[MS-ASHTTP]:
ActiveSync HTTP Protocol Specification

Revision Summary

Date	Revision History	Revision Class	Comments
12/03/2008	1.0.0	Major	Initial release.
02/04/2009	1.0.1	Editorial	Revised and edited technical content.
03/04/2009	1.0.2	Editorial	Revised and edited technical content.
04/10/2009	2.0.0	Major	Updated technical content and applicable product releases.
07/15/2009	3.0.0	Major	Revised and edited for technical content.
11/04/2009	4.0.0	Major	Updated and revised the technical content.
02/10/2010	5.0.0	Major	Updated and revised the technical content.
05/05/2010	6.0.0	Major	Updated and revised the technical content.
08/04/2010	7.0	Major	Significantly changed the technical content.
11/03/2010	7.1	Minor	Clarified the meaning of the technical content.
03/18/2011	8.0	Major	Significantly changed the technical content.
08/05/2011	8.1	Minor	Clarified the meaning of the technical content.
10/07/2011	8.2	Minor	Clarified the meaning of the technical content.
01/20/2012	9.0	Major	Significantly changed the technical content.
04/27/2012	9.1	Minor	Clarified the meaning of the technical content.
07/16/2012	10.0	Major	Significantly changed the technical content.
10/08/2012	11.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction

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

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

1.1 Glossary

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

Augmented Backus-Naur Form (ABNF)
GUID
Hypertext Transfer Protocol (HTTP)
Hypertext Transfer Protocol over Secure Sockets Layer (HTTPS)
Secure Sockets Layer (SSL)
XML

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

alias
base64 encoding
calendar
contact
encrypted message
Global Address List (GAL)
Inbox folder
locale
mailbox
meeting
meeting request
Message object
MIME message
Out of Office (OOF)
plain text
recipient
S/MIME (Secure/Multipurpose Internet Mail Extensions)
Sent Items folder
Uniform Resource Identifier (URI)
Uniform Resource Locator (URL)
Wireless Application Protocol (WAP) Binary XML (WBXML)
XML schema definition (XSD)

The following terms are specific to this document:

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.

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

1.2 References

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

1.2.1 Normative References

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

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

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

[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://ietf.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.ietf.org/rfc/rfc2616.txt

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

[RFC2822] Resnick, P., Ed., "Internet Message Format", STD 11, 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

[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", July 2003, http://msdn.microsoft.com/en-us/library/ms972959.aspx

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

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

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

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 protocol versions and commands.

1.4 Relationship to Other Protocols

This protocol uses an HTTP connection between the client and server. A TCP/IP network transports messages between a client and server by using the HTTP 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**, 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. The MS-ASProtocolVersions header (section 2.2.4.1.2.2) is examined to determine the supported 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 absolute URI (as specified in [RFC2616] section 3.2.1) consists of a scheme indicator, the host name, and the path, followed by a query value that contains all the parameters and some of the request headers. The query value can be either **plain text** or a byte sequence encoded with base64 encoding. The relative URI consists of the path and the query value. Either form can be used.

Using base64 encoding is optional. For more details about base64 encoding, see [RFC2045].

The format of the path is either of the following:

```
/Microsoft-Server-ActiveSync?<text query value>
/Microsoft-Server-ActiveSync?<base64-encoded query value>
```

The following examples are equivalent:

```
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?jAAJBAp2MTQwRGV2aWNlAApTbWFydFBob251 HTTP/1.1
Content-Type: application/vnd.ms-sync.wbxml
User-Agent: ASOM
Host: Contoso.com
Content-Length: 866
```

For more details about query values encoded with base64 encoding, see section 2.2.1.1.1.1.

2.2.1.1.1.1 Base64 Encoded Query Value

The following is an example of a URI query value encoded with base64 encoding:

```
/Microsoft-Server-ActiveSync?jAAJBAp2MTQwRGV2aWNlAApTbWFydFBob251
```

The sequence of bytes represents the URI request parameters. The following table provides the details of the sequence of bytes. This byte sequence is divided into fields in the same order as the following table. Once the byte sequence is generated, it is encoded with base64 encoding prior to being appended to the request URI.

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	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. 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 2.2.1.1.1.2).

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. A value of 0 indicates that the **Device ID** field is absent.

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 2.2.2.41. 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. This field is only present if there are command-specific parameters associated with the command specified by the **Command code** field. See section 2.2.1.1.1.3 for a list of command-specific parameters.

2.2.1.1.1.1 Encoded Parameter

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
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 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.

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**. See for more details.

