

# [MS-ASHTTP]: ActiveSync HTTP Protocol Specification

---

## Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

## 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	6.0.1	Editorial	Changed language and formatting in the technical content.
11/03/2010	6.1	Minor	Clarified the meaning of the technical content.

# Contents

<b>1 Introduction</b>	<b>5</b>
1.1 Glossary	5
1.2 References	6
1.2.1 Normative References	6
1.2.2 Informative References	7
1.3 Overview	7
1.4 Relationship to Other Protocols	7
1.5 Prerequisites/Preconditions	8
1.6 Applicability Statement	8
1.7 Versioning and Capability Negotiation	8
1.8 Vendor-Extensible Fields	8
1.9 Standards Assignments	8
<b>2 Messages</b>	<b>9</b>
2.1 Transport	9
2.2 Message Syntax	9
2.2.1 Request	9
2.2.1.1 Request Format	9
2.2.1.1.1 Request Line	9
2.2.1.1.1.1 Base64-Encoded URI	10
2.2.1.1.1.2 Command-Specific URI Parameters	12
2.2.1.1.1.3 Command Codes	12
2.2.1.1.1.4 Command Parameters	13
2.2.1.1.1.5 Device ID	14
2.2.1.1.1.6 Device Types	14
2.2.1.1.2 Request Headers	14
2.2.1.1.2.1 Authorization	14
2.2.1.1.2.2 Content-Type	15
2.2.1.1.2.3 HTTP OPTIONS	15
2.2.1.1.3 Request Body	15
2.2.2 Response	15
2.2.2.1 Response Format	15
2.2.2.1.1 Status Line	15
2.2.2.1.1.1 HTTP Status Code Categories	16
2.2.2.1.1.2 Troubleshooting Tip	17
2.2.2.1.1.3 Troubleshooting HTTP Error 503	17
2.2.2.1.1.4 Troubleshooting HTTP Error 451	18
2.2.2.1.2 Response Headers	18
2.2.2.1.2.1 MS-ASProtocolVersions	19
2.2.2.1.3 Response Body	19
2.2.3 Namespace	19
2.2.4 Simple Types	19
2.2.5 Complex Types	19
2.2.6 Elements	20
2.2.7 Attributes	20
2.2.8 Groups	20
2.2.9 Attributes Groups	20
2.2.10 Messages	20
<b>3 Protocol Details</b>	<b>21</b>

3.1	Common Details .....	21
3.1.1	Abstract Data Model .....	21
3.1.2	Timers .....	21
3.1.3	Initialization .....	21
3.1.4	Higher Layer Triggered Events .....	21
3.1.5	Message Processing Events and Sequencing Rules .....	21
3.1.6	Timer Events .....	21
3.1.7	Other Local Events .....	21
<b>4</b>	<b>Protocol Examples .....</b>	<b>22</b>
4.1	FolderSync Request and Response .....	22
4.2	HTTP OPTIONS Command Request and Response .....	22
4.3	SendMail Request and Response .....	23
4.4	CreateFolder Request and Response .....	24
<b>5</b>	<b>Security .....</b>	<b>25</b>
5.1	Security Considerations for Implementers .....	25
5.2	Index of Security Parameters .....	25
<b>6</b>	<b>Appendix A: Product Behavior .....</b>	<b>26</b>
<b>7</b>	<b>Change Tracking .....</b>	<b>27</b>
<b>8</b>	<b>Index .....</b>	<b>29</b>

# 1 Introduction

All communication between the client and server is initiated by the client and is based on request/response commands. When the client communicates with the server, the client sends a request to the server as an **HTTP POST**, by using UTF-8 encoding. The server sends back a response to the **HTTP POST**. The request and response each have a **header** and might have a body. The header 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.

In addition to the **HTTP POST** request/response commands, the **HTTP OPTIONS** command specifies the command to be implemented.

The ActiveSync protocol is designed to enable a client device to synchronize data with the data that is stored on the server.

