.com</DisplayName>
        <EmailAddress>glen@adventureworks.com</EmailAddress>
        <Availability>
          <Status>162</Status>
        </Availability>
      </Recipient>
      <Recipient>
        <Type>1</Type>
        <DisplayName>Tom Higginbotham (Sr.)</DisplayName>
        <EmailAddress>tomhig@contoso.com</EmailAddress>
        <Availability>
          <Status>161</Status>
        </Availability>
      </Recipient>
      <Recipient>
        <Type>2</Type>
        <DisplayName>Steve Riley</DisplayName>
        <EmailAddress>steve@contoso.com</EmailAddress>
        <Availability>
          <Status>1</Status>
          <MergedFreeBusy>333333333333333333300000010020022000000100000000</MergedFreeBusy>
        </Availability>
      </Recipient>
      <Recipient>
        <Type>2</Type>
        <DisplayName>bonnie@adventureworks.com</DisplayName>
        <EmailAddress>bonnie@adventureworks.com</EmailAddress>
        <Availability>
          <Status>162</Status>
        </Availability>
      </Recipient>
    </Response>
  </ResolveRecipients>
```


Copyright © 2008 Microsoft Corporation.

4.13 Using the Supported Element and Ghosted Elements in the Sync Command

This section provides sample messages related to ghosted contact elements and use of the <Supported> element. Many elements in the Contact and Calendar class can be ghosted, as specified in [MS-ASCNTC] and [MS-ASCAL]. When a property is ghosted, its value is retained on the server even when the client sends a **Sync** request with a <Change> block for only a subset of the class elements. Values for non-ghosted elements are deleted from the server if a value is not specified in the **Sync** request <Change> block.

The <Supported> element is included in **Sync** command requests to inform the server that the client is only keeping track of the elements included as children of the <Supported> element and is not tracking the values of the rest of the class elements.

The example in this section shows the communication between the client and server when the <Supported> element is used, when the client makes changes to the <Supported> elements, and when the server makes changes to the Contacts class.

4.13.1 Initial Folder Sync

The following examples show the initial **FolderSync** command request and response. The **FolderSync** request uses a <SyncKey> value of "0" to indicate an initial synchronization. The **FolderSync** response includes information to populate the user folders on the client device: the folder <DisplayName> values, <ServerId> values, parent folder (<ParentId>) values, and folder <Type> values.

4.13.1.1 Request

```
POST /Microsoft-Server-
ActiveSync?Cmd=FolderSync&User=deviceuser&DeviceId=v140Device&DeviceType=PPC HTTP/1.1
Content-Type: application/vnd.ms-sync.wbxml
MS-ASProtocolVersion: 14.0
User-Agent: ASOM
Host: contoso.com
Content-Length: 13
Cache-Control: no-cache
Authorization: Basic YXJocnB6LWRvbVxzLmJvbGxlczpKJHAxdGVy

<?xml version="1.0" encoding="utf-8"?>
<FolderSync  xmlns="FolderHierarchy:">
  <SyncKey>0</SyncKey>
</FolderSync>
```

4.13.1.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/vnd.ms-sync.wbxml
X-MS-RP: 2.0,2.1,2.5,12.0,12.1,14.0
MS-ASProtocolVersions: 2.0,2.1,2.5,12.0,12.1,14.0
MS-ASProtocolCommands:
Sync,SendMail,SmartForward,SmartReply,GetAttachment,GetHierarchy,CreateCollection,DeleteColle
ction,MoveCollection,FolderSync,FolderCreate,FolderDelete,FolderUpdate,MoveItems,GetItemEstim
ate,MeetingResponse,Search,Settings,Ping,ItemOperations,Provision,ResolveRecipients,ValidateC
ert
Date: Wed, 01 Apr 2009 06:33:13 GMT
Content-Length: 346
```


Copyright © 2008 Microsoft Corporation.

```
<?xml version="1.0" encoding="utf-8"?><FolderSync xmlns="FolderHierarchy:">
  <Status>1</Status>
  <SyncKey>1</SyncKey>
  <Changes>
    <Count>11</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>
      <DisplayName>Drafts</DisplayName>
      <Type>3</Type>
    </Add>
    <Add>
      <ServerId>5</ServerId>
      <ParentId>0</ParentId>
      <DisplayName>Inbox</DisplayName>
      <Type>2</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>
```


Copyright © 2008 Microsoft Corporation.

```
    <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>
  </Changes>
</FolderSync>
```

<70>

4.13.2 Sync Command

The following examples show the initial **Sync** command request and response. The <Supported> element is included in the request with two child elements, <JobTitle> and <Department>, to indicate to the server that these two elements are being tracked by the client. Note that the <SyncKey> value is set to "0" when the <Supported> element is included in the request. If the <SyncKey> value is set to a non-zero value and the <Supported> element is included, a <Status> value of "4" is returned from the server to indicate a protocol error.

The **Sync** command response indicates that the **Sync** request was processed successfully.

4.13.2.1 Request

```
POST /Microsoft-Server-ActiveSync?Cmd=Sync&User=deviceuser&DeviceId=v140Device&DeviceType=PPC
HTTP/1.1
Content-Type: application/vnd.ms-sync.wbxml
MS-ASProtocolVersion: 14.0
User-Agent: ASOM
Host: exh-b-252
Content-Length: 20
Cache-Control: no-cache
Authorization: Basic YXJocnB6LWRvbVxzLmJvbGxlczpKJHAxdGVy

<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:" xmlns:A1="POOMCONTACTS:">
  <Collections>
    <Collection>
      <SyncKey>0</SyncKey>
      <CollectionId>2</CollectionId>
      <Supported>
        <A1:JobTitle/>>
        <A1:Department/>>
      </Supported>
    </Collection>
  </Collections>
</Sync>
```


Copyright © 2008 Microsoft Corporation.

4.13.2.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/vnd.ms-sync.wbxml
Date: Wed, 01 Apr 2009 06:35:02 GMT
Content-Length: 33

<?xml version="1.0" encoding="utf-8"?><Sync xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>878266863</SyncKey>
      <CollectionId>2</CollectionId>
      <Status>1</Status>
    </Collection>
  </Collections>
</Sync>
```

4.13.3 Sync Contacts

The following examples show the **Sync** command request and response for the Contacts class. The request includes the <CollectionId> value of "2", which corresponds to the contacts folder as created in section 4.13.1.2.

The **Sync** command response indicates that the **Sync** request was processed successfully.

4.13.3.1 Request

```
POST /Microsoft-Server-ActiveSync?Cmd=Sync&User=deviceuser&DeviceId=v140Device&DeviceType=PPC
HTTP/1.1
Content-Type: application/vnd.ms-sync.wbxml
MS-ASProtocolVersion: 14.0
User-Agent: ASOM
Host: exh-b-252
Content-Length: 20
Cache-Control: no-cache
Authorization: Basic YXJocnB6LWRvbVxzLmJvbGxlczpKJHAxdGVy

<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:" xmlns:A1="POOMCONTACTS:">
  <Collections>
    <Collection>
      <SyncKey>878266863</SyncKey>
      <CollectionId>2</CollectionId>
      <DeletesAsMoves/>>
      <GetChanges/>>
    </Collection>
  </Collections>
</Sync>
```

4.13.3.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/vnd.ms-sync.wbxml
Date: Wed, 01 Apr 2009 06:38:34 GMT
Content-Length: 448
```


