[MS-ASHTTP]:

Exchange ActiveSync: HTTP Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

Revision Summary

Date	Revision History	Revision Class	Comments
12/3/2008	1.0.0	Major	Initial release.
2/4/2009	1.0.1	Editorial	Revised and edited technical content.
3/4/2009	1.0.2	Editorial	Revised and edited technical content.
4/10/2009	2.0.0	Major	Updated technical content and applicable product releases.
7/15/2009	3.0.0	Major	Revised and edited for technical content.
11/4/2009	4.0.0	Major	Updated and revised the technical content.
2/10/2010	5.0.0	Major	Updated and revised the technical content.
5/5/2010	6.0.0	Major	Updated and revised the technical content.
8/4/2010	7.0	Major	Significantly changed the technical content.
11/3/2010	7.1	Minor	Clarified the meaning of the technical content.
3/18/2011	8.0	Major	Significantly changed the technical content.
8/5/2011	8.1	Minor	Clarified the meaning of the technical content.
10/7/2011	8.2	Minor	Clarified the meaning of the technical content.
1/20/2012	9.0	Major	Significantly changed the technical content.
4/27/2012	9.1	Minor	Clarified the meaning of the technical content.
7/16/2012	10.0	Major	Significantly changed the technical content.
10/8/2012	11.0	Major	Significantly changed the technical content.
2/11/2013	11.0	None	No changes to the meaning, language, or formatting of the technical content.
7/26/2013	12.0	Major	Significantly changed the technical content.
11/18/2013	12.0	None	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	12.0	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	13.0	Major	Significantly changed the technical content.
7/31/2014	13.0	None	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	13.1	Minor	Clarified the meaning of the technical content.
5/26/2015	14.0	Major	Significantly changed the technical content.
6/30/2015	15.0	Major	Significantly changed the technical content.
9/14/2015	16.0	Major	Significantly changed the technical content.
6/9/2016	17.0	Major	Significantly changed the technical content.

Date	Revision History	Revision Class	Comments
2/28/2017	18.0	Major	Significantly changed the technical content.
9/19/2017	19.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction

The Exchange ActiveSync: HTTP Protocol enables client devices to synchronize data with the data that is stored on the server.

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

alias: An alternate name that can be used to reference an object or element.

Augmented Backus-Naur Form (ABNF): A modified version of Backus-Naur Form (BNF), commonly used by Internet specifications. ABNF notation balances compactness and simplicity with reasonable representational power. ABNF differs from standard BNF in its definitions and uses of naming rules, repetition, alternatives, order-independence, and value ranges. For more information, see [RFC5234].

base64 encoding: A binary-to-text encoding scheme whereby an arbitrary sequence of bytes is converted to a sequence of printable ASCII characters, as described in [RFC4648].

calendar: A date range that shows availability, **meetings**, and appointments for one or more users or resources. See also Calendar object.

contact: (1) A presence entity (presentity) whose presence information can be tracked.

(2) An object of the contact class that represents a company or person whom a user can contact.

encrypted message: An Internet email message that is in the format described by [RFC5751] and uses the EnvelopedData CMS content type described in [RFC3852], or the **Message object** that represents such a message.

Global Address List (GAL): An address list that conceptually represents the default address list for an address book.

globally unique identifier (GUID): A term used interchangeably with universally unique identifier (UUID) in Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the value. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] must be used for generating the **GUID**. See also universally unique identifier (UUID).

Hypertext Transfer Protocol (HTTP): An application-level protocol for distributed, collaborative, hypermedia information systems (text, graphic images, sound, video, and other multimedia files) on the World Wide Web.

Hypertext Transfer Protocol Secure (HTTPS): An extension of HTTP that securely encrypts and decrypts web page requests. In some older protocols, "Hypertext Transfer Protocol over Secure Sockets Layer" is still used (Secure Sockets Layer has been deprecated). For more information, see [SSL3] and [RFC5246].

Inbox folder: A special folder that is the default location for **Message objects** received by a user or resource.

locale: A collection of rules and data that are specific to a language and a geographical area. A locale can include information about sorting rules, date and time formatting, numeric and monetary conventions, and character classification.

mailbox: A message store that contains email, calendar items, and other **Message objects** for a single recipient.

meeting: An event with attendees.

meeting request: An instance of a Meeting Request object.

Message object: A set of properties that represents an email message, appointment, contact, or other type of personal-information-management object. In addition to its own properties, a Message object contains recipient properties that represent the addressees to which it is addressed, and an attachments table that represents any files and other Message objects that are attached to it.

MIME message: A message that is as described in [RFC2045], [RFC2046], and [RFC2047].

OAuth: The OAuth 2.0 authorization framework [RFC6749].

Out of Office (OOF): One of the possible values for the free/busy status on an appointment. It indicates that the user will not be in the office during the appointment.

plain text: Text that does not have markup. See also plain text message body.

recipient: An entity that can receive email messages.

S/MIME (Secure/Multipurpose Internet Mail Extensions): A set of cryptographic security services, as described in [RFC5751].

Secure Sockets Layer (SSL): A security protocol that supports confidentiality and integrity of messages in client and server applications that communicate over open networks. SSL uses two keys to encrypt data-a public key known to everyone and a private or secret key known only to the recipient of the message. SSL supports server and, optionally, client authentication using X.509 certificates. For more information, see [X509]. The SSL protocol is precursor to **Transport Layer Security (TLS)**. The TLS version 1.0 specification is based on SSL version 3.0 [SSL3].

Sent Items folder: A special folder that is the default location for storing copies of **Message objects** after they are submitted or sent.

server ID: A unique identifier that is assigned by the server to each object that can be synchronized. A client stores the server ID for each object and is able to locate an object when given a server ID.

SSL/TLS handshake: The process of negotiating and establishing a connection protected by **Secure Sockets Layer (SSL)** or **Transport Layer Security (TLS)**. For more information, see [SSL3] and [RFC2246].

Transport Layer Security (TLS): A security protocol that supports confidentiality and integrity of messages in client and server applications communicating over open networks. **TLS** supports server and, optionally, client authentication by using X.509 certificates (as specified in [X509]). **TLS** is standardized in the IETF TLS working group.

Uniform Resource Identifier (URI): A string that identifies a resource. The URI is an addressing mechanism defined in Internet Engineering Task Force (IETF) Uniform Resource Identifier (URI): Generic Syntax [RFC3986].

Uniform Resource Locator (URL): A string of characters in a standardized format that identifies a document or resource on the World Wide Web. The format is as specified in [RFC1738].

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

XML: The Extensible Markup Language, as described in [XML1.0].

XML schema definition (XSD): The World Wide Web Consortium (W3C) standard language that is used in defining XML schemas. Schemas are useful for enforcing structure and constraining the types of data that can be used validly within other XML documents. XML schema definition refers to the fully specified and currently recommended standard for use in authoring XML schemas.

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

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

1.2.1 Normative References

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

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

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

[MS-LCID] Microsoft Corporation, "Windows Language Code Identifier (LCID) Reference".