## 1.1 Glossary

The following terms are defined in [\[MS-OXGLOS\]](#):

- alias**
- ASCII**
- attachment**
- base64 encoding**
- calendar**
- collection**
- contact**
- folder**
- Global Address List**
- GUID**
- header**
- Hypertext Transfer Protocol (HTTP)**
- Inbox folder**
- mailbox**
- meeting request**
- message**
- MIME**
- locale**
- plain text**
- Out of Office (OOO)**
- property (1)**
- recipient**
- Secure Sockets Layer (SSL)**
- Sent Items folder**
- S/MIME**
- Synchronization**
- Uniform Resource Identifier (URI)**
- Uniform Resource Locator (URL)**
- Wireless Application Protocol (WAP) Binary XML (WBXML)**
- Web server**
- XML**

## XML namespace XML Schema Definition (XSD)

The following terms are specific to this document:

**server ID:** A unique identifier assigned by the server to each object that can be synchronized. The 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

### 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](mailto: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-ASAIRS] Microsoft Corporation, "[ActiveSync AirSyncBase Namespace Protocol Specification](#)", December 2008.

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

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

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

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

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

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

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

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

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

[MS-LCID] Microsoft Corporation, "Windows Language Code Identifier (LCID) Reference", March 2007, <http://msdn.microsoft.com/en-us/library/cc233965.aspx>

[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.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.ietf.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", 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, <ftp://ftp.rfc-editor.org/in-notes/rfc4985.txt>

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

### 1.2.2 Informative References

[MS-OXGLOS] Microsoft Corporation, "[Exchange Server Protocols Master Glossary](#)", April 2008.

[MSDN-APM] Marquardt, Thomas, "ASP.NET Performance Monitoring, and When to Alert Administrators", July 2003, <http://msdn.microsoft.com/en-us/library/ms972959.aspx>

### 1.3 Overview

This specification describes the ActiveSync protocol, which is used to write software that will synchronize server data with a client mobile device. The protocol relies on a client/server architecture. The server is a **Web server** and is running the ActiveSync **synchronization** server software. The client is the mobile device. 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.

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. 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.

### 1.4 Relationship to Other Protocols

The ActiveSync 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 protocols specified in [\[MS-ASAIRS\]](#), [\[MS-ASCAL\]](#), [\[MS-ASCMD\]](#), [\[MS-ASCNTC\]](#), [\[MS-ASDOC\]](#), [\[MS-ASDTYPE\]](#), [\[MS-ASEMAIL\]](#), [\[MS-ASPROV\]](#), [\[MS-ASTASK\]](#), and [\[MS-ASWBXML\]](#) use the ActiveSync protocol.

## 1.5 Prerequisites/Preconditions

When using an HTTP connection secured by the **Secure Sockets Layer (SSL)** as transport, 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 employs the ActiveSync protocol to communicate calendar, contact, email, task, note, and other data between a mail server and a mobile device.

An ActiveSync protocol server can support multiple versions of the ActiveSync protocol. This document 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. [<1>](#) Any deviations in behavior between 14.1 and 14.0, or 14.1 and 12.1 are specified in the product behavior appendices of this document and any dependent protocol specifications. Dependent protocol specifications are specified in section [1.4](#).

## 1.7 Versioning and Capability Negotiation

The **HTTP OPTIONS** command is used by the client to discover which versions of the ActiveSync protocol are supported by the server. See section [2.2.1.1.2.3](#) for a discussion of the **HTTP OPTIONS** command and section [4.2](#) for an example of an **HTTP OPTIONS** request and response. The latest supported version of the protocol is 14.0. Older supported versions include 2.0, 2.1, 2.5, 12.0, 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.