Copyright © 2008 Microsoft Corporation.

```
<?xml version="1.0" encoding="utf-8"?><Sync xmlns:A1="POOMCONTACTS:"
xmlns:A12="POOMCONTACTS2:" xmlns:A17="AirSyncBase:" xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>619052475</SyncKey>
      <CollectionId>2</CollectionId>
      <Status>1</Status>
      <Commands>
        <Add>
          <ServerId>2:1</ServerId>
          <ApplicationData>
            <A17:Body>
              <A17:Type>1</A17:Type>
              <A17:EstimatedDataSize>0</A17:EstimatedDataSize>
              <A17:Truncated>1</A17:Truncated>
            </A17:Body>
            <A1:WebPage>http://contoso.com</A1:WebPage>
            <A1:BusinessCountry>USA</A1:BusinessCountry>
            <A1:Department>Executive</A1:Department>
            <A1:Email1Address>"president@contoso.com"
&lt;president@contoso.com&gt;</A1:Email1Address>
            <A1:FileAs>Hassall, Mark</A1:FileAs>
            <A1:FirstName>Mark</A1:FirstName>
            <A1:HomeCity>Seattle</A1:HomeCity>
            <A1:HomeCountry>USA</A1:HomeCountry>
            <A1:HomePhoneNumber>(206) 555-0100</A1:HomePhoneNumber>
            <A1:HomePostalCode>98000</A1:HomePostalCode>
            <A1:HomeState>WA</A1:HomeState>
            <A1:HomeStreet>234 Main Street</A1:HomeStreet>
            <A1:BusinessCity>Seattle</A1:BusinessCity>
            <A1:MiddleName>I</A1:MiddleName>
            <A1:MobilePhoneNumber>(206) 555-0101</A1:MobilePhoneNumber>
            <A1:CompanyName>Contoso Inc.</A1:CompanyName>
            <A1:BusinessPostalCode>98000</A1:BusinessPostalCode>
            <A1:AssistantName>Andy Jacobs</A1:AssistantName>
            <A1:AssistantTelephoneNumber>(206) 555-0102</A1:AssistantTelephoneNumber>
            <A1:LastName>Hassall</A1:LastName>
            <A1:BusinessState>WA</A1:BusinessState>
            <A1:BusinessStreet>123 Main Street</A1:BusinessStreet>
            <A1:BusinessPhoneNumber>(206) 555-0103</A1:BusinessPhoneNumber>
            <A1:JobTitle>President</A1:JobTitle>
            <A1:OfficeLocation>TopFloor</A1:OfficeLocation>
            <A12:ManagerName>Roya Asbari</A12:ManagerName>
            <A17:NativeBodyType>1</A17:NativeBodyType>
          </ApplicationData>
        </Add>
      </Commands>
    </Collection>
  </Collections>
</Sync>
```


4.13.4 Sync Client Changes

The following example shows the **Sync** command request and response when the client updates the values for the <JobTitle> and <Department> elements. Because all the other contact properties are ghosted, they are not deleted when server processes this request.

The **Sync** command response indicates that the **Sync** request was processed successfully.

Copyright © 2008 Microsoft Corporation.

4.13.4.1 Request

```
POST /Microsoft-Server-ActiveSync?Cmd=Sync&User=deviceuser&DeviceId=v140Device&DeviceType=PPC
HTTP/1.1
Content-Type: application/vnd.ms-sync.wbxml
MS-ASProtocolVersion: 14.0
User-Agent: ASOM
Host: contoso.com
Content-Length: 20
Cache-Control: no-cache
Authorization: Basic YXJocnB6LWRvbVxzLmJvbGxlczpKJHAxdGVy

<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:" xmlns:A1="POOMCONTACTS:">
  <Collections>
    <Collection>
      <SyncKey>619052475</SyncKey>
      <CollectionId>2</CollectionId>
      <Commands>
        <Change>
          <ServerID>2:1</ServerID>
          <ApplicationData>
            <A1:JobTitle>Sales Manager</A1:JobTitle>
            <A1:Department>Marketing</A1:Department>
          </ApplicationData>
        </Change>
      </Commands>
    </Collection>
  </Collections>
</Sync>
```

4.13.4.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/vnd.ms-sync.wbxml
Date: Wed, 01 Apr 2009 06:51:02 GMT
Content-Length: 33

<?xml version="1.0" encoding="utf-8"?><Sync xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>716498022</SyncKey>
      <CollectionId>2</CollectionId>
      <Status>1</Status>
    </Collection>
  </Collections>
</Sync>
```

4.13.5 Sync Server Changes

The following **Sync** command request and response show the **Sync** command request and response when the <Manager> value is changed and the <AssistantName> value is deleted on the server.

Copyright © 2008 Microsoft Corporation.

4.13.5.1 Request

```
POST /Microsoft-Server-ActiveSync?Cmd=Sync&User=deviceuser&DeviceId=v140Device&DeviceType=PPC
HTTP/1.1
Content-Type: application/vnd.ms-sync.wbxml
MS-ASProtocolVersion: 14.0
User-Agent: ASOM
Host: exh-b-252
Content-Length: 20
Cache-Control: no-cache
Authorization: Basic YXJocnB6LWRvbVxzLmJvbGxlczpKJHAxdGVy

<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:" xmlns:A1="POOMCONTACTS:">
  <Collections>
    <Collection>
      <SyncKey>716498022</SyncKey>
      <CollectionId>2</CollectionId>
      <DeletesAsMoves/>
      <GetChanges/>
    </Collection>
  </Collections>
</Sync>
```

4.13.5.2 Response

```
HTTP/1.1 200 OK
Content-Type: application/vnd.ms-sync.wbxml
Date: Wed, 01 Apr 2009 06:55:21 GMT
Content-Length: 424

<?xml version="1.0" encoding="utf-8"?><Sync xmlns:A1="POOMCONTACTS:" xmlns:A17="AirSyncBase:"
xmlns="AirSync:">
  <Collections>
    <Collection>
      <SyncKey>103384063</SyncKey>
      <CollectionId>2</CollectionId>
      <Status>1</Status>
      <Commands>
        <Change>
          <ServerId>2:1</ServerId>
          <ApplicationData>
            <A17:Body>
              <A17:Type>1</A17:Type>
              <A17:EstimatedDataSize>0</A17:EstimatedDataSize>
              <A17:Truncated>1</A17:Truncated>
            </A17:Body>
            <A1:WebPage>http://contoso.com</A1:WebPage>
            <A1:BusinessCountry>USA</A1:BusinessCountry>
            <A1:Department>Marketing</A1:Department>
            <A1:Email1Address>"president@contoso.com"
&lt;president@contoso.com&gt;</A1:Email1Address>
            <A1:FileAs>Hassall, Mark</A1:FileAs>
            <A1:FirstName>Mark</A1:FirstName>
            <A1:HomeCity>Seattle</A1:HomeCity>
            <A1:HomeCountry>USA</A1:HomeCountry>
            <A1:HomePhoneNumber>(206) 555-0100</A1:HomePhoneNumber>
            <A1:HomePostalCode>98000</A1:HomePostalCode>
```


Copyright © 2008 Microsoft Corporation.