[RFC1945] Berners-Lee, T., Fielding, R., and Frystyk, H., "Hypertext Transfer Protocol -- HTTP/1.0", RFC 1945, May 1996, http://www.ietf.org/rfc/rfc1945.txt

[RFC2045] Freed, N., and Borenstein, N., "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996, http://www.rfc-editor.org/rfc/rfc2045.txt

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

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

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000, http://www.rfc-editor.org/rfc/rfc2818.txt

[RFC2822] Resnick, P., Ed., "Internet Message Format", RFC 2822, April 2001, http://www.ietf.org/rfc/rfc2822.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

[RFC4985] Santesson, S., "Internet X.509 Public Key Infrastructure Subject Alternative Name for Expression of Service Name", RFC 4985, August 2007, http://www.rfc-editor.org/rfc/rfc4985.txt

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

[RFC5246] Dierks, T., and Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008, http://www.ietf.org/rfc/rfc5246.txt

[RFC6265] Barth, A., "HTTP State Management Mechanism", RFC 6265, April 2011, https://tools.ietf.org/html/rfc6265

[RFC6749] Hardt, D., Ed., "The OAuth 2.0 Authorization Framework", RFC 6749, October 2012, http://www.rfc-editor.org/rfc/rfc6749.txt

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

1.2.2 Informative References

[MSDN-APM] Marquardt, T., "ASP.NET Performance Monitoring, and When to Alert Administrators", http://msdn.microsoft.com/en-us/library/ms972959.aspx

1.3 Overview

This protocol is used to synchronize server data with a client mobile device. The protocol relies on a client/server architecture. In this specification, the term "client" is used to refer to the software that is running on the device and communicating to the server by means of the ActiveSync protocol. The term "server" refers to the synchronization engine that communicates the synchronization protocol to the client.

All communication between the client and server is initiated by the client and is based on request/response messages. When the client communicates with the server, the client sends a request to the server as an **HTTP** **POST** method, using UTF-8 encoding. The server sends back a response to the HTTP **POST**. The request and response each have a start-line, headers, and might have a body. The format is dictated by the HTTP/1.1 standard. The HTTP **POST** request header contains certain parameters that are set by the client, as specified later in this document. The HTTP **POST** response header is created by the server, and its contents are specified later in this document. The format of the body for both request and response depends on the type of request. Generally, the request/response body contains **Wireless Application Protocol (WAP) Binary XML (WBXML)** formatted data. Each HTTP **POST** request contains a single command, such as the **Sync** command. A typical session includes several commands and, therefore, several HTTP **POST** requests.

In addition to the HTTP **POST** request/response commands, the HTTP **OPTIONS** command response provides the supported ActiveSync capabilities of the server, including supported commands and supported protocol versions.

1.4 Relationship to Other Protocols

This protocol uses either an **HTTP** connection or an **HTTPS** connection between the client and server. A TCP/IP network transports messages between a client and server by using either the HTTP protocol or the HTTPS protocol, by means of a series of request and response calls. The protocol specified in [MS-ASCMD] uses this protocol as a transport.

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

1.5 Prerequisites/Preconditions

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

1.6   Applicability Statement

This protocol specifies the transport mechanism for the commands defined in [MS-ASCMD] and all data structures associated with those commands. It is applicable to any client or server that synchronizes **calendar**, **contact (2)**, e-mail, task, note, and other data between a mail server and a mobile device.

1.7   Versioning and Capability Negotiation

The **HTTP** **OPTIONS** command (section 2.2.3) is used by the client to discover which versions of the ActiveSync protocol are supported by the server. To determine the supported versions, the client examines the MS-ASProtocolVersions header (section 2.2.4.1.2.2), which is returned in the HTTP **OPTIONS** command response.

The client uses the MS-ASProtocolVersion header (section 2.2.1.1.2.6) of the HTTP **POST** command (section 2.2.1) to indicate to the server which ActiveSync protocol version it is using.

The latest version of the ActiveSync protocol that the client or server can support is 16.1. Older versions include 16.0, 14.1, 14.0, 12.1, 12.0, and 2.5. Some commands and functionality described in the ActiveSync protocol documentation are not supported by all of the protocol versions. See the command and element descriptions in the ActiveSync protocol documents to determine which commands, elements, and capabilities are supported by the protocol versions.

1.8   Vendor-Extensible Fields

None

1.9   Standards Assignments

None

2 Messages

2.1 Transport

Messages are transported by using **HTTP** **POST** and HTTP **OPTIONS**, as specified in [RFC2616]. These commands are sent via HTTP or **Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)**. The query parameters in the request **URI** can be encoded with **base64 encoding** (see section 2.2.1.1.1.1) or in plain text (see section 2.2.1.1.1.2). The body of the HTTP message contains the **WBXML** that is required by the command being communicated in the message. The commands are specified in [MS-ASCMD].

2.2 Message Syntax

The XML markup that constitutes the request body (section 2.2.1.1.3) or the response body (section 2.2.2.1.3) is transmitted between client and server by using **Wireless Application Protocol (WAP) Binary XML (WBXML)**, as specified in [WBXML1.2].

The following are the two general types of messages:

- **HTTP** **POST**

- HTTP **OPTIONS**

2.2.1 HTTP POST Request

The client creates a request by using the **HTTP** **POST** command to initiate communications between the client and the server.

2.2.1.1 Request Format

Each command is sent from the client to the server as an **HTTP** **POST** containing command data. As specified by [RFC2616], the format is as follows.

```
Request-line
Request-headers
CR/LF
Request Body
```

2.2.1.1.1 Request Line

The request line consists of the method indicator, **POST**, followed by the **URI**, followed by the **HTTP** version, as follows.

```
POST <URI> HTTP/1.1
```

The URI can be either an absolute URI or a relative URI, as specified in [RFC2616] section 3.2.1. The absolute URI consists of a scheme indicator, the host name, and the path, followed by a query value. The relative URI consists of the path and the query value.

The path and query value in the URI have the following format.

```
/<ActiveSync virtual directory name>?<query value>
```

The query value in the URI contains all of the URI parameters and can contain some of the request headers. The format can be either **plain text**, as specified in section 2.2.1.1.1.2, or **base64 encoding**, as specified in section 2.2.1.1.1.1. Either format can be used with protocol versions 12.1, 14.0, 14.1, 16.0, and 16.1. The base64 encoding format is not supported by protocol versions 2.5 and 12.0; the plain text format is supported by all protocol versions.

The following two examples are equivalent. The first example uses the plain text query value, and the second example uses the base64-encoded query value.

```
POST /Microsoft-Server-
ActiveSync?Cmd=Sync&User=rmjones&DeviceId=v140Device&DeviceType=SmartPhone HTTP/1.1
Content-Type: application/vnd.ms-sync.wbxml
MS-ASProtocolVersion: 14.0
User-Agent: ASOM
Host: Contoso.com
Accept-Language: en-us
Content-Length: 868

POST /Microsoft-Server-ActiveSync?jAAJBAp2MTQwRGV2aWNlAApTbWFydFBob25l HTTP/1.1
Content-Type: application/vnd.ms-sync.wbxml
User-Agent: ASOM
Host: Contoso.com
Content-Length: 866
```

2.2.1.1.1.1   Base64-Encoded Query Value

The base64-encoded query value uses **base64 encoding** to specify the **URI** parameters and request headers. The URI parameters and request headers are contained within the fields of a byte sequence. Once the byte sequence is created, it is converted to base64 as specified in [RFC2045]. The base64-encoded query value is then appended to the request URI.

The following is an example of a URI that contains a base64-encoded query value.

```
/Microsoft-Server-ActiveSync?jAAJBAp2MTQwRGV2aWNlAApTbWFydFBob25l
```

NOTE: The base64-encoded query value is supported only by protocol versions 12.1, 14.0, 14.1, 16.0, and 16.1. If the client uses protocol version 2.5 or 12.0, the plain text query value, as specified in section 2.2.1.1.1.2, MUST be used in the request URI.

The fields of the byte sequence are as follows.

0	1	2	3	4	5	6	7	8	9	1 0	1	2	3	4	5	6	7	8	9	2 0	1	2	3	4	5	6	7	8	9	3 0	1
Protocol version								Command code								Locale																
Device ID length								Device ID (variable)																								
...																																
Policy key length								Policy key (optional)																								
...								Device type length								Device type (variable)																
...																																

Command parameters (variable)
...

Protocol version (1 byte): An integer that specifies the version of the ActiveSync protocol that is being used. This value SHOULD<1> be 141, 160 or 161. This value MAY<2> be 140 or 121.