## 1.8 Vendor-Extensible Fields

None

## 1.9 Standards Assignments

None

## 2 Messages

### 2.1 Transport

Messages are transported by using **HTTP POST**, as specified in [\[RFC2616\]](#). The query parameters in the request **URI** MAY be **base64 encoded** (see section [2.2.1.1.1](#)). The body of the HTTP message contains the **XML** 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 or the Response Body is transmitted between client and server by using Wireless Application Protocol (WAP) Binary XML (WBXML), as specified in [\[WBXML1.2\]](#).

The following are the three general types of messages:

- Request
- Response
- **HTTP OPTIONS**

#### 2.2.1 Request

The client creates a message 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 HTTP, 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 consists of a scheme indicator and the host name, which are optional, 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 base64-encoded byte sequence.

base64 encoding is optional. For more details about base64 encoding, see [\[RFC2045\]](#).

The format of the URI 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  
Content-Type: application/vnd.ms-sync.wbxml  
MS-ASProtocolVersion: 14.0  
User-Agent: ASOM  
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  
Content-Length: 866
```

For more details about base64 encoding URIs, see section [2.2.1.1.1.1](#).

### 2.2.1.1.1.1 Base64-Encoded URI

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

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

The base64-encoded sequence of bytes represents the URI request parameters. The following table provides the details of the base64-encoded sequence of bytes. This byte sequence is divided into fields in the same order as the following table.

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																								
...																																		
...																																		
...																																		
...										Policy key length										Policy key														
...																		Device type length						Device type										
...																																		
...																																		
...																														Command parameters (variable)				
...																																		

Size	Field	Description
1 byte	Protocol version	An integer that specifies the version of the ActiveSync protocol that is being used. This value MUST be 140. <2>
1 byte	Command code	An integer that specifies the command (see table of command codes in section 2.2.1.1.1.3).
2 bytes	Locale	An integer that specifies the <b>locale</b> of the language that is used for the response. Locale integer values are specified in [MS-LCID].
1 byte	Device ID length	An integer that specifies the length of the device ID. A value of 0 indicates that the device ID field is absent.
0 - 16 bytes	Device ID	A string or a <b>GUID</b> that identifies the device. <3>
1 byte	Policy key length	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.
0 or 4 bytes	Policy key	An integer that indicates the state of policy settings on the client device.
1 byte	Device type length	An integer that specifies the length of the device type value.
0 - 16 bytes	Device type	A string that specifies the type of client device. For details, see section 2.2.1.1.1.6.
Variable	Command parameters	A set of parameters that varies depending on the command. Each parameter consists of a tag, length, and value field in that order. For details, see section 2.2.1.1.1.4.

### 2.2.1.1.1.2 Command-Specific URI Parameters

The following URI parameters, also called command parameters, are specific to the ActiveSync commands. For more details about specific commands, see [\[MS-ASCMD\]](#).

Parameter	Description	Used by
AttachmentName	A string that specifies the name of the <b>attachment</b> file to be retrieved.	GetAttachment<4>
CollectionId	A string that specifies the <b>server ID</b> of the <b>collection</b> that contains the message to be forwarded or replied to.	SmartForward, SmartReply
ItemId	A string that specifies the server ID of the message item 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 <b>meeting</b> .	SmartForward, SmartReply
SaveInSent	A character that specifies whether a copy of the message will be saved in the <b>Sent Items folder</b> . Set this parameter to T to instruct the server to save the message 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

### 2.2.1.1.1.3 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 URI to specify the command. For more details, see [\[MS-ASCMD\]](#).

Code	Command	Description
0	Sync	Synchronizes changes in a collection between the client and the server.
1	SendMail	Sends mail to the server. This command is issued in the <b>HTTP POST</b> command's Uniform Resource Identifier (URI), and does not contain an XML body. The body will instead contain the <b>MIME</b> -formatted message.
2	SmartForward	Forwards messages without retrieving the full message from the server.
3	SmartReply	Replies to messages without retrieving the full message from the server.
4	GetAttachment	Retrieves an e-mail attachment from the server.<5>
9	FolderSync	Synchronizes the collection hierarchy but does not synchronize the items in the collections.
10	FolderCreate	Creates an e-mail, <b>calendar</b> , or <b>contacts folder</b> 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.

Code	Command	Description
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 <b>meeting request</b> in the user's <b>Inbox</b> .
16	Search	Finds and retrieves information about contacts and <b>recipients</b> in the <b>Global Address List</b> .
17	Settings	Supports getting and setting global <b>properties</b> , such as <b>Out of Office (OOO)</b> 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 <b>S/MIME</b> certificates so that clients can send encrypted S/MIME e-mail messages.
22	ValidateCert	Validates a certificate that has been received through an S/MIME mail.

#### 2.2.1.1.1.4 Command Parameters

The following table lists the fields of a command parameter.

Size	Field	Description
1 byte	Tag	An integer that specifies the name of the parameter.