```
            <A1:HomeState>WA</A1:HomeState>
            <A1:HomeStreet>234 Main Street</A1:HomeStreet>
            <A1:BusinessCity>Seattle</A1:BusinessCity>
            <A1:MiddleName>I</A1:MiddleName>
            <A1:MobilePhoneNumber>(206) 555-0101</A1:MobilePhoneNumber>
            <A1:CompanyName>Contoso Inc.</A1:CompanyName>
            <A1:BusinessPostalCode>98000</A1:BusinessPostalCode>
            <A1:AssistantTelephoneNumber>(206) 555-0102</A1:AssistantTelephoneNumber>
            <A1:LastName>Hassall</A1:LastName>
            <A1:BusinessState>WA</A1:BusinessState>
            <A1:BusinessStreet>123 Main Street</A1:BusinessStreet>
            <A1:BusinessPhoneNumber>(206) 555-0103</A1:BusinessPhoneNumber>
            <A1:JobTitle>Sales Manager</A1:JobTitle>
            <A1:OfficeLocation>TopFloor</A1:OfficeLocation>
            <A12:ManagerName>Carole Poland</A12:ManagerName>
            <A17:NativeBodyType>1</A17:NativeBodyType>
          </ApplicationData>
        </Change>
      </Commands>
    </Collection>
  </Collections>
</Sync>
```


Copyright © 2008 Microsoft Corporation.

5 Security

5.1 Security Considerations for Implementers

The 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	2.2.1.12
ValidateCert Command	2.2.1.20

Copyright © 2008 Microsoft Corporation.

6 Appendix A: Product Behavior

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

- Microsoft Exchange Server 2007
- Microsoft Exchange Server 2010

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

Unless otherwise specified, any statement of optional behavior in this specification prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that product does not follow the prescription.

<1> Section 2.2.1.1: For details about the **Autodiscover** service, see [AUTODISCOVER]. For more details about the Autodiscover HTTP Service, see [MS-OXDSCLI].

<2> Section 2.2.1.1.2.2: In Exchange 2007, the <Culture> element always returns "en:en", regardless of the culture that is sent by the client.

<3> Section 2.2.1.1.2.12: In Exchange 2007 and Exchange 2010, this value is retrieved from the MobileClientCertTemplateName attribute of the ActiveSync virtual directory object.

<4> Section 2.2.1.1.2.15: In Exchange 2007 and Exchange 2010, the URL is retrieved from the ExternalUrl attribute of the ActiveSync virtual directory object in Active Directory.

<5> Section 2.2.1.1.2.15: The URL value for the CertEnroll Type is retrieved from the MobileClientCertificateAuthorityURL attribute of the ActiveSync virtual directory object in Active Directory.

<6> Section 2.2.1.1.2.16: If the <Type> element value is *MobileSync*, then the <URL> element value is retrieved from the *ExternalUrl* attribute of the ActiveSync virtual directory object in Active Directory that is a child object of the server that services the user's mailbox. If the <Type> element value is *CertEnroll*, then the <URL> value is retrieved from the MobileClientCertificateAuthorityURL attribute of the ActiveSync virtual directory object in Active Directory that is a child object of the server that services the user's mailbox.

<7> Section 2.2.1.6: The **GetAttachment** command is not supported when the MS-ASProtocolVersion header is set to 14.0 in the **GetAttachment** command request. Use the <Fetch> element of the **ItemOperations** command instead.

<8> Section 2.2.1.7.1.3: The <FilterType> element is not a supported child element of <Collection> when the MS-ASProtocolVersion header is set to 14.0.

<9> Section 2.2.1.7.1.4: In MS-ASProtocolVersion 2.0, 2.1, 2.5, 12.0 and 12.1, the <SyncKey> element is placed after the <FilterType> element in a **GetItemEstimate** command request. In MS-ASProtocolVersion 14.0, the <SyncKey> element is the first child element of <Collection> in a **GetItemEstimate** command request.

<10> Section 2.2.1.7.1.6: The <ConversationMode> element is only supported when the MS-ASProtocolVersion header is set to 14.0.

Copyright © 2008 Microsoft Corporation.

<11> Section 2.2.1.7.1.7: The <Options> element is only valid in a **GetItemEstimate** request when the MS-ASProtocolVersion header is set to 14.0.

<12> Section 2.2.1.7.1.8: The <Class> element is only supported within the Options block when the MS-ASProtocolVersion header is set to 14.0. If the MS-ASProtocolVersion is set to 2.0, 2.1, 2.5, 12.0, or 12.1, the <Class> element is a child of the <Collection> block.

<13> Section 2.2.1.7.1.9: The <FilterType> element is only supported within the <Options> block when the MS-ASProtocolVersion header is set to 14.0. If the MS-ASProtocolVersion is set to 2.0, 2.1, 2.5, 12.0, or 12.1, the <Class> element is a child of the <Collection> block.

<14> Section 2.2.1.7.1.9: The <Collection> element is not a supported parent element of <FilterType> when the MS-ASProtocolVersion header is set to 14.0.

<15> Section 2.2.1.7.1.10: The <MaxItems> element is only supported when the MS-ASProtocolVersion header is set to 14.0.

<16> Section 2.2.1.8.2.2: The location is indicated by a server ID (<ServerId> element) if an ActiveSync ID is being used to identify the item.

<17> Section 2.2.1.8.2.17: The <Move> element is only supported when the MS-ASProtocolVersion header is set to 14.0.

<18> Section 2.2.1.8.2.18: The <ConversationId> element is only supported when the MS-ASProtocolVersion header is set to 14.0.

<19> Section 2.2.1.8.2.19: The <DstFldId> element is only supported when the MS-ASProtocolVersion header is set to 14.0.

<20> Section 2.2.1.8.2.20: The <MoveAlways> element is only supported when the MS-ASProtocolVersion header is set to 14.0.

<21> Section 2.2.1.8.3.6: The location is indicated by a server ID (<ServerId> element) if an ActiveSync ID is being used to identify the item.

<22> Section 2.2.1.11.1.1: The *Notes* value is only valid when the MS-ASProtocolVersion header is set to 14.0.

<23> Section 2.2.1.13: Retrival of free/busy information using the <Availability> element in the **ResolveRecipients** command is only supported when the MS-ASProtocolVersion header is set to 14.0.

<24> Section 2.2.1.13.1.6: Some fields that are ANR-indexed 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 it can be extended to include other fields.

<25> Section 2.2.1.13.1.7: The <Availability> element is only supported in MS-ASProtocolVersion 14.0.

<26> Section 2.2.1.13.1.8: The <StartTime> element is only supported in MS-ASProtocolVersion 14.0.

<27> Section 2.2.1.13.1.9: The <EndTime> element is only supported in MS-ASProtocolVersion 14.0.

<28> Section 2.2.1.13.2.11: The <Availability> element is only supported in MS-ASProtocolVersion 14.0.

Copyright © 2008 Microsoft Corporation.

<29> Section 2.2.1.13.2.12: Some fields that are ANR-indexed in Active Directory by default 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.

<30> Section 2.2.1.13.2.14: The <Availability> element is only supported in MS-ASProtocolVersion 14.0.

<31> Section 2.2.1.13.2.15: The <MergedFreeBusy> element is only supported in MS-ASProtocolVersion 14.0.

<32> Section 2.2.1.14: The **Search** command provides support for the following: the ability to search the Exchange mailbox, and the ability to browse the Microsoft Windows SharePoint Services technology Document Libraries or Universal Naming Convention (UNC) Shares. Mailbox and Windows SharePoint Services and UNC are represented as new stores within the **Search** command, and each has associated options, query, and schema.

<33> Section 2.2.1.14.1.2: While the Or keyword is supported in the protocol, Exchange 2007 does not support the <Or> keyword and will always return a SearchTooComplex, status 8. <And> or <Or> operations can only be used as the top level node (immediately under <Query>) – this is currently a server-side limitation.

<34> Section 2.2.1.14.1.2: Properties are indexed by Exchange 2007 Content Indexing (CI).

<35> Section 2.2.1.14.1.9: And operations can only be used as the top level node (immediately under Query) – this is currently a server-side limitation. The server responds with SearchTooComplex, status 8.

<36> Section 2.2.1.14.1.10: <Or> elements can only be used as the top level node (immediately under <Query>) – this is currently a server-side limitation. The server responds with *SearchTooComplex*, status 8.