Command code (1 byte): An integer that specifies the command (see table of command codes in section 2.2.1.1.1.1.2).

Locale (2 bytes): An integer that specifies the **locale** of the language that is used for the response. Locale integer values are specified in [MS-LCID].

Device ID length (1 byte): An integer that specifies the length of the **Device ID** field. This value MUST be greater than 0.

Device ID (variable): A string or a **GUID** that identifies the device. For details, see section 2.2.1.1.1.2.3. The length of this field is specified by the **Device ID length** field.

Policy key length (1 byte): An integer that specifies the length of the policy key. The only valid values are 0 or 4. A value of 0 indicates that the policy key field is absent.

Policy key (4 bytes, optional): An unsigned integer that indicates the state of policy settings on the client device, as specified in [MS-ASPROV] section 2.2.2.42. If the value of the **Policy key length** field is 0, this field is absent.

Device type length (1 byte): An integer that specifies the length of the **Device type** field.

Device type (variable): A string that specifies the type of client device. For details, see section 2.2.1.1.1.2.4. The length of this field is specified by the **Device type length** field.

Command parameters (variable): An array of **Encoded Parameter** structures as specified in section 2.2.1.1.1.1.1. This field is only present if there are command-specific parameters associated with the command specified by the **Command code** field. See section 2.2.1.1.1.1.3 for a list of command-specific parameters.

2.2.1.1.1.1.1 Encoded Parameter

0	1	2	3	4	5	6	7	8	9	1 0	1	2	3	4	5	6	7	8	9	2 0	1	2	3	4	5	6	7	8	9	3 0	1
Tag								Length								Value (variable)															
...																															

Tag (1 byte): An integer that identifies the parameter. See section 2.2.1.1.1.1.3 for a list of tags and their corresponding parameters.

Length (1 byte): An integer that specifies the length of the parameter value. Valid values are from 0 to 255 characters.

Value (variable): The value of the parameter. The size of this field is specified by the **Length** field.

2.2.1.1.1.1.2 Command Codes

The following table provides the numeric codes that correspond to the ActiveSync commands. The numeric code is used in the **Command code** field of the base64 encoded **URI** to specify the command. For more details, see [MS-ASCMD].

Code	Command	Description
0	**Sync**	Synchronizes changes in a folder between the client and the server.
1	**SendMail**	Sends mail to the server. This command is issued in the **HTTP POST** command's URI, and does not contain an XML body. The body will instead contain the **MIME message**.
2	**SmartForward**	Forwards a **Message object** without retrieving the full Message object from the server.
3	**SmartReply**	Replies to a Message object without retrieving the full Message object from the server.
4	**GetAttachment**	Retrieves an e-mail attachment from the server.
9	**FolderSync**	Synchronizes the folder hierarchy but does not synchronize the items in the folders.
10	**FolderCreate**	Creates an e-mail, **calendar**, or **contacts** folder on the server.
11	**FolderDelete**	Deletes a folder from the server.
12	**FolderUpdate**	Moves a folder from one location to another on the server and is used to rename folders.
13	**MoveItems**	Moves items from one folder to another.
14	**GetItemEstimate**	Gets an estimate of the number of items in a folder that is synchronized.
15	**MeetingResponse**	Used to accept, tentatively accept, or decline a **meeting request** in the user's **Inbox folder**.
16	**Search**	Finds and retrieves information about contacts (2) and **recipients** in the **Global Address List**.
17	**Settings**	Supports getting and setting global properties, such as **Out of Office (OOF)** and device information.
18	**Ping**	Requests that the server monitor specified folders for changes that would require the client to resynchronize.
19	**ItemOperations**	Identifies the body of the request or response as containing a set of commands operating on items.
20	**Provision**	Gets the security policy settings set by the server administrator, such as the user's minimum password length requirement.
21	**ResolveRecipients**	Resolves a list of supplied recipients and optionally fetches their **S/MIME** certificates so that clients can send **encrypted messages**.
22	**ValidateCert**	Validates a certificate that has been received through an S/MIME mail.
23	**Find**	Searches for items in the mailbox using KQL syntax.

2.2.1.1.1.1.3 Command Parameters

The following table lists the tag values that correspond to the names of the command parameters. For additional details about the **AttachmentName**, **CollectionId**, **ItemId**, **LongId**, and **Occurrence** command parameters, see section 2.2.1.1.1.2.5.

Tag	Parameter Name
0	**AttachmentName**
1	**CollectionId**
3	**ItemId**
4	**LongId**
6	**Occurrence**
7	**Options**
8	**User**

The following table describes the **Options** and **User** command parameters.

Parameter	Description	Used By
Options	A single-byte bitmask that specifies command options. See the table below for valid flags for this bitmask.	**SmartReply**, **SmartForward**, **SendMail**, **ItemOperations**
User	A string that specifies the user ID in a format that can be logged in the Web server log.	Any command

The following table specifies the valid bit flags for the **Options** parameter.

Flag	Value	Meaning
SaveInSent	0x01	Set this flag to instruct the server to save the **Message object** in the user's **Sent Items folder**. Valid for **SendMail**, **SmartForward**, and **SmartReply**.
AcceptMultiPart	0x02	Set this flag to instruct the server to return the requested item in multipart format. Valid for **ItemOperations**. For more details, see section 2.2.1.1.2.5.

2.2.1.1.1.2   Plain Text Query Value

The plain text query value uses **plain text** to specify the **URI** parameters. The **Augmented Backus-Naur Form (ABNF)** notation, as specified in [RFC5234], is used to define the syntax.

```
plain-text-query       = command-spec '&' user-spec '&' device-id-spec '&' device-type-spec
*('&' parameter-spec)

command-spec           = "Cmd=" command-name
user-spec              = "User=" user-name
device-id-spec         = "DeviceId=" device-id
device-type-spec       = "DeviceType=" device-type
parameter-spec         = parameter-name "=" parameter-value

command-name           = 1*ALPHA
user-name              = 1*VCHAR
device-id              = 1*32(ALPHA / DIGIT)
device-type            = 1*VCHAR
```

```
parameter-name          = 1*ALPHA
parameter-value         = 1*VCHAR
```

2.2.1.1.1.2.1 Command

The ActiveSync command to be executed is specified by the `command-spec` **ABNF** rule portion of the plain text query value. Valid values, represented by the `command-name` ABNF rule, are specified in the "Command" column of the table in section 2.2.1.1.1.2.

2.2.1.1.1.2.2 User Name

The user ID of the user is specified by the `user-spec` **ABNF** rule portion of the plain text query value.

2.2.1.1.1.2.3 Device ID

The device ID is specified by the `device-id-spec` **ABNF** rule portion of the plain text query value. The value, represented by the `device-id` ABNF rule, is a string that specifies the device. Each device MUST have a unique device ID string. Each request from the device MUST include the same device ID string.

2.2.1.1.1.2.4 Device type

The device type is specified by the `device-type-spec` **ABNF** rule portion of the plain text query value. The value, represented by the `device-type` ABNF rule, is any string that specifies a device type. "SP" specifies a SmartPhone and "PPC" specifies a PocketPC. Other client devices send unique strings for their specific device type. Each request from a client device MUST include the same device type string.

2.2.1.1.1.2.5 Command-Specific URI Parameters

The following **URI** parameters, also called command parameters, are specific to the ActiveSync commands. They are specified by the `parameter-spec` **ABNF** rule portion of the plain text query value. Valid values for the parameter name, represented by the `parameter-name` ABNF rule, are specified by the "Parameter" column in the following table. Valid parameter values, represented by the `parameter-value` ABNF rule, are specified in the "Description" column.