1 byte	Length of value	An integer that specifies the length of the parameter value, from 0 to 255 characters.
Variable	Value	The value of the parameter.

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](#).

Tag	Parameter Name
0	AttachmentName
3	ItemId
4	LongId
6	Occurrence
7	Options

Tag	Parameter Name
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. The following bits are valid: 0x01: SaveInSent 0x02: AcceptMultiPart	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

#### 2.2.1.1.1.5 Device ID

A string that specifies the device. Each device **MUST** have a unique **DeviceId** string. Each request from the device **MUST** include the same **DeviceId** string.

#### 2.2.1.1.1.6 Device Types

Any string that specifies a device type can be valid. "SP" specifies a SmartPhone and "PPC" specifies a PocketPC. Other devices send unique strings for their specific device type. Each request from the device **MUST** include the same **DeviceType** string.

#### 2.2.1.1.2 Request Headers

The HTTP/1.1 protocol 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 **MUST** be supplied for ActiveSync synchronization protocol requests. Note that requests are UTF-8 encoded.

Header	Required	Value	Notes
Authorization	Yes	<b>String</b> ( <a href="#">[MS-ASDTYPE]</a> section 2.6)	Specifies that user credentials are sent by using HTTP basic authentication. For details, see section <a href="#">2.2.1.1.2.1</a> .
Content-Type	Depends on the command.	<b>String</b> (Depends on the command. For details about individual commands, see <a href="#">[MS-ASCMD]</a> section 2.2.2.)	Specifies that the media type of the request body is WBXML. For more details, see <a href="#">[MS-ASWBXML]</a> . Other types of content, such as <a href="#">[RFC2822]</a> , can also be specified, depending on the command.

#### 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 base64 encoded. Secure Sockets Layer (SSL) **SHOULD** be enabled between the client and the server whenever this header is sent. For user *fakename* and password *x\$pIAK9@p9!*, the Authorization header would be as follows.

Authorization: Basic ZmFrZXVzZXI6eCRwSUFL0UBwOSE=

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 **MUST** be set to text/xml.<6>

### 2.2.1.1.2.3 HTTP OPTIONS

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 can use the **HTTP OPTIONS** command to determine whether the server supports the same versions of the protocol that the client supports. If the client and server both support multiple versions of the protocol, the client **SHOULD** use the most recent version (the greatest numbered version) of the protocol that is available on the server.

The argument supplied to the **HTTP OPTIONS** command in a client request **MUST** be "/Microsoft-Server-ActiveSync".

For an example of the **HTTP OPTIONS** command, see section [4.2](#).

### 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 Response

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

### 2.2.2.1 Response Format

Each command is sent from the server to the client as in the format of HTTP/1.1. Note that these responses are UTF-8 encoded. As specified by HTTP, the format is the same as for 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:

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 <a href="#">2.2.2.1.1.2</a> .
403 Forbidden	The user is not enabled for ActiveSync synchronization. For details about how to troubleshoot this error, see section <a href="#">2.2.2.1.1.2</a> .
404 Not Found	The specified URI could not be found or the server is not a valid server with ActiveSync.
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 <a href="#">2.2.2.1.1.2</a> .
501 Not Implemented	The server does not support the functionality that is required to fulfill the request. This status code MAY 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 <a href="#">2.2.2.1.1.3</a> .
507 Insufficient Disk Space	The user's mailbox is full.

The following table lists another possible HTTP status code.

Status code	Description
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. The device MUST redirect all future requests to the server specified by the X-MS-Location header in the response. For all other status codes (401, 403, 500), and if the MS-X-Location header <b>URL</b> value is not specified with the 451 status code, follow the full <b>Autodiscover</b> command process, as specified in <a href="#">[MS-ASCMD]</a> . For more details about how to troubleshoot this error, see section <a href="#">2.2.2.1.1.4</a> .

### 2.2.2.1.1.1 HTTP Status Code Categories

The following table lists the categories for the HTTP status codes.

Status code range	Status type
100 – 199	Informational - Request received, processing is continuing.
200 – 299	Success - The action was received, understood, and accepted.
300 – 399	Redirection - Further action <b>MUST</b> be taken to complete the request.
400 – 499	Client error - The request contains incorrect syntax or cannot be fulfilled by the server due to an issue with the client request.
500 – 599	Server error - The server has erred or is otherwise incapable of fulfilling the request.

For more details about HTTP status codes, see [\[RFC2616\]](#) section 10.

### 2.2.2.1.1.2 Troubleshooting Tip

The client **SHOULD** send an **Autodiscover** command request to the server if any of the following HTTP errors are encountered:

- 401
- 403
- 500

If the user's **mailbox** has moved, if the client is trying to connect to 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 the server **MUST** respond to any command request with error code 451 and include the URL of a preferred server in the exception message. The device **MUST** redirect all future requests to the preferred server by using the URL that was provided. For all other status codes (401, 403, 500), or if the URL is not specified with the 451 status code, then the full **Autodiscover** command process is followed, as specified in [\[MS-ASCMD\]](#).

The **HTTP OPTIONS** command is the first command that is sent to the preferred server.

### 2.2.2.1.1.3 Troubleshooting HTTP Error 503

One of the causes of HTTP error 503 is that more users than 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