<37> Section 2.2.1.14.1.11: The <Class> element cannot be under a <Query> or <Or> node, but rather be under the topmost <And> – this is currently a server-side limitation. The server responds with *SearchTooComplex*, status 8.

<38> Section 2.2.1.14.1.11: The following classes are supported for mailbox searches when the MS-ASProtocolVersion header is set to 12.0 or 12.1: Email, Calendar, Contacts, Tasks. The SMS class is only available if the MS-ASProtocolVersion header is set to 14.0.

<39> Section 2.2.1.14.1.14: The <GreaterThan> element cannot be under a <Query> or <Or> node, but rather be under the topmost <And> – this is currently a server-side limitation. The server responds with *SearchTooComplex*, status 8.

<40> Section 2.2.1.14.1.15: The <LessThan> element cannot be under a <Query> or <Or> node, but rather be under the topmost <And> – this is currently a server-side limitation. The server responds with *SearchTooComplex*, status 8.

<41> Section 2.2.1.14.1.17: The <FreeText> element cannot be under a <Query> or <Or> node, but rather be under the topmost <And> – this is currently a server-side limitation. The server responds with *SearchTooComplex*, status 8.

<42> Section 2.2.1.14.1.18: The <CollectionId> element cannot be under a <Query> or <Or> node, but rather be under the topmost <And> – this is currently a server-side limitation. The server responds with *SearchTooComplex*, status 8.

<43> Section 2.2.1.14.1.19: The <ConversationId> element is only valid when the MS-ASPRotocolVersion header is set to 14.0.

Copyright © 2008 Microsoft Corporation.

<44> Section 2.2.1.14.1.19: The <ConversationId> element cannot be under a <Query> or <Or> node, but rather be under the topmost <And> – this is currently a server-side limitation. The server responds with *SearchTooComplex*, status 8.

<45> Section 2.2.1.16: In Exchange 2007 and Exchange 2010, the **Settings** command is used to perform the following operations: get or set the Out of Office (OOF) settings for the user, send device information to the computer that is running Exchange Server for display in the user and IT interfaces, implement the device password—that is, the personal identification number (PIN)—recovery, and retrieve a list of a user's e-mail addresses.

<46> Section 2.2.1.16.1.8: Exchange 2007 and Exchange 2010 require that the reply message for unknown external and known external audiences be the same.

<47> Section 2.2.1.16.1.15: This information is also reflected in Microsoft Office Outlook® Web Access mobile device console.

<48> Section 2.2.1.16.1.26: The <EnableOutboundSMS> element is only valid when the MS-ASProtocolVersion header is set to 14.0.

<49> Section 2.2.1.16.1.27: The <MobileOperator> element is only valid when the MS-ASProtocolVersion header is set to 14.0.

<50> Section 2.2.1.16.2.9: Exchange 2007 requires that the reply message for unknown external and known external audiences be the same.

<51> Section 2.2.1.16.2.16: This information is reflected both in the Microsoft Office Outlook® Web Access mobile device console and the output to administrative tasks (for example, reporting).

<52> Section 2.2.1.19.1.12: The <Supported> element is not supported when the MS-ASProtocolVersion header value is set to 12.1.

<53> Section 2.2.1.19.1.14: The <ConversationMode> element is only supported when the MS-ASProtocolVersion header is set to 14.0.

<54> Section 2.2.1.19.1.26: The <Class> element is only supported within the <Options> block when the MS-ASProtocolVersion header is set to 14.0. If the MS-ASProtocolVersion is set to 2.0, 2.1, or 2.5, the <Class> element is a child of the <Collection> block.

<55> Section 2.2.1.19.1.27: The <MaxItems> element is only valid when the MS-ASProtocolVersion header is set to 14.0.

<56> Section 2.2.2.14: In Exchange 2007, this was an HTTP 400 response.

<57> Section 2.2.2.14: In Exchange 2007, this was an HTTP 400 response, or 500 for **SendMail**.

<58> Section 2.2.2.14: In Exchange 2007, this was an HTTP 500 response.

<59> Section 2.2.2.14: In Exchange 2007, this was an HTTP 503 response.

<60> Section 2.2.2.14: In Exchange 2007, this was an HTTP 403 response.

<61> Section 2.2.2.14: In Exchange 2007, this was an HTTP 507 response.

<62> Section 2.2.2.14: In Exchange 2007, this was an HTTP 500 response, or 403 for **Provision**.

<63> Section 2.2.2.14: In Exchange 2007, this was an HTTP 501 response.

Copyright © 2008 Microsoft Corporation.

<64> Section 2.2.2.14: In Exchange 2007, this was an HTTP 400 response, or 505 for version 1.0 devices.

<65> Section 2.2.2.14: In Exchange 2007, this was an HTTP 449 response, or 403 if there was no policy key header.

<66> Section 2.2.2.14: In Exchange 2007, this was an HTTP 449 response.

<67> Section 2.2.2.14: In Exchange 2007, this was an HTTP 400 or 501 response.

<68> Section 4.2.4: In Exchange 2007 and Exchange 2010, this information is retrieved from Active Directory directory service information on the ActiveSync virtual directory object.

<69> Section 4.2.6: In Exchange 2007 and Exchange 2010, the 401-1.htm Web page that is installed in the Help subdirectory of the Autodiscover physical directory can be configured as shown in this section to provide additional troubleshooting details.

<70> Section 4.13.1.2: The GetHierarchy, CreateCollection, DeleteCollection, and MoveCollection commands are not supported when the MS-ASProtocolVersion HTTP header is set to 14.0.

Copyright © 2008 Microsoft Corporation.

7 Change Tracking

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

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

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

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

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

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

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

- New content added.
- Content update.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.
- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- New content added for template compliance.
- Content updated for template compliance.

Copyright © 2008 Microsoft Corporation.

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

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

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

Protocol syntax refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.

Protocol revision refers to changes made to a protocol that affect the bits that are sent over the wire.

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
1 Introduction	52994 Added notes and tasks to the list of items that can be synchronized.	Y	Content update.
1 Introduction	Added SMS messages to the list of objects that can be synchronized.	Y	Content update.
1.1 Glossary	49550 Added terms and descriptions for "certificate authority (CA)" and "certificate revocation lists (CRL)".	N	New content added.
1.1 Glossary	53580 Per MS-OXGLOS, Wireless Access Protocol (WAP) should actually be Wireless Application Protocol (WAP). Term was updated accordingly.	N	Content update.
1.2.1 Normative References	49550 Added [MSFT-CRL], [RFC3280], and [X509] as normative references.	Y	New content added.
1.4 Relationship to Other Protocols	48662 Added information about the following related protocols: [MS-ASWBXML], [MS-ASNOTE], [MS-ASMS], [MS-ASAIRS], [MS-ASPROV], [MS-OXDSCLI], and [MS-ASCON].	Y	New content added.
1.5 Prerequisites/Preconditions	48661 Added information about underlying protocols.	N	Content update.
2.2.1.1 Autodiscover	49235 Updated information about when an updated URL is retrieved.	Y	Content update.
2.2.1.1.2.2 Culture	Changed "en:en" to "en:us".	Y	Content update.

Copyright © 2008 Microsoft Corporation.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
2.2.1.1.2.6 Error	52785 Added <Response> as parent element and <ErrorCode> as child element.	Y	Content update.
2.2.1.1.2.10 Response	49313 Changed number allowed to "1...1" and removed information about multiple <Response> elements in a single command response.	Y	Content updated due to protocol revision.
2.2.1.1.2.12 ServerData	Updated product behavior note to include Exchange 2010.	Y	Product behavior note updated.
2.2.1.1.2.14 Status	49029 Added details about the success code.	Y	Content update.