Parameter	Description	Used by
AttachmentName	A string that specifies the name of the attachment file to be retrieved.	**GetAttachment**
CollectionId	A string that specifies the **server ID** of the folder that contains the **Message object** to be forwarded or replied to.	**SmartForward**, **SmartReply**
ItemId	A string that specifies the server ID of the Message object to be forwarded or replied to.	**SmartForward**, **SmartReply**
LongId	A string that references a result set that was returned in the **Search** command response.	**SmartForward**, **SmartReply**
Occurrence	A string that specifies the ID of a particular occurrence in a recurring **meeting**.	**SmartForward**, **SmartReply**
SaveInSent	A character that specifies whether a copy of the Message object will be saved in the **Sent Items folder**. Set this parameter to T to instruct the server to save the Message object in the user's Sent Items folder; otherwise, set the parameter to F. The SaveInSent parameter is set to F by default.	**SmartForward**, **SmartReply**, **SendMail**

For more details about specific commands, see [MS-ASCMD].

2.2.1.1.2 Request Headers

The HTTP/1.1 protocol ([RFC2616]) defines several headers that can be sent from the client to the server on an **HTTP** **POST** request. The headers follow the request line in the HTTP portion of a request. The following headers are used in ActiveSync synchronization protocol requests. Note that requests are UTF-8 encoded.

Header	Required	Notes
Accept-Language	No.	For details, see section 2.2.1.1.2.1.
Authorization	Yes, if using basic or **OAuth** authentication.	For details, see section 2.2.1.1.2.2.
Content-Type	Depends on the command.	Specifies that the media type of the request body is WBXML. Other types of content, such as [RFC2822], can also be specified, depending on the command. For more details, see section 2.2.1.1.2.3.
Cookie	Depends on the contents of previous server responses and the protocol version in use.	Contains one or more cookies that the client previously received from the server in a Set-Cookie header. For more details, see section 2.2.1.1.2.4.
MS-ASAcceptMultiPart	No	Specifies that the client wants items returned in multipart format. For more details, see section 2.2.1.1.2.5.
MS-ASProtocolVersion	No if using a base64 encoded query value; yes if using a plain text query value.	Specifies the version of the ActiveSync protocol that the client supports. For more details, see section 2.2.1.1.2.6.
User-Agent	No	Contains information about the client sending the request. For more details, see section 2.2.1.1.2.7.
X-MS-PolicyKey	Depends on the command.	Specifies the policy key assigned by the server to the client. For more details, see section 2.2.1.1.2.8.

2.2.1.1.2.1   Accept-Language

The Accept-Language header is used to define the **locale** of the client which is used when performing searches with the **Find** or **Search** command requests. If the accept language is not specified, the search is conducted by using the server language.

2.2.1.1.2.2   Authorization

Users authenticate through the ActiveSync protocol by using **HTTP** basic authentication, **OAuth** or a client certificate. Credentials are passed in different formats depending upon the form of authentication.

For HTTP basic authentication, credentials are encoded with **base64 encoding**. For user *fakename* and password *x$pIAK9@p9!*, the following is the authorization header:

```
Authorization: Basic ZmFrZXVzZXI6eCRwwSUFLOUBwOSE=
```

For details about HTTP basic authentication, see [RFC1945] section 11.1.

For OAuth, an access token is obtained from the authorization server in response to an authorization grant. The access token is then used to obtain a protected resource from the resource server. The following is an example of an authorization header:

```
Authorization: Bearer <<token>>
```

For details about the OAuth 2.0 framework, see [RFC6749].

For authentication using a client certificate, the client MUST NOT send an authorization header. The server prompts the client for a certificate as part of the initial **SSL/TLS handshake** or as part of a **TLS** renegotiation.

If no client certificate exists, the client MUST complete the SSL/TLS handshake.

For details about providing a client certificate during a SSL/TLS handshake, see [RFC5246] section 7.4.6.

2.2.1.1.2.3   Content-Type

The Content-Type header indicates the format of the data sent in the request body. When the request body for a command is in **WBXML** format, the Content-Type header value MUST be set to either "application/vnd.ms-sync.wbxml", or the shortened string "application/vnd.ms-sync". The shortened string is not allowed by protocol versions 2.5 and 12.0.

For the **Autodiscover** command ([MS-ASCMD] section 2.2.1.1), which specifies an **XML** request body format, the Content-Type header SHOULD be set to "text/xml" or MAY<3> be set to "text/html". If the request has no body, the Content-Type header SHOULD NOT be present.

2.2.1.1.2.4   Cookie

The Cookie header contains one or more cookies that the client previously received from the server in a Set-Cookie header (section 2.2.2.1.2.12). Each cookie consists of the name and value. Multiple cookies are separated by a semi-colon. For details about the syntax, see [RFC6265].

Clients using protocol version 16.0 or 16.1 or clients using **OAuth** authentication for any protocol version MUST be able to parse and interpret a Set-Cookie header that is received from the server. If the server response includes the Set-Cookie header, clients using protocol version 16.0 or 16.1 or clients using OAuth authentication for any protocol version MUST provide these cookies in a Cookie header when sending future requests to the server.

2.2.1.1.2.5   MS-ASAcceptMultiPart

The MS-ASAcceptMultiPart header is used to control the delivery of the content requested by the **Fetch** element ([MS-ASCMD] section 2.2.3.67.1) in an **ItemOperations** request ([MS-ASCMD] section 2.2.1.10). This header is optional for **ItemOperations** requests, and SHOULD NOT be used for other command requests. This header SHOULD NOT be used if the base64-encoded query value is being used. Instead, the **AcceptMultipart** flag in the **Options** parameter SHOULD be used, as specified in section 2.2.1.1.1.3.

If this header is present and the value is 'T', the client is requesting that the server return content in multipart format. If the header is not present, or is present and set to 'F', the client is requesting that the server return content in inline format. For more details, see [MS-ASCMD] section 2.2.1.10.1.

This header is not supported by protocol version 2.5.

2.2.1.1.2.6   MS-ASProtocolVersion

The MS-ASProtocolVersion header indicates the protocol version that the client is using to format the request. This header SHOULD NOT be used if the base64-encoded query value is being used. Instead, the **Protocol version** field of the base64-encoded query value SHOULD be set, as specified in section 2.2.1.1.1.1.

The following values, which correspond to the ActiveSync protocol versions, are valid: "16.1", "16.0", "14.1", "14.0", "12.1", "12.0", and "2.5". The latest version is 16.1.

2.2.1.1.2.7   User-Agent

The format of the User-Agent header is specified in [RFC2616] section 14.43. This header SHOULD be included in command requests.

2.2.1.1.2.8   X-MS-PolicyKey

The X-MS-PolicyKey header contains the client's current policy key, as specified in [MS-ASPROV] section 2.2.2.42. This header SHOULD NOT be used if the base64-encoded query value is being used. Instead, the **Policy key** field of the base64-encoded query value SHOULD be set, as specified in section 2.2.1.1.1.1.

2.2.1.1.3 Request Body

The request body contains data sent to the server. The request body, if any, is in **WBXML**, except the **Autodiscover** command, which is in **XML**. Three commands have no body in certain contexts: **GetAttachment**, **Sync**, and **Ping**. For more details about the request bodies of individual commands, see [MS-ASCMD] section 2.2.1.

2.2.2   HTTP POST Response

After receiving and interpreting a request, a server responds with an **HTTP** response that contains data returned from the server.

2.2.2.1 Response Format

Each command response is sent from the server to the client as an **HTTP POST** response. Note that these responses are UTF-8 encoded. As specified by [RFC2616], the format is the same as for the following requests.

```
Status-line
Response-headers
CR/LF
Message Body
```

2.2.2.1.1 Status Line

The status line consists of the **HTTP** version and a status code. The following is an example of a response status line.

```
HTTP/1.1 200 OK
```

The following table lists some common HTTP status codes.