2.2.1.1.1.2 Plain Text Query Value

The **Augmented Backus-Naur Form (ABNF)** notation ([RFC5234]) is used to specify the format of the plain text query value.

```
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**

Parameter	Description	Used by
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
Authorization	Yes	Specifies that user credentials are sent by using HTTP basic authentication. For details, see section 2.2.1.1.2.1.
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 2.2.1.1.2.2.
MS-ASAcceptMultiPart	No	Specifies that the client wants items returned in multipart format. For more details, see section 2.2.1.1.2.3.
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 2.2.1.1.2.4.
User-Agent	No	Contains information about the client sending the request. For more details, see section 2.2.1.1.2.5.
X-MS-PolicyKey	Depends on the command.	Specifies the policy key assigned by the server to the client. For more details, see section 2.2.1.1.2.6.

2.2.1.1.2.1 Authorization

User credentials are sent from the client to the server by using HTTP basic authentication, in which the credentials are encoded with base64 encoding. For user *fakename* and password *x$pIAK9@p9!*, the Authorization header would be as follows.

```
Authorization: Basic ZmFrZXVzZXI6eCRwSUFLOUBwOSE=
```

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

2.2.1.1.2.2 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". For the **Autodiscover** command ([MS-ASCMD] section 2.2.2.1), 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.

2.2.1.1.2.3 MS-ASAcceptMultiPart

The MS-ASAcceptMultiPart header is used to control the delivery of the content requested by the **Fetch** element ([MS-ASCMD] section 2.2.3.63.1) in an **ItemOperations** request ([MS-ASCMD] section 2.2.2.8). 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 2.2.2.8.1.

2.2.1.1.2.4 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.

Valid values for this header are "14.1", "14.0", and "12.1".

2.2.1.1.2.5 User-Agent

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

2.2.1.1.2.6 X-MS-PolicyKey

The X-MS-PolicyKey header contains the client's current policy key, as specified in [MS-ASPROV] section 2.2.2.41. 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 2.2.2.

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 how to troubleshoot this error, see section 3.1.5.2.1.
403 Forbidden	The user is not enabled for ActiveSync synchronization. For details about how to troubleshoot 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.
500 Internal Server Error	The server encountered an unexpected condition that prevented it from fulfilling the request. For details about how to troubleshoot 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 how to troubleshoot 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	Controls how the response is cached.
Content-Encoding	gzip	Required when the content is compressed; otherwise, this header is not included. Specifies the HTTP compression format that is used in the response.
Content-Length	56	Required. 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	8.1	Optional. Indicates the version of the ActiveSync server that was used to handle the request.
X-MS-Location		Optional. Used in conjunction with a 451 Redirect status code. Specifies the **URL** to use for future requests.
X-MS-RP	12.1,14.0, 14.1	Optional. Indicates to the client that the client has to perform a full resynchronization because of a server upgrade.

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

2.2.2.1.2.1 Cache-Control

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 required. 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 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.

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. Three commands have no XML body in certain contexts: **GetAttachment**, **Sync**, and **Ping**. For more details about the response bodies of individual commands, see [MS-ASCMD] section 2.2.2.

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 absolute URI consists of a scheme indicator, the host name, and the path. The relative URI consists of the path. Either form can be used.

The format of the path is the following:

```
/Microsoft-Server-ActiveSync
```

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	Indicates the protocol versions supported by the server.
MS-ASProtocolVersions	12.1, 14.0, 14.1	Indicates the protocol versions supported 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 indicates the versions of the ActiveSync protocol supported by the server. Values that correspond to the versions specified by [MS-ASCMD] are "12.1", "14.0", and "14.1".

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 2.2.2.1) 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 2.2.1.1.2.4. The client SHOULD<4> 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 2.2.1.1.2.1) 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 2.2.2.

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 2.2.2.19.

3.1.5.2 Handling a Failed Response

Unless otherwise specified in the following sections, all HTTP status codes that do not indicate success are interpreted and handled as specified in [RFC2616].

3.1.5.2.1 HTTP Error 401, 403, and 500

The client SHOULD send an **Autodiscover** command request ([MS-ASCMD] section 2.2.2.1) to the server if the server responds to any command with an HTTP Error 401, 403, or 500.

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 2.2.2.19.

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

One of the causes of HTTP error 503 is that more users than are allowed by the server's request queue limit have sent requests to a single server.

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<5> 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].

3.1.6 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 Requests

The server SHOULD parse requests from clients as specified in section 2.2.1 and [MS-ASCMD] section 2.2.2. The server MUST conform to the protocol version specified by the MS-ASProtocolVersion header (section 2.2.1.1.2.4) in the client request.

The server SHOULD format a response to the request 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 with the URL of the correct server.

If the server returns an HTTP 503 error, it MAY<6> include a Retry-After header ([RFC2616]) in the response with an estimate of how many seconds will elapse before the server is expected to be able to process the request.

If the server requires the client to reinitialize its synchronization state, it SHOULD include an X-MS-RP header, MS-ASProtocolCommands header, and a MS-ASProtocolVersions header in its response to the client.

The server MAY<7> 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 2 changes within a 24-hour time period, but MAY<8> 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<9> block clients for a different amount of time.

If the server blocks a client for changing its User-Agent header value, it returns an error that depends on the value of the MS-ASProtocolVersion header (section 2.2.1.1.2.4). If the value of the MS-ASProtocolVersion header is 14.0 or 14.1, the server MUST return a **Status** element ([MS-ASCMD] section 2.2.3.162) with a value of **DeviceIsBlockedForThisUser** (129), as specified in [MS-ASCMD] section 2.2.4. If the value of the MS-ASProtocolVersion header is 12.1, the server must return an HTTP error 403.

3.2.5.2 Handling HTTP OPTIONS Command

The server response to the HTTP OPTIONS command request MUST contain the MS-ASProtocolCommands header set to a comma-delimited list of the supported ActiveSync commands, and a MS-ASProtocolVersions header set to a comma-delimited list of the supported ActiveSync protocol versions.

The latest supported version of the protocol is 14.1. Older supported versions include 14.0 and 12.1. Not all commands and functionality described in the ActiveSync protocol documentation are supported by the older protocol versions. See the Product Behavior section in each document to determine which commands and capabilities are not available in older protocol versions.

A protocol server can support multiple versions of this protocol. This specification and any protocol specifications that cite it as a dependency are applicable when the value of the MS-ASProtocolVersion header is set to 14.1, 14.0, or 12.1.<10>

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 released service packs:

- Microsoft® Exchange Server 2007 Service Pack 1 (SP1)
- Microsoft® Exchange Server 2010
- Microsoft® Exchange Server 2013
- Windows® Communication Apps

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

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

<1> Section 2.2.1.1.1.1: Exchange 2007 SP1 and the initial release version of Exchange 2010 do not set the **Protocol version** field to 141.

<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 2.2.1.1.2.2: Exchange 2007 SP1 accepts a Content-Type header value of either "text/xml" or "text/html" for the **Autodiscover** command.

<4> Section 3.1.3: Windows Communication Apps only support protocol versions 12.1 and 14.0.

<5> 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.

<6> Section 3.2.5.1: Exchange 2010 and Exchange 2013 sometimes include a Retry-After header with HTTP 503 error responses.

<7> Section 3.2.5.1: Exchange 2013 can be configured to track changes to the User-Agent header, but does not do so by default.

<8> Section 3.2.5.1.1: Exchange 2013 can be configured to use different values for the allowed number of changes and the time period.

<9> Section 3.2.5.1.1: Exchange 2013 can be configured to block clients for an amount of time other than 14 hours.

<10> Section 3.2.5.2: Exchange 2007 SP1 does not return MS-ASProtocolVersions values of 14.1 and 14.0. The initial release version of Exchange 2010 does not return the MS-ASProtocolVersions value of 14.1.

7 Change Tracking

This section identifies changes that were made to the [MS-ASHTTP] protocol document between the July 2012 and October 2012 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

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 or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- The removal of a document from the documentation set.
- Changes made for template compliance.

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 **Editorial** means that the language and formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

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

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.

- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- New content added for template compliance.
- Content updated for template compliance.
- Content removed for template compliance.
- Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated.**

Some important terms used in the change 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.

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
1.2.2 Informative References	Added the reference [MS-OXPROTO].	N	Content updated.
1.4 Relationship to Other Protocols	Added informative reference information for overview of relationships between this and other protocols.	N	Content updated.
2.2.1.1.1.2 Plain Text Query Value	Revised the ABNF rule for device-id.	N	Content updated.
3.1.3 Initialization	Added product behavior note to describe the versions of the protocol supported by Windows Communication Apps.	Y	New product behavior note added.
3.1.5.2.3 HTTP Error 503	Added product behavior note to describe how Windows Communication Apps retries requests in the presence of a Retry-After header.	Y	New product behavior note added.
3.2.5.1 Handling HTTP Requests	Added Exchange Server 2013 to the product behavior note.	Y	Product behavior note updated.

8 Index

V