2.2.1.1.2.14 Status	48670 Updated sentence about handling unknown <Status> values.	Y	Content update.
2.2.1.1.2.15 Type	Updated product behavior note to include Exchange 2010.	Y	Product behavior note updated.
2.2.1.1.2.18 ErrorCode	52785 Added new section.	Y	New content added.
2.2.1.2.1 Request	49159 Changed the value of status code returned when the request does not adhere to the schema to "10".	Y	Content update.
2.2.1.2.1.2 SyncKey	49454 Updated information about <SyncKey> mismatch.	Y	Content update.
2.2.1.2.1.2 SyncKey	49486 Changed number allowed in response to 0...1.	Y	Content update.
2.2.1.2.1.5 Type	48670 Removed value 19 as it is a default folder type and cannot be used by the FolderCreate command.	Y	Content removed.
2.2.1.2.1.5 Type	Removed Type value 18 as it cannot be used by the FolderCreate command.	Y	Content removed.
2.2.1.2.2.2 ServerId	49532 Changed two instances of the word "object" to "folder".	N	Content update.
2.2.1.2.2.2	49536	Y	Content

Copyright © 2008 Microsoft Corporation.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
ServerId	Clarified that the folder being located is on the client, changed instances of "server ID" to "<ServerId>", and removed details about internal client behavior.		update.
2.2.1.2.2.4 SyncKey	49454 Updated information about <SyncKey> mismatch.	Y	Content update.
2.2.1.2.2.4 SyncKey	49486 Changed number allowed in response to 0...1.	Y	Content update.
2.2.1.3.1 Request	52986 Updated <SyncKey> and <ServerId> definitions.	Y	Content update.
2.2.1.3.1.2 SyncKey	49454 Updated information about <SyncKey> mismatch.	Y	Content update.
2.2.1.3.1.2 SyncKey	49486 Changed number allowed in response to 0...1.	Y	Content update.
2.2.1.3.1.3 ServerId	49532 Changed an instance of the word "object" to "folder".	N	Content update.
2.2.1.3.2.2 SyncKey	49454 Updated information about <SyncKey> mismatch.	Y	Content update.
2.2.1.3.2.2 SyncKey	49486 Changed number allowed in response to 0...1.	Y	Content update.
2.2.1.4.1 Request	52986 Updated SyncKey definition.	Y	Content update.
2.2.1.4.1.2 SyncKey	49454 Updated information about <SyncKey> mismatch.	Y	Content update.
2.2.1.4.2 Response	49492 Added minOccurs and maxOccurs values.	Y	Content update.
2.2.1.4.2.3 SyncKey	49454 Updated information about <SyncKey> mismatch.	Y	Content update.
2.2.1.4.2.3 SyncKey	49492 Changed number allowed to 0...1 (optional).	Y	Content update.
2.2.1.4.2.4 Changes	49492 Changed number allowed to 0...1 (optional).	Y	Content update.
2.2.1.4.2.5 Count	49491 Changed number allowed to 0...1 (optional).	Y	Content update.
2.2.1.4.2.8 ServerId	49492 Changed number allowed to 0...N (optional).	Y	Content update.
2.2.1.4.2.9	49492	Y	Content

Copyright © 2008 Microsoft Corporation.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
ParentId	Changed number allowed to 0...N (optional).		update.
2.2.1.4.2.10 DisplayName	49492 Changed number allowed to 0...N (optional).	Y	Content update.
2.2.1.4.2.11 Type	49492 Changed number allowed to 0...N (optional).	Y	Content update.
2.2.1.5.1 Request	52986 Changed <SyncKey>, <ServerId>, <ParentId>, and <DisplayName> definitions.	Y	Content update.
2.2.1.5.1.2 SyncKey	49454 Updated information about <SyncKey> mismatch.	Y	Content update.
2.2.1.5.2.2 Status	47854 Changed description for <Status> value 2.	Y	Content update.
2.2.1.5.2.3 SyncKey	49454 Updated information about <SyncKey> mismatch.	Y	Content update.
2.2.1.6 GetAttachment	Added version support information.	Y	Content update.
2.2.1.6.2 Response	Added information about the response.	Y	Content update.
2.2.1.7.1 Request	Added the child elements of <Options> to the XSD.	Y	New content added.
2.2.1.7.1 Request	51961 Changed the maxOccurs value of the <Options> element to 2.	Y	Content update.
2.2.1.7.1.3 Collection	49644 Changed number allowed to 1...300.	Y	Content update.
2.2.1.7.1.3 Collection	52986 Changed product behavior note to specify MS-ASProtocolVersion header value.	Y	Content update.
2.2.1.7.1.4 SyncKey	49454 Updated information about <SyncKey> mismatch.	Y	Content update.
2.2.1.7.1.4 SyncKey	52986 Updated product behavior note to specify behavior in 12.1 and 14.0.	Y	Product behavior note updated.
2.2.1.7.1.6 ConversationMode	52986 Updated product behavior note to specify MS-ASProtocolVersion value.	Y	Content update.
2.2.1.7.1.6	Changed "multiple classes" to "multiple	Y	Content

Copyright © 2008 Microsoft Corporation.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
ConversationMode	collections".		update.
2.2.1.7.1.7 Options	49527 Changed product behavior note to specify MS-ASProtocolVersion value.	Y	Product behavior note updated.
2.2.1.7.1.7 Options	51961 Changed maximum allowed to 2.	Y	Content update.
2.2.1.7.1.8 Class	49527 Changed product behavior note to include ActiveSync version instead of Exchange Server version.	Y	Content update.
2.2.1.7.1.8 Class	49648 Updated product behavior note to include 14.0.	Y	Content update.
2.2.1.7.1.9 FilterType	49658 Updated text to identify specific error values and added column for contact values.	Y	Content update.
2.2.1.7.1.9 FilterType	49648 Added product behavior note.	Y	New product behavior note added.
2.2.1.7.1.10 MaxItems	52986 Changed product behavior note to specify MS-ASProtocolVersion value.	Y	Product behavior note updated.
2.2.1.7.2.4 Collection	49650 Updated number allowed value.	Y	Content update.
2.2.1.8.1 Delivery of Content Requested by Fetch	49540 Changed DeviceId in example from v120 to v140.	N	Content update.
2.2.1.8.2 Request	49546 Removed content about status 4 and sending responses.	Y	Content removed.
2.2.1.8.2 Request	49635 Added minOccurs and maxOccurs values to <Username> and <Password>.	Y	Content update.
2.2.1.8.2.2 Fetch	49585 Removed duplicate information about Inline and Multipart content.	N	Content update.
2.2.1.8.2.2 Fetch	49586 Removed information about errors caused by <Fetch>.	Y	Content removed.

Copyright © 2008 Microsoft Corporation.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
2.2.1.8.2.2 Fetch	Added <LongId> as a child element.	Y	Content update.
2.2.1.8.2.3 EmptyFolderContents	49682 Changed number allowed to 0...1 (optional).	Y	Content update.
2.2.1.8.2.7 Store	49608 Changed the data type and number allowed values.	Y	Content update.
2.2.1.8.2.8 MIMESupport	49632 Clarified when the <Fetch> request MUST include the <MIMESupport> element.	Y	Content update.
2.2.1.8.2.8 MIMESupport	49633 Changed data type to unsigned byte.	Y	Content update.
2.2.1.8.2.10 LongId	Added new section.	Y	New content added.
2.2.1.8.2.13 Schema	49581 Updated child elements value to point to content class protocol documents.	Y	Content update.
2.2.1.8.2.14 Range	49643 Changed data type description.	Y	Content update.
2.2.1.8.2.17 Move	52986 Added product behavior note.	N	Content update.
2.2.1.8.2.18 ConversationId	52986 Added product behavior note.	Y	Content update.
2.2.1.8.2.19 DstFldId	52986 Added product behavior note.	Y	Content update.
2.2.1.8.2.20 MoveAlways	49639 Changed the value of the data type and number allowed.	Y	Content update.
2.2.1.8.2.20 MoveAlways	52986 Added product behavior note.	Y	New product behavior note added.
2.2.1.8.3 Response	49674 Added <Move> element to the XSD.	Y	Content update.
2.2.1.8.3 Response	49677 Changed namespace of <LinkId> element.	Y	Content update.
2.2.1.8.3 Response	49641 Added <CollectionId>, <ServerId> and <Class>	Y	New content

Copyright © 2008 Microsoft Corporation.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
	to response XSD.		added.
2.2.1.8.3 Response	49682 Changed <EmptyFolderContents> maxOccurs to unbounded, and changed <CollectionId> minOccurs to 1.	Y	Content update.
2.2.1.8.3.2 Response	49682 Changed number allowed to 0...1 (optional).	Y	Content update.
2.2.1.8.3.5 ServerId	52811 Added new section.	Y	New content added.
2.2.1.8.3.6 Fetch	49586 Removed information about errors caused by <Fetch>.	Y	Content removed.
2.2.1.8.3.6 Fetch	Added <LongID> as a child element.	Y	Content update.
2.2.1.8.3.8 LongId	Added new section.	Y	New content added.
2.2.1.8.3.9 Class	49129 Removed <Schema> from parent element column in table.	Y	Content removed.
2.2.1.8.3.9 Class	49317 Removed <Content> from parent element column in table.	Y	Content removed.
2.2.1.8.3.9 Class	49134 Changed <Fetch> (request only) to <Fetch> (response only).	Y	Content update.