Status code	Description
200 OK	The command succeeded.
400 Bad Request	The request could not be understood by the server due to malformed syntax. If the client repeats the request without modifications, then the same error occurs.
401 Unauthorized	The resource requires authorization or authorization was refused. For details about the client's handling of this error, see section 3.1.5.2.1.
403 Forbidden	The user is not enabled for ActiveSync synchronization. For details about the client's handling of this error, see section 3.1.5.2.1.
404 Not Found	The specified **URI** could not be found or the server is not a valid server with ActiveSync.
451 Redirect	The device is trying to connect to a server that cannot access the user's **mailbox**, or there is a more efficient server to use to reach the user's mailbox. For details about the client's handling of this error, see section 3.1.5.2.2.
456 Blocked	The user's account is blocked. For details about the client's handling of this error, see section 3.1.5.2.4.
457 Expired Password	The user's password has expired. For details about the client's handling of this error, see section 3.1.5.2.4.
500 Internal Server Error	The server encountered an unexpected condition that prevented it from fulfilling the request. For details about the client's handling of this error, see section 3.1.5.2.1.
501 Not Implemented	The server does not support the functionality that is required to fulfill the request. This status code SHOULD be returned by the server when the server does not recognize the request method or is not able to support it for any resource. In the case of other malformed requests, the server returns status code 400.
502 Proxy Error	The specified server could not be found.
503 Service Unavailable	The service is unavailable. For details about the client's handling of this error, see section 3.1.5.2.3.
507 Insufficient Disk Space	The user's mailbox is full.

2.2.2.1.2 Response Headers

This protocol and [RFC2616] define several headers that can be sent from the server to the client in an **HTTP** **POST** response. The headers follow the status line in the HTTP part of a response. The following table lists some common headers that can be set by the server in response to client requests.

Header	Example value	Notes
Cache-Control	private	Optional. Controls how the response is cached.
Content-Encoding	gzip	Required when the content is compressed; otherwise, this header is not included. Specifies the HTTP

Header	Example value	Notes
		compression format that is used in the response.
Content-Length	56	Optional. Specifies the size of the response body in bytes.
Content-Type	application/vnd.ms-sync.wbxml	Required. Specifies that the media-type of the response body is **WBXML**. Other types of content, such as [RFC2822], can also be specified.
MS-Server-ActiveSync	15.1	Optional. Indicates the version of the ActiveSync server that was used to handle the request.
X-MS-Location	https://mail.contoso.com/Microsoft-Server-ActiveSync	Optional. Used in conjunction with a 451 Redirect status code. Specifies the **URL** to use for future requests.
MS-ASProtocolCommands	Sync,SendMail,SmartForward,SmartReply, GetAttachment,GetHierarchy,CreateCollection, DeleteCollection,MoveCollection,FolderSync, FolderCreate,FolderDelete,FolderUpdate, MoveItems,GetItemEstimate,MeetingResponse, Search,Settings,Ping,ItemOperations, Provision,ResolveRecipients,ValidateCert, Find	Optional. Indicates the commands supported by the server.
MS-ASProtocolVersions	2.5,12.0,12.1,14.0,14.1,16.0,16.1	Optional. Indicates the protocol versions supported by the server.
X-MS-RP	2.5,12.0,12.1,14.0,14.1,16.0,16.1	Optional. Indicates to the client that the client has to perform a full resynchronization.
X-MS-Credential-Service-Url	https://portal.microsoftonline.com/ChangePassword.aspx	Optional. Contains a URL for reset of user's password.
X-MS-Credentials-Expire	13	Optional. Indicates the number of days remaining until expiration of a user's password.
Set-Cookie	X-Cookie=value; expires=Wed, 08-Jul-2015 23:40:27 GMT; path=/Microsoft-Server-ActiveSync; secure; HttpOnly	Optional. Contains one or more cookies returned by the server.

Header	Example value	Notes
X-BEServer	EXCH-SERV-1	Optional. Contains the name of the server that processed the request.
X-FEServer	EXCH-SERV-1	Optional. Contains the name of the server(s) that routed the request.
request-id	7faa449e-4912-4a79-aade-afee642c2c36	Optional. Contains a server-generated identifier for the request.

When protocol version 12.1, 14.0, 14.1, 16.0, or 16.1 is used: Some of the headers in the response can be eliminated when the response is to an HTTP **POST** request and the response has HTTP status 200. When these two conditions are met, only the following headers are necessary in the response:

- Content-Length

- Content-Type, required only if Content-Length is greater than zero.

2.2.2.1.2.1   Cache-Control

This header is optional. The value of this header controls how the response is cached, as specified in [RFC2616] section 14.9.

2.2.2.1.2.2   Content-Encoding

This header is required if the response body is compressed. Otherwise, it is omitted. See [RFC2616] section 14.11.

2.2.2.1.2.3   Content-Length

This header is optional. The format is specified in [RFC2616] section 14.13.

2.2.2.1.2.4   Content-Type

This header is required. If the response body is **WBXML**, the value of this header MUST be "application/vnd.ms-sync.wbxml". Otherwise, an appropriate value SHOULD be used as specified in [RFC2616] section 14.17.

2.2.2.1.2.5   MS-Server-ActiveSync

This header is optional. It contains an implementation-specific string indicating the version of the server.

2.2.2.1.2.6   X-MS-Location

This header is optional. It is used when the **HTTP** status code is 451 to provide a **URL** to use for subsequent requests.

2.2.2.1.2.7   MS-ASProtocolCommands

The MS-ASProtocolCommands header contains a comma-delimited list of the ActiveSync commands supported by the server. It will be returned in an HTTP POST response if the server requires the client to reinitialize its synchronization state.

2.2.2.1.2.8 MS-ASProtocolVersions

The MS-ASProtocolVersions header contains a comma-delimited list of the ActiveSync protocol versions that the server supports. It will be returned in HTTP POST response headers if the server requires the client to reinitialize its synchronization state.

The following values correspond to the ActiveSync protocol versions that are specified by [MS-ASCMD]: "16.1", "16.0", "14.1", "14.0", "12.1", "12.0", and "2.5". The latest version is 16.1.

2.2.2.1.2.9 X-MS-RP

This header is optional. Its presence in a response indicates that a condition on the server (such as a server upgrade) requires the client to discard its local data and resynchronize. The value of this header indicates the protocol versions the server supports.

This header is not supported by protocol version 2.5.

2.2.2.1.2.10 X-MS-Credential-Service-Url

This header is optional. This header contains the **URL** for a self-service web site that allows a user to reset the user password.

This header is required in the response to an **Autodiscover** command request ([MS-ASCMD] section 2.2.1.1) when the user's password is either near expiration or expired. The point at which a password is near expiration is determined by the implementer.

2.2.2.1.2.11 X-MS-Credentials-Expire

This header is optional. This header contains an integer that indicates the number of days remaining until the user's password expires. A value of 0 (zero) indicates that the password will expire in less than 24 hours.

This header can be included in a response to provide advance warning of password expiration.

2.2.2.1.2.12 Set-Cookie

This header is optional. The Set-Cookie header contains one cookie returned by the server. Each cookie consists of a name, a value, and the following attributes: *Expires*, *Path*, *Secure*, and *HttpOnly*. Multiple instances of this header can be returned with a different cookie name in each instance of the header.

For details about the syntax of this header, see [RFC6265].

2.2.2.1.2.13 X-MS-ASThrottle

This header is optional. The X-MS-ASThrottle header specifies the condition under which the server MAY<4> throttle the client device.

2.2.2.1.2.14 X-BEServer

This header is optional. The X-BEServer header contains the name of the server that processed the request.

2.2.2.1.2.15 X-FEServer

This header is optional. The X-FEServer header contains the name of the server(s) that routed the request.