2.2.1.8.3.10 Properties	49682 Changed number allowed to 0...1 (optional).	Y	Content update.
2.2.1.8.3.11 Range	49643 Changed data type description.	Y	Content update.
2.2.1.8.3.12 Status	48002 Removed <Status> value 13.	Y	Content removed.
2.2.1.8.3.12 Status	49698 Removed sentence about <Status> value not being returned.	Y	Content removed.
2.2.1.8.3.12 Status	52802 Added <Status> values 155 and 156.	Y	New product behavior note added.

Copyright © 2008 Microsoft Corporation.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
2.2.1.9.1.1 MeetingResponse	49647 Added new section.	Y	New content added.
2.2.1.9.1.2 CollectionId	49647 Added link to <LongId> section.	N	Content update.
2.2.1.9.1.2 CollectionId	49651 Rephrased sentence about the Inbox folder.	N	Content update.
2.2.1.9.1.3 Request	49647 Added link to <LongId> section.	N	Content update.
2.2.1.9.1.4 RequestId	49647 Added link to <LongId> section.	N	Content update.
2.2.1.9.2.1 MeetingResponse	49647 Added new section.	Y	New content added.
2.2.1.9.2.5 Status	49656 Removed sentence about Sync command <Status> values.	N	Content removed.
2.2.1.10.2.1 DstMsgId	49849 Changed number allowed to 0...1 (optional).	Y	Content update.
2.2.1.10.2.5 Status	50896 Removed <Status> value 6.	Y	Content update.
2.2.1.11 Ping	52850 Added information about Sync response being applied before issuing a Ping request.	Y	Content update.
2.2.1.11 Ping	Changed information about empty Ping requests.	Y	Content update.
2.2.1.11.1.1 Class	52986 Added product behavior note.	Y	New content added.
2.2.1.12 Provision	49960 Removed behavior information.	Y	Content update.
2.2.1.13 ResolveRecipients	52824 Added information about new free/busy retrieval functionality.	Y	Content update.
2.2.1.13.1 Request	49868 Removed duplicate element end tag from XSD.	Y	Content removed.
2.2.1.13.1 Request	52824 Updated XSD to include new <Availability>, <StartTime>, and <EndTime> elements.	Y	Content update.

Copyright © 2008 Microsoft Corporation.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
2.2.1.13.1.1 CertificateRetrieval	49859 Changed the descriptions for CertificateRetrieval values 2 and 3.	Y	Content update.
2.2.1.13.1.1 CertificateRetrieval	49849 Changed number allowed to 0...1 (optional).	Y	Content update.
2.2.1.13.1.3 Options	52824 Added <Availability> as a child element.	Y	New content added.
2.2.1.13.1.6 To	52824 Added information about new free/busy functionality.	Y	Content update.
2.2.1.13.1.7 Availability	52824 Added new section.	Y	New content added.
2.2.1.13.1.8 StartTime	52824 Added new section.	Y	New content added.
2.2.1.13.1.9 EndTime	52824 Added new section.	Y	New content added.
2.2.1.13.2 Response	50542 Changed the order of </xs:complextype> and </xs:sequence> in the XSD.	N	Content update.
2.2.1.13.2 Response	49921 Added <Certificate> element to the XSD.	N	New content added.
2.2.1.13.2 Response	49948 Added a maxlength value to the <To> element.	Y	New content added.
2.2.1.13.2 Response	49849 Changed the maxOccurs value of the <Response> element to 1.	N	Content update.
2.2.1.13.2.2 CertificateCount	49942 Clarified the number allowed value.	Y	Content update.
2.2.1.13.2.4 DisplayName	49942 Clarified the number allowed.	Y	Content update.
2.2.1.13.2.5 EmailAddress	49942 Clarified the number allowed value.	Y	Content update.
2.2.1.13.2.6 MiniCertificate	49942 Clarified the number allowed.	Y	Content update.

Copyright © 2008 Microsoft Corporation.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
2.2.1.13.2.10 Response	49849 Changed number allowed to 0...1 (optional).	N	Content update.
2.2.1.13.2.11 Status	47942 Removed <Status> value 9.	Y	Content removed.
2.2.1.13.2.11 Status	48739 Removed sentence about assuming success if <Status> is not returned.	Y	Content removed.
2.2.1.13.2.11 Status	52824 Added <Availability> as a parent element and added new <Status> values returned under the <Availability> element.	Y	New content added.
2.2.1.13.2.11 Status	49915 Updated the number allowed value.	Y	Content update.
2.2.1.13.2.12 To	52824 Added information about ambiguous users.	Y	Content update.
2.2.1.13.2.13 Type	49959 Changed data type to unsigned byte.	Y	Content update.
2.2.1.13.2.14 Availability	52824 Added new section.	Y	New content added.
2.2.1.13.2.15 MergedFreeBusy	52824 Added new section.	Y	New content added.
2.2.1.14 Search	49993 Updated the <Range> description.	Y	Content update.
2.2.1.14 Search	50100 Updated list of ANR indexed properties.	Y	Content update.
2.2.1.14 Search	Added document library to Search command description.	Y	Content update.
2.2.1.14.1.2 Query	50006 Changed parent element to <Store>.	Y	Content update.
2.2.1.14.1.2 Query	50008 Changed data type to Container.	Y	Content update.
2.2.1.14.1.2 Query	49865 Changed SearchTooComplex value to 8.	Y	Content update.
2.2.1.14.1.3 Options	50010 Changed parent element to <Store>.	Y	Content update.
2.2.1.14.1.3 Options	49481 Removed <Username> and <Password> from	Y	Content removed.

Copyright © 2008 Microsoft Corporation.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
	Mailbox store.		
2.2.1.14.1.4 Range	Added information about a default maximum of 100 items being returned.	Y	Content update.
2.2.1.14.1.5 Schema	49864 Added new section.	Y	New content added.
2.2.1.14.1.9 And	48208 Removed <And> and <Or> from parent elements. Changed number allowed to 0...N.	Y	Content update.
2.2.1.14.1.9 And	49865 Changed SearchTooComplex value to 8.	Y	Content update.
2.2.1.14.1.9 And	48210 Removed <EqualTo> as a child element.	Y	Content removed.
2.2.1.14.1.10 Or	49865 Changed the SearchTooComplex value to 8.	Y	Content update.
2.2.1.14.1.10 Or	48210 Removed <EqualTo> as a child element.	Y	Content removed.
2.2.1.14.1.11 Class	49865 Changed the SearchTooComplex value to 8.	Y	Content update.
2.2.1.14.1.11 Class	Modified the product behavior note to use the MS-ASProtocolVersion header values and include the SMS class.	Y	Product behavior note updated.
2.2.1.14.1.13 EqualTo	49865 Changed the SearchTooComplex value to 8.	Y	Content update.
2.2.1.14.1.13 EqualTo	48210 Modified information about parent elements.	Y	Content removed.
2.2.1.14.1.14 GreaterThan	49865 Changed the SearchTooComplex value to 8.	Y	Content update.
2.2.1.14.1.15 LessThan	49865 Changed the SearchTooComplex value to 8.	Y	Content update.
2.2.1.14.1.17 FreeText	49865 Changed the SearchTooComplex value to 8.	Y	Content update.
2.2.1.14.1.18 CollectionId	49865 Changed the SearchTooComplex value to 8.	Y	Content update.
2.2.1.14.1.19 ConversationId	49865 Changed the SearchTooComplex value to 8.	Y	Content update.
2.2.1.14.1.19	52986	Y	New

Copyright © 2008 Microsoft Corporation.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
ConversationId	Added product behavior note.		product behavior note added.
2.2.1.14.2.2 Properties	49787 Updated the child elements.	Y	Content update.
2.2.1.14.2.2 Properties	50014 Removed information about the Properties element in the Search request.	Y	Content removed.
2.2.1.14.2.7 Status	48140 Removed <Status> value 9.	Y	Content removed.
2.2.1.15 SendMail	50242 Changed "can" to "SHOULD".	Y	Content update.
2.2.1.15.1.4 Mime	53397 Added information about transferring the content as an opaque BLOB and changed the data type to byte array.	Y	Content update.
2.2.1.15.2.2 Status	Removed sentence about the <Status> element not being returned.	Y	Content removed.
2.2.1.16 Settings	Modified product behavior note to include Exchange 2010 behavior.	Y	Product behavior note updated.
2.2.1.16.1 Request	50259 Removed <TelephonyInformation> from the XSD.	Y	Content update.
2.2.1.16.1 Request	50168 Added minOccurs and maxOccurs for <Oof> element.	Y	Content update.
2.2.1.16.1 Request	50224 Removed <AlternateMailboxInformationVersion> element from XSD.	Y	Content removed.
2.2.1.16.1.1 Settings	50139 Updated child elements to include <Oof>, <DeviceInformation>, <DevicePassword>, and <UserInformation>.	Y	Content update.
2.2.1.16.1.1 Settings	50143 Removed duplicate information about the Settings command.	Y	Content removed.
2.2.1.16.1.2 Oof	50149 Reworded statement about <Get> and <Set> elements.	Y	Content update.

Copyright © 2008 Microsoft Corporation.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
2.2.1.16.1.3 Get	50168 Added number allowed for <UserInformation>.	Y	Content update.
2.2.1.16.1.3 Get	50250 Changed information about the data type and parent elements of <Get> in requests and responses.	Y	Content update.
2.2.1.16.1.4 Set	47819 Changed child elements in a <DeviceInformation> response to none.	Y	Content update.
2.2.1.16.1.4 Set	50224 Added <UserAgent> as a child element of a <DeviceInformation> request.	Y	Content update.
2.2.1.16.1.4 Set	50184 Changed required to optional.	Y	Content update.
2.2.1.16.1.4.2 Device Information Property	50236 Changed "Active Sync" to "the server".	N	Content update.
2.2.1.16.1.6 StartTime	47820 Removed requirement about <Oof> being set to 2.	Y	Content update.
2.2.1.16.1.6 StartTime	50237 Changed number allowed.	Y	Content update.
2.2.1.16.1.7 EndTime	50237 Changed number allowed.	Y	Content update.
2.2.1.16.1.8 OofMessage	51963 Updated details about <AppliesToExternalKnown> and <AppliesToExternalUnknown> being returned in responses.	Y	Content update.
2.2.1.16.1.8 OofMessage	Updated product behavior note to include Exchange 2010.	Y	Content update.
2.2.1.16.1.10 AppliesToInternal	50246 Changed data type to Empty, and described the Empty tag.	Y	Content update.
2.2.1.16.1.10 AppliesToInternal	50241 Clarified the number allowed value.	Y	Content update.
2.2.1.16.1.11 AppliesToExternalKnown	50246 Changed data type to Empty, and described the Empty tag.	Y	Content update.
2.2.1.16.1.11 AppliesToExternalKnown	50241 Clarified number allowed value.	Y	Content update.
2.2.1.16.1.12	50246	Y	Content

Copyright © 2008 Microsoft Corporation.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
AppliesToExternalUnknown	Changed data type to Empty, and described the Empty tag.		update.
2.2.1.16.1.12 AppliesToExternalUnknown	50241 Clarified number allowed value.	Y	Content update.
2.2.1.16.1.13 Enabled	50247 Changed data type to String.	Y	Content update.
2.2.1.16.1.15 DeviceInformation	50261 Changed required to optional.	Y	Content update.
2.2.1.16.1.26 EnableOutboundSMS	52986 Added product behavior note.	Y	New product behavior note added.
2.2.1.16.1.27 MobileOperator	52986 Added product behavior note.	Y	New product behavior note added.
2.2.1.16.2.1 Settings	50139 Updated child elements to include <Oof>, <DeviceInformation>, <DevicePassword>, and <UserInformation>.	Y	Content update.
2.2.1.16.2.1 Settings	50143 Removed duplicate information about the Settings command.	N	Content removed.
2.2.1.16.2.2 Status	51963 Updated information about status values for <OOF> node.	Y	Content update.
2.2.1.16.2.2 Status	48752 Removed global <Status> values 3, 4, 5, & 6.	Y	Content removed.
2.2.1.16.2.2 Status	48822 Removed sentence about <Status> value not being returned.	Y	Content removed.
2.2.1.16.2.2 Status	48570 Removed <Status> value 3 from <DevicePassword> response.	Y	Content removed.
2.2.1.16.2.2 Status	48573 Removed <Status> values 3, 4, 5, & 6 from <DeviceInformation> response.	Y	Content removed.
2.2.1.16.2.2 Status	50248 Added <UserInformation> to parent elements.	Y	Content update.

Copyright © 2008 Microsoft Corporation.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
2.2.1.16.2.3 Oof	50149 Reworded statement about <Get> and <Set> elements.	Y	Content update.
2.2.1.16.2.4 Get	50250 Changed information about the data type and parent elements of <Get> in requests and responses.	Y	Content update.
2.2.1.16.2.5 Set	47819 Changed child elements in a <DeviceInformation> response to none.	Y	Content update.
2.2.1.16.2.5 Set	50253 Added <EnableOutboundSMS> and <MobileOperator> as child elements in a <DeviceInformation> request.	Y	Content update.
2.2.1.16.2.5 Set	50224 Added <UserAgent> as a child element of a <DeviceInformation> request.	Y	Content update.
2.2.1.16.2.5 Set	50184 Changed required to optional.	Y	Content update.
2.2.1.16.2.5.2 Device Information Property	50236 Changed "Active Sync" to "the server".	N	Content update.
2.2.1.16.2.7 StartTime	47820 Removed requirement about <Oof> being set to 2.	Y	Content update.
2.2.1.16.2.9 OofMessage	51963 Updated details about <AppliesToExternalKnown> and <AppliesToExternalUnknown> being returned in responses.	Y	Content update.
2.2.1.16.2.9 OofMessage	50264 Added product behavior note.	Y	Content update.
2.2.1.16.2.11 AppliesToInternal	50241 Clarified number allowed value.	Y	Content update.
2.2.1.16.2.11 AppliesToInternal	50246 Changed data type to Empty and described the Empty element.	Y	Content update.
2.2.1.16.2.12 AppliesToExternalKnown	50241 Clarified number allowed value.	Y	Content update.
2.2.1.16.2.12 AppliesToExternalKnown	50246 Changed data type to Empty and explained the Empty data type.	Y	Content update.
2.2.1.16.2.13	50241	Y	Content

Copyright © 2008 Microsoft Corporation.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
AppliesToExternalUnknown	Clarified number allowed value.		update.
2.2.1.16.2.13 AppliesToExternalUnknown	50246 Changed data type to Empty and described the Empty element.	Y	Content update.
2.2.1.16.2.15 ReplyMessage	50249 Removed duplicate content about <OOF> element.	N	Content removed.
2.2.1.16.2.16 DeviceInformation	50261 Changed required to optional.	Y	Content update.
2.2.1.17.1.9 InstanceId	50371 Changed data type to String.	Y	Content update.
2.2.1.17.1.10 Mime	50377 Changed number allowed to 1...1 (required).	Y	Content update.
2.2.1.17.1.10 Mime	53397 Added information about transferring the content as an opaque BLOB and changed the data type to byte array.	Y	Content update.
2.2.1.18.1.5 Source	50377 Changed the number allowed to 1...1 (required).	Y	Content update.
2.2.1.18.1.9 InstanceId	50371 Changed data type to String.	Y	Content update.
2.2.1.18.1.10 Mime	50377 Changed number allowed to 1...1 (required).	Y	Content update.
2.2.1.18.1.10 Mime	53397 Added information about transferring the content as an opaque BLOB and changed the data type to byte array.	Y	Content update.
2.2.1.19.1 Request	51961 Added minOccurs="0" maxOccurs="2" to the <Options> element.	Y	Content update.
2.2.1.19.1.1 Sync	50267 Changed number allowed to 1...1 (required).	Y	Content update.
2.2.1.19.1.1 Sync	50673 Added elements to the child elements column.	Y	Content update.
2.2.1.19.1.3 HeartbeatInterval	50268 Changed minimum value to 1.	Y	Content update.
2.2.1.19.1.4 Partial	49640 Added information about the Empty data type.	Y	Content update.
2.2.1.19.1.5	50271	Y	Content

Copyright © 2008 Microsoft Corporation.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
WindowSize	Reworded sentence about changing the window size during a synchronization.		update.
2.2.1.19.1.5 WindowSize	50639 Added information about setting the <WindowSize> element to a value greater than 512 or to 0 (zero).	Y	Content update.
2.2.1.19.1.6 Add	47905 Changed reference from "Add command" to "Add element".	N	Content update.
2.2.1.19.1.10 Collection	50275 Removed <Class> element as a child of <Collection> element.	Y	Content update.
2.2.1.19.1.10 Collection	50274 Changed number allowed to 1...N.	Y	Content update.
2.2.1.19.1.12 Supported	47887 Added product behavior note.	Y	Content update.
2.2.1.19.1.13 GetChanges	47977 Changed information about setting <GetChanges> to 0 or 1.	Y	Content update.
2.2.1.19.1.13 GetChanges	47977 Removed paragraph about calendar event conflicts.	Y	Content removed.
2.2.1.19.1.14 ConversationMode	52986 Updated product behavior note to specify MS-ASProtocolVersion header value.	Y	Product behavior note updated.
2.2.1.19.1.17 Commands	52803 Added <SoftDelete> as a child element.	Y	New content added.
2.2.1.19.1.18 Delete	47980 Changed SoftDelete section to Delete section.	Y	Content update.
2.2.1.19.1.21 Options	50333 Removed <ConversationMode> from child elements column.	Y	Content update.
2.2.1.19.1.21 Options	51961 Changed the maximum number allowed to 2.	Y	Content update.
2.2.1.19.1.22 Conflict	50376 Changed data type to unsigned byte.	Y	Content update.
2.2.1.19.1.23 FilterType	48074 Clarified the use of <FilterType> on calendar items.	Y	Content update.

Copyright © 2008 Microsoft Corporation.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
2.2.1.19.1.23 FilterType	48076 Changed information about <FilterType> in contact Sync requests.	Y	Content update.
2.2.1.19.1.23 FilterType	50376 Changed data type to Unsigned byte.	Y	Content update.
2.2.1.19.1.23 FilterType	48073 Added sentence stating that clients MUST send a maximum of 1 <FilterType> element.	Y	Content update.