2.2.2.1.2.16 request-id

This header is optional. The request-id header contains a server-generated identifier for the request.

2.2.2.1.3 Response Body

The response body contains data returned from the server. The response body, if any, is in **WBXML**, except the **Autodiscover** command, which is in **XML**. Two commands have no XML body in certain contexts: **GetAttachment** and **Sync**. For more details about the response bodies of individual commands, see [MS-ASCMD] section 2.2.1.

2.2.3   HTTP OPTIONS Request

The **HTTP OPTIONS** command, which is specified by [RFC2616], is used to discover what protocol versions are supported, and which protocol commands are supported on the server. The client uses the HTTP **OPTIONS** command to determine whether the server supports the same versions of the protocol that the client supports.

2.2.3.1 Request Format

As specified by [RFC2616], the format is as follows.

```
Request-line
Request-headers
```

2.2.3.1.1 Request Line

The request line consists of the method indicator, **OPTIONS**, followed by the **URI**, followed by the **HTTP** version, as follows.

```
OPTIONS <URI> HTTP/1.1
```

The URI can be either an absolute URI or a relative URI, as specified in [RFC2616] section 3.2.1. The absolute URI consists of a scheme indicator, the host name, and the path. The relative URI consists of the path.

The path in the URI has the following format.

```
/<ActiveSync virtual directory name>
```

2.2.3.1.2 Request Headers

The authorization header is required. For more information on the authorization header requirements, see section 2.2.1.1.2.2.

2.2.4   HTTP OPTIONS Response

After receiving an **HTTP OPTIONS** request, a server responds with an HTTP **OPTIONS** response that specifies the protocol versions it supports.

2.2.4.1 Response Format

Each response is sent from the server to the client as an **HTTP OPTIONS** response. Note that these responses are UTF-8 encoded. As specified by [RFC2616], the format is the same as for the following requests:

```
Status-line
Response-headers
```

2.2.4.1.1 Status Line

The status line for an **HTTP OPTIONS** response is identical to the status line for an HTTP **POST** response, specified in section 2.2.2.1.1.

2.2.4.1.2 Response Headers

This protocol defines headers that can be sent from the server to the client in an **HTTP OPTIONS** response in addition to headers defined in [RFC2616]. The headers follow the status line in the HTTP part of a response.

Header	Example value	Notes
MS-ASProtocolCommands	Sync,SendMail,SmartForward,SmartReply, GetAttachment,GetHierarchy,CreateCollection, DeleteCollection,MoveCollection,FolderSync, FolderCreate,FolderDelete,FolderUpdate, MoveItems,GetItemEstimate,MeetingResponse, Search,Settings,Ping,ItemOperations, Provision,ResolveRecipients,ValidateCert,Find	Indicates the commands supported by the server.
MS-ASProtocolVersions	2.5,12.0,12.1,14.0,14.1,16.0,16.1	Indicates the protocol versions supported by the server.
Set-Cookie	X-Cookie=value; expires=Wed, 08-Jul-2015 23:40:27 GMT; path=/Microsoft-Server-ActiveSync; secure; HttpOnly	Optional. Contains one or more cookies returned by the server.

2.2.4.1.2.1   MS-ASProtocolCommands

The MS-ASProtocolCommands header contains a comma-delimited list of the ActiveSync commands supported by the server.

2.2.4.1.2.2   MS-ASProtocolVersions

The MS-ASProtocolVersions header contains a comma-delimited list of the ActiveSync protocol versions that the server supports.

The following values correspond to the ActiveSync protocol versions that are specified by [MS-ASCMD]: "16.1", "16.0", "14.1", "14.0", "12.1", "12.0", and "2.5". The latest version is 16.1.

2.2.4.1.2.3   Set-Cookie

For details about this header see section 2.2.2.1.2.12.

3   Protocol Details

3.1   Client Details

3.1.1   Abstract Data Model

None.

3.1.2   Timers

None.

3.1.3   Initialization

The client SHOULD send an **Autodiscover** command request ([MS-ASCMD] section 2.2.1.1) to the server to determine the correct server **URL** to use for all subsequent commands.

After determining the correct server URL, the client SHOULD send an **HTTP OPTIONS** command to the server, as specified in section 2.2.3. The client SHOULD<5> use the most recent version (the greatest numbered version) of the protocol that is supported by the client and server.

3.1.4   Higher-Layer Triggered Events

Synchronizing changes on the client requires the client to send a command to the server.

3.1.4.1   Sending a Command Request

Command requests MUST be formatted as specified in section 2.2.1 and sent via **HTTP**. **Secure Sockets Layer (SSL)** SHOULD be enabled between the client and the server whenever the Authorization header (section 2.2.1.1.2.2) is sent. The client SHOULD wait for a server response to the request.

Clients that include the User-Agent HTTP header SHOULD NOT change the value of this header between consecutive command requests, unless a major change to the client has occurred, such as an operating system upgrade.

3.1.5   Message Processing Events and Sequencing Rules

Clients receive HTTP responses from the server only in response to HTTP requests sent by the client.

3.1.5.1   Handling a Successful Response

If the HTTP status code indicates that the request succeeded (its value is between 200 and 299, as specified in [RFC2616]), the response is interpreted as specified in [MS-ASCMD] section 2.2.1.

If the server returns an X-MS-RP header in the response, the client MUST reinitialize its synchronization state as specified in [MS-ASCMD] section 2.2.1.21. If the X-MS-RP header is received in response to a **FolderSync** request that has a synchronization key of 0, the client can ignore the X-MS-RP header.

If the server returns the X-MS-Credentials-Expire header (section 2.2.2.1.2.11) in the response, the client SHOULD send an **Autodiscover** command request ([MS-ASCMD] section 2.2.1.1) to retrieve the **URL** for the self-service web site that allows a user to do a password reset. The **Autodiscover**

command response will include the X-MS-Credential-Service-Url header (section 2.2.2.1.2.10), which contains the URL for the self-service web site. The client SHOULD provide this URL to the user.

3.1.5.2 Handling a Failed Response

Any **HTTP** status code that is not between 200 and 299, as specified in [RFC2616], indicates that the request failed. The following sections specify the client's handling of certain HTTP status codes that are returned by the server when a request fails. All other HTTP status codes that indicate a failed request are interpreted and handled as specified in [RFC2616].

3.1.5.2.1 HTTP Error 401, 403, and 500

If the server responds to any command with an **HTTP** error 401, 403, or 500, the client SHOULD send an **Autodiscover** command request ([MS-ASCMD] section 2.2.1.1) to the server.

3.1.5.2.2 HTTP Error 451

If the client is attempting to connect to the wrong server (that is, a server that cannot access the user's mailbox), or if there is a more efficient server to use to reach the user's mailbox, then a 451 Redirect error is returned.

The error returned by the wrong server resembles the following:

```
OPTIONS /Microsoft-Server-ActiveSync
Content-Type: application/vnd.ms-sync.wbxml
MS-ASProtocolVersion: 14.0

HTTP/1.1 451
Date: Tue, 08 Dec 2009 19:43:24 GMT
Server: Microsoft-IIS/7.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
X-MS-Location: https://mail.contoso.com/Microsoft-Server-ActiveSync
Cache-Control: private
Content-Length: 0
```

If an X-MS-Location header is present in the response, all subsequent requests SHOULD use the URL specified within the X-MS-Location header. If the server does not provide an X-MS-Location header in its response to the client, then the full **Autodiscover** command process is followed, as specified in [MS-ASCMD].

3.1.5.2.3 HTTP Error 503

The server returns an HTTP error 503 when more users than are allowed by the server's request queue limit have sent requests to a single server or when the actions of the client have triggered throttling.