2.2.1.19.1.24 MIMETruncation	49213 Added information about <MIMETruncation> being an approximate size.	Y	Content update.
2.2.1.19.1.24 MIMETruncation	49211 Removed sentence about <MIMETruncation> element in Fetch operation.	Y	Content removed.
2.2.1.19.1.24 MIMETruncation	50376 Changed data type to unsigned byte.	Y	Content update.
2.2.1.19.1.25 MIMESupport	50376 Changed data type to unsigned byte.	Y	Content update.
2.2.1.19.1.25 MIMESupport	49212 Added sentence stating that the client MUST send a maximum of one <MIMESupport> element.	Y	Content update.
2.2.1.19.1.26 Class	52986 Changed product behavior note to specify MS-ASProtocolVersion header value.	N	Content update.
2.2.1.19.1.27 MaxItems	50350 Changed an instance of "can" to "MUST".	Y	Content update.
2.2.1.19.1.27 MaxItems	52986 Changed the product behavior note to specify the MS-ASProtocolVersion header value.	N	Content update.
2.2.1.19.2 Response	52803 Added <SoftDelete> element to XSD.	Y	New content added.
2.2.1.19.2 Response	50675 Moved the <Limit> element, removed a <xs:sequence> tag, and added a <xs:choice> tag.	Y	Content update.
2.2.1.19.2 Response	50553 Added maxOccurs="unbounded" to the <Change> element.	Y	Content update.
2.2.1.19.2 Response	50547 Added minOccurs and maxOccurs values to	Y	Content update.

Copyright © 2008 Microsoft Corporation.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
	elements that did not have them specified.		
2.2.1.19.2 Response	50468 Removed <MimeData>, <MimeSize>, and <MimeTruncated> from the XSD.	Y	Content removed.
2.2.1.19.2.1 Add	50386 Changed SHOULD to MUST.	Y	Content update.
2.2.1.19.2.5 Collection	50448 Removed sentence about Sync requests only containing one <Collection> element.	Y	Content removed.
2.2.1.19.2.8 Commands	50450 Added <SoftDelete> as a child element.	Y	New content added.
2.2.1.19.2.9 Delete	50467 Added new section.	Y	New content added.
2.2.1.19.2.10 SoftDelete	52803 Updated description of element.	Y	Content update.
2.2.1.19.2.13 MoreAvailable	49640 Added information about the Empty data type.	Y	Content update.
2.2.1.19.2.13 MoreAvailable	50639 Added information about setting the <WindowSize> element to a value greater than 512 or to 0 (zero).	Y	Content update.
2.2.1.19.2.14 Responses	50480 Added <Fetch> and <Delete> to example.	Y	Content update.
2.2.1.19.2.15 ServerId	50483 Removed "(response only)" from <Fetch> in the parent elements column.	Y	Content update.
2.2.1.19.2.16 Status	50458 Removed statement about assuming success, as this element is required in non-empty responses.	Y	Content removed.
2.2.1.19.2.16 Status	50459 Updated the descriptions for <Status> values 13 and 16.	N	Content update.
2.2.1.19.2.17 Sync	50673 Added elements to the child elements column.	Y	Content update.
2.2.1.19.3.1.1 Sync Command Request for Calendar Items	52986 Added <DisallowNewTimeProposal>, <ResponseRequested>, <CalendarType>, <IsLeapMonth>, and <Attendees> to XSD.	Y	Content update.

Copyright © 2008 Microsoft Corporation.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
2.2.1.19.3.4.1 Sync Command Request for E-Mail	52986 Updated <DateReceived> definition and added <To>, <From>, <InternetCPID>, <Importance>, <Categories>, and <GhostingProps>.	Y	Content update.
2.2.1.19.3.5.1 Sync Command Request for Tasks	52986 Updated entire XSD.	Y	Content update.
2.2.1.20.1 Request	52986 Added ending </xs:schema> tag.	Y	Content update.
2.2.1.20.1.3 CertificateChain	48299 Changed "Number allowed" to "0...1 (optional)".	Y	Content update.
2.2.1.20.1.5 CheckCRL	49315 Added information about setting <CheckCRL> to FALSE.	Y	Content update.
2.2.1.20.2.2 Certificate	49551 Changed parent element to <ValidateCert>, and minimum number allowed to 0.	Y	Content update.
2.2.2.7 MoveItems Status Codes	50896 Removed <Status> value 6.	Y	Content update.
2.2.2.11 Search Status Codes	48140 Removed <Status> value 9.	Y	Content removed.
3.1.5 Message Processing Events and Sequencing Rules	50608 Changed link to MS-ASHTTP from section 2.2 to 2.2.1.	N	Content update.
3.1.5.1 Downloading Policy Settings	50621 Updated parameters in figure 4.	Y	Content update.
3.1.5.2 Synchronizing a Folder Hierarchy	50613 Updated information about the <WindowSize> element.	N	Content update.
3.1.5.2 Synchronizing a Folder Hierarchy	50615 Changed "initial synchronization key" to "new synchronization key".	N	Content update.
3.1.5.2 Synchronizing a Folder Hierarchy	50610 Rewrote the content of the section.	Y	Content update.
3.1.5.2 Synchronizing a Folder Hierarchy	47858 Added information about sending initial FolderSync command with a <SyncKey> value of zero (0).	N	Content update.
3.1.5.3 Synchronizing Inbox,	50612 Updated information regarding first and second	N	Content update.

Copyright © 2008 Microsoft Corporation.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
Calendar, Contacts, and Tasks Folders	Sync command.		
3.1.5.4 Receiving and Accepting Meeting Requests	50634 Removed figure.	Y	Content removed.
3.1.5.4 Receiving and Accepting Meeting Requests	50635 Added asterisk (*) to step 2.	Y	Content update.
3.1.5.4 Receiving and Accepting Meeting Requests	Added step 5 to the table.	Y	Content update.
4.2.4 Response - Case Server Settings	Updated product behavior note to include Exchange 2010.	Y	Product behavior note updated.
4.2.6 Response – Case Framework Default	52765 Updated information about the 401-1.htm file.	Y	Content update.
4.5 Fetching E-Mail and Attachments by Using the ItemOperations Command	52994 Changed figure number to 7.	Y	Content update.
4.5.3.3 Fetch Request	50676 Changed </airsync:CollectionId> to </airsync:LongId>.	Y	Content update.
4.5.4 Fetching an Attachment	50669 Added a ";" to ">."	N	Content update.
4.5.4.2 Sync Response	50670 Updated the <airsyncbase:Data> element.	N	Content update.
4.8.1 Request	Added <MobileOperator> and <UserAgent>.	Y	Content update.
4.11.1.2 Response	50671 Changed "deviceuser2"<someone1@example.com> to "deviceuser2"<someone2@example.com>.	N	Content update.
4.11.2 Forward a Search Result	Changed example code.	Y	Content update.
4.12 Resolving Recipients and Retrieving Free/Busy Data by Using the ResolveRecipients	52824 Updated topic title to reflect new functionality.	Y	Content update.

Copyright © 2008 Microsoft Corporation.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
Command			
4.12.3 Retrieving Free/Busy Data By Using the ResolveRecipients Command	52824 Added new topic.	Y	New content added.
4.12.3.1 Request to Retrieve Free/Busy Data	52824 Added new example to demonstrate the new free/busy functionality.	Y	New content added.
4.12.3.2 Response with MergedFreeBusy Data	52824 Added new example to demonstrate the new free/busy functionality.	Y	New content added.
4.13.1.1 Request	Updated MS-ASProtocolVersion value to 14.0.	N	Content update.
4.13.1.2 Response	53339 Added behavior note.	Y	New product behavior note added.
4.13.2.1 Request	Changed the MS-ASProtocolVersion value.	Y	Content update.
4.13.3.1 Request	Changed the MS-ASProtocolVersion value.	Y	Content update.
4.13.4.1 Request	Changed the MS-ASProtocolVersion value.	Y	Content update.
4.13.5.1 Request	Changed the MS-ASProtocolVersion value.	Y	Content update.
2.2.1.7.2.5 Class	49646 Removed the Class topic.	N	Content removed.
3.1.3.1 Initial Synchronization	50606 Removed the Initial Synchronization topic.	N	Content removed.
3.1.5.2 Synchronizing Inbox, Calendar, Contacts, and Tasks Folders	50610 Switched the order of sections 3.1.5.2 and 3.1.5.3.	N	Content update.
2.2.1.19.2.23 Wait and 2.2.19.2.24 HeartbeatInterval	50466 Removed the Wait and HeartbeatInterval topics from the Sync command response section.	N	Content removed.
	50468	N	Content

Copyright © 2008 Microsoft Corporation.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
2.2.1.19.2.14 MimeData	Removed the MimeData topic.		removed.
2.2.119.2.15 MimeSize	50468 Removed the MimeSize topic.	N	Content removed.
2.2.1.19.2.16 MimeTruncated	50468 Removed the MimeTruncated topic.	N	Content removed.
2.2.1.19.2.13 MessageClass	50468 Removed the MessageClass topic.	N	Content removed.
2.2.1.14.1.5 Properties	53336 Removed the Properties topic.	N	Content removed.

Copyright © 2008 Microsoft Corporation.

8 Index

Copyright © 2008 Microsoft Corporation.