The error returned by the server resembles the following.

```
OPTIONS /Microsoft-Server-ActiveSync
Content-Type: application/vnd.ms-sync.wbxml
MS-ASProtocolVersion: 14.0

HTTP/1.1 503 Service Unavailable
Connection: close
Date: Mon, 02 Mar 2009 23:51:51 GMT
Server: Microsoft-IIS/7.0
X-Powered-By: ASP.NET
Content-Type: text/html
```

If a Retry-After header ([RFC2616]) is present in the response, the client SHOULD<6> retry the request after waiting the number of seconds indicated by the Retry-After header. Any such value represents an estimate of when the server is expected to be able to process the request.

If a Retry-Header is not present in the response, the client can retry the request after waiting a few seconds. The time to wait between continuous requests that result in HTTP error 503 responses can be increased exponentially to a predetermined maximum.

For more details about ASP.NET performance monitoring properties, see [MSDN-APM].

3.1.5.2.4 HTTP Error 456 and 457

If the server responds to an **Autodiscover** command request ([MS-ASCMD] section 2.2.1.1) with an **HTTP** error 456, the client SHOULD stop sending requests to the server and then prompt the user to contact the administrator.

If the server responds to an **Autodiscover** command request with an HTTP error 457, the client SHOULD stop sending requests to the server and then prompt the user to reset the user password. The client SHOULD direct the user to the self-service web site that allows a user to do a password reset. The **URL** for this web site is contained in the X-MS-Credential-Service-Url header (section 2.2.2.1.2.10) of the server's response.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2   Server Details

The server only responds to client requests by returning HTTP responses as specified in section 2.2.2 and never initiates communication with the client.

3.2.1 Abstract Data Model

None.

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Message Processing Events and Sequencing Rules

The server can receive **HTTP POST** requests (section 2.2.1) or HTTP **OPTIONS** requests (section 2.2.3) from the client.

3.2.5.1 Handling HTTP POST Command Requests

The server parses HTTP **POST** requests from clients as specified in section 2.2.1. The ActiveSync command contained within an HTTP **POST** request is parsed as specified in [MS-ASCMD] section 2.2.1. The server MUST conform to the protocol version that is specified by the MS-ASProtocolVersion header (section 2.2.1.1.2.6) in the client request. The server formats an HTTP **POST** response, as specified in section 2.2.2, with an appropriate **HTTP** status code, as specified in section 2.2.2.1.1.

If the server returns an HTTP 451 error and knows the **URL** of the correct server, it SHOULD include an X-MS-Location header (section 2.2.2.1.2.6) with the URL of the correct server. If the server returns an HTTP 503 error, it MAY<7> include a Retry-After header, as specified in [RFC2616], in the response with an estimate of the number of seconds that will elapse before the server is expected to be able to process the request. If the server returns an HTTP 457 error, it MUST include the X-MS-Credential-Service-Url header (section 2.2.2.1.2.10) in its response.

If the user's password is near expiration, the server SHOULD include the X-MS-Credentials-Expire header (section 2.2.2.1.2.11) in its response. This header provides advance warning to the client of password expiration. The point at which the server begins this warning is determined by the implementer.

If the client sends a request to synchronize the folder hierarchy with a synchronization key of 0 or the server requires the client to reinitialize its synchronization state, the server SHOULD include an X-MS-RP header, an MS-ASProtocolCommands header, and an MS-ASProtocolVersions header in its response to the client.

The server MAY<8> track the value of the User-Agent header for consecutive command requests from a specific device and block devices that change the value of this header more than a configured limit within a configured timespan. Servers that do this tracking SHOULD use the algorithm specified in section 3.2.5.1.1.

3.2.5.1.1 User-Agent Change Tracking

Servers SHOULD limit changes to the User-Agent header value from a device to two changes within a 24-hour time period, but MAY<9> use different values for the number of changes or the time period. The server SHOULD block clients that exceed this limit for 14 hours, but MAY<10> block clients for a different amount of time.

3.2.5.2 Handling HTTP OPTIONS Command Requests

The server parses HTTP **OPTIONS** requests from clients as specified in section 2.2.3 and formats its response as specified in section 2.2.4. The server's response MUST contain both the MS-ASProtocolCommands header, as specified in section 2.2.4.1.2.1, and the MS-ASProtocolVersions header, as specified in section 2.2.4.1.2.2. The server uses these headers to indicate which ActiveSync commands and which ActiveSync protocol versions it supports.

A protocol server can support multiple versions of the ActiveSync protocol. This specification, and any protocol specifications that cite it as a dependency, apply to the server configuration when the value of the MS-ASProtocolVersions header is set to a value that includes "16.1", "16.0", "14.1", "14.0", "12.1", "12.0", or "2.5".<11>

The latest version of the ActiveSync protocol is 16.1. Older versions are 16.0, 14.1, 14.0, 12.1, 12.0, and 2.5. Some commands and functionality described in the ActiveSync protocol documentation are not supported by all of the protocol versions. See the command and element descriptions in the ActiveSync protocol documents to determine which commands, elements, and capabilities are supported by the protocol versions.

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

4   Protocol Examples

4.1   FolderSync Request and Response

The following is a typical ActiveSync protocol command request. The **FolderSync** command, user **alias**, device ID, and device type are specified as **URI** query parameters. The Content-Type header specifies that the request body is WBXML. The MS-ASProtocolVersion header specifies that protocol 14.0 is being used. Some command requests contain additional URI query parameters or do not specify a request body. The **HTTP POST URI** command parameter is the same as the command in the topmost element of the request XML body. For details about the commands and associated **XML schema definitions (XSDs)**, see [MS-ASCMD]. The WBXML-encoded body is decoded for clarity.

Request

```
POST /Microsoft-Server-
ActiveSync?Cmd=FolderSync&User=fakename&DeviceId=v140Device&DeviceType=SmartPhone HTTP/1.1
Content-Type: application/vnd.ms-sync.wbxml
MS-ASProtocolVersion: 14.0
User-Agent: ASOM
Host: Contoso.com

<?xml version="1.0" encoding="utf-8"?>
<FolderSync xmlns="FolderHierarchy:">
  <SyncKey>2</SyncKey>
</FolderSync>
```

The following is a typical FolderSync command response. The status line specifies the HTTP/1.1 protocol and that the command succeeded. The Content-Length header specifies that the response body is 56 bytes and the Content-Type header shows that the response body is in WBXML format. Some command responses do not contain WBXML bodies.

Response

```
HTTP/1.1 200 OK
Content-Type: application/vnd.ms-sync.wbxml

Date: Thu, 12 Mar 2009 19:34:31 GMT
Content-Length: 25

<?xml version="1.0" encoding="utf-8"?>
<FolderSync>
  <Status>1</Status>
  <SyncKey>2</SyncKey>
  <Changes>
    <Count>0</Count>
  </Changes>
</FolderSync>
```

4.2   FolderSync Request and Redirect Response

The following is the same request from the example described in section 4.1. In this example, configuration changes on the server have caused the "contoso.com" host to no longer be the optimal host for the user.

Request

```
POST /Microsoft-Server-
ActiveSync?Cmd=FolderSync&User=fakename&DeviceId=v140Device&DeviceType=SmartPhone HTTP/1.1
Content-Type: application/vnd.ms-sync.wbxml
MS-ASProtocolVersion: 14.0
User-Agent: ASOM
Host: Contoso.com

<?xml version="1.0" encoding="utf-8"?>
<FolderSync xmlns="FolderHierarchy:">
  <SyncKey>2</SyncKey>
</FolderSync>
```

The server redirects the client to the "mail.contoso.com" host using an **HTTP** status code 451 and the
X-MS-Location header.

```
HTTP/1.1 451
Date: Thu, 12 Mar 2009 20:16:22 GMT
X-MS-Location: https://mail.contoso.com/Microsoft-Server-ActiveSync
Content-Length: 0
```

4.3   HTTP OPTIONS Command Request and Response

The following example illustrates the use of the **HTTP OPTIONS** command. The MS-
ASProtocolVersions header in the server response shows that versions 1.0, 2.0, 2.1, 2.5, 12.0, 12.1,
and 14.0 of the protocol are supported on the server. The MS-ASProtocolCommands header in the
server response lists the commands that are supported. It is recommended that protocol clients not
trigger on the build number of the protocol server, which can change because of server updates. The
build number shown in the examples might differ from those seen in a development or production
environment.

Request

```
OPTIONS /Microsoft-Server-ActiveSync HTTP/1.1
Host: Contoso.com
```

Response

```
HTTP/1.1 200 OK
Cache-Control: private
Allow: OPTIONS,POST
Server: Microsoft-IIS/7.0
MS-Server-ActiveSync: 14.00.0536.000
MS-ASProtocolVersions: 2.0,2.1,2.5,12.0,12.1,14.0
MS-ASProtocolCommands: Sync,SendMail,SmartForward,SmartReply,GetAttachment,GetHierarchy,
CreateCollection,DeleteCollection,MoveCollection,FolderSync,FolderCreate,
FolderDelete,FolderUpdate,MoveItems,GetItemEstimate,MeetingResponse,Search,
Settings,Ping,ItemOperations,Provision,ResolveRecipients,ValidateCert
Public: OPTIONS,POST
X-AspNet-Version: 2.0.50727
X-Powered-By: ASP.NET
Date: Thu, 12 Mar 2009 20:03:29 GMT
Content-Length: 0
```

4.4   SendMail Request and Response

The following example illustrates the command to send mail to a specific user.

Request

```
POST /Microsoft-Server-
ActiveSync?Cmd=SendMail&User=fakeusername&DeviceId=v140Device&DeviceType=SmartPhone HTTP/1.1
Content-Type: application/vnd.ms-sync.wbxml
MS-ASProtocolVersion: 14.0
X-MS-PolicyKey: 2034202722
User-Agent: ASOM
Host: BIRSKK-dom.extest.microsoft.com

<?xml version="1.0" encoding="utf-8"?>
<SendMail
xmlns="ComposeMail:">
  <ClientId>633724606026842453</ClientId>
  <Mime>From: fakeuser@Contoso.com
To: fakeuser@Contoso.com
Cc:
Bcc:
Subject: From NSync
MIME-Version: 1.0
Content-Type: text/plain; charset="iso-8859-1"
Content-Transfer-Encoding: 7bit
X-MimeOLE: Produced By Microsoft MimeOLE V6.00.2900.3350
This is the body text.</Mime>
</SendMail>
```

Response

```
HTTP/1.1 200 OK
s
Date: Thu, 12 Mar 2009 20:16:22 GMT
Content-Length: 0
```

4.5   CreateFolder Request and Response

The following example illustrates the command to create a new folder. For details about the associated XML schema definitions (XSD), see [MS-ASCMD].

Request:

```
POST /Microsoft-Server-
ActiveSync?Cmd=FolderCreate&User=fakename@Contoso.com&DeviceId=v140Device&DeviceType=SmartPho
ne HTTP/1.1
Content-Type: application/vnd.ms-sync.wbxml
MS-ASProtocolVersion: 14.0
User-Agent: ASOM
Host: Contoso.com

<?xml version="1.0" encoding="utf-8"?>
<FolderCreate xmlns="FolderHierarchy:">
  <SyncKey>3</SyncKey>
  <ParentId>5</ParentId>
  <DisplayName>CreateNewFolder</DisplayName>
  <Type>12</Type>
</FolderCreate>
```

Response:

```
HTTP/1.1 200 OK
```

```
Content-Type: application/vnd.ms-sync.wbxml

Date: Thu, 12 Mar 2009 20:s26:06 GMT
Content-Length: 24

<?xml version="1.0" encoding="utf-8"?>
<FolderCreate
xmlns="FolderHierarchy:">
  <Status>1</Status>
  <SyncKey>4</SyncKey>
  <ServerId>23</ServerId>
</FolderCreate>
```

5 Security

5.1 Security Considerations for Implementers

There are no special security considerations specific to this specification. It is recommended that communication between the client and server occur across an HTTP connection secured by the **Secure Sockets Layer (SSL)** protocol.

When connecting to a server using SSL, clients are required to support server certificates that use the Subject Alternative Name for domain names, as specified in [RFC4985], as well as wildcard certificate names, as specified in [RFC2818] and [RFC3280].

5.2 Index of Security Parameters

None.

6   Appendix A: Product Behavior

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

- Microsoft Exchange Server 2007 Service Pack 1 (SP1)

- Microsoft Exchange Server 2010

- Microsoft Exchange Server 2013

- Microsoft Exchange Server 2016

- Windows 8.1

- Windows 10 operating system

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms "SHOULD" or "SHOULD NOT" implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term "MAY" implies that the product does not follow the prescription.

<1> Section 2.2.1.1.1.1: Exchange 2007 SP1 and the initial release version of Exchange 2010 do not set the **Protocol version** field to 141, 160 or 161. Microsoft Exchange Server 2010 Service Pack 1 (SP1) and Exchange 2013 do not set the **Protocol version** field to 160 or 161.

<2> Section 2.2.1.1.1.1: Exchange 2007 SP1 sets the **Protocol version** field to 121. The initial release version of Exchange 2010 sets the **Protocol version** field to 140.

<3> Section 2.2.1.1.2.3: Exchange 2007 SP1 accepts a Content-Type header value of either "text/xml" or "text/html" for the **Autodiscover** command.

<4> Section 2.2.2.1.2.13: The X-MS-ASThrottle header and throttling are not supported in Exchange 2007 SP1 and Exchange 2010. The X-MS-ASThrottle header and throttling are supported in Exchange 2013 and Exchange 2016 but are disabled by default.

<5> Section 3.1.3: Windows Communication Apps only support protocol versions 12.1 and 14.0.

<6> Section 3.1.5.2.3: Windows Communication Apps ignore the Retry-After header and instead retry after a set time. The set time increases exponentially when multiple 503 errors are received.

<7> Section 3.2.5.1: Exchange 2010 and Exchange 2013 sometimes include a Retry-After header with HTTP 503 error responses.

<8> Section 3.2.5.1: Exchange 2013 and Exchange 2016 can be configured to track changes to the User-Agent header, but do not do so by default.

<9> Section 3.2.5.1.1: Exchange 2013 and Exchange 2016 can be configured to use different values for the allowed number of changes and the time period.

<10> Section 3.2.5.1.1: Exchange 2013 and Exchange 2016 can be configured to block clients for an amount of time other than 14 hours.

<11> Section 3.2.5.2: Exchange 2007 SP1 does not return the value "16.1", "16.0", "14.1", or "14.0" in the MS-ASProtocolVersions header. The initial release version of Exchange 2010 does not return the

value "16.1", "16.0" or "14.1" in the MS-ASProtocolVersions header. Exchange 2010 SP1 and Exchange 2013 do not return the value "16.1" or "16.0" in the MS-ASProtocolVersions header.

7 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as Major, Minor, or None.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

▪ A document revision that incorporates changes to interoperability requirements.
▪ A document revision that captures changes to protocol functionality.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **None** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

The changes made to this document are listed in the following table. For more information, please contact dochelp@microsoft.com.

Section	Description	Revision class
2.2.1.1.2.2 Authorization	Updated description for the Authorization header.	Minor
2.2.2.1.2 Response Headers	Added optional debugging headers X-BEServer, X-FEServer, and request-id to the table.	Major
2.2.2.1.2.14 X-BEServer	Added section for this header.	Major
2.2.2.1.2.15 X-FEServer	Added section for this header.	Major
2.2.2.1.2.16 request-id	Added section for this header.	Major

8 Index