```

The client can retry the request that generated the HTTP error 503 response 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 [\[MSDN-APM\]](#).

#### 2.2.2.1.1.4 Troubleshooting HTTP Error 451

If the user 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. If the wrong server has the URL for a preferred server, then the wrong server provides that URL in its response back to the client, using the X-MS-Location header. If the wrong 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 [\[MS-ASCMD\]](#).

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.exchange.microsoft.com/Microsoft-Server-ActiveSync
Cache-Control: private
Content-Length: 0
```

All subsequent requests to the X-MS-Location SHOULD use the URL specified within the X-MS-Location header of the HTTP error 451 response.

#### 2.2.2.1.2 Response Headers

The ActiveSync protocols 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
Content-Length	56	Required. Specifies the size of the response body in bytes.
Content-Encoding	gzip	Required when the content is base64-encoded; otherwise, this header is not included. Specifies the HTTP compression format that is used in the response.
Cache-Control	private	Controls how the response is cached.
Content-Type	application/vnd.ms-sync.wbxml	Required. Specifies that the media-type of the request body is WBXML. Other types of content, such as <a href="#">[RFC2822]</a> , can also be specified.
Date	Tue, 17 Mar 2009 21:44:09 GMT	Optional. Indicates the date and time at which the response message originated.
Server	Microsoft-IIS/7.0	Optional. Indicates the software that was used by the server to handle the request.

Header	Example value	Notes
MS-Server-ActiveSync	8.1	Optional. Indicates the version of the ActiveSync server that was used to handle the request.
X-MS-RP	2.0,2.1,2.5,12.0,12.1,14.0	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 MS-ASProtocolVersions

The MS-ASProtocolVersions header indicates the version of the ActiveSync protocol supported by the client. Server responses to client command requests **MUST** conform to the same version specified by the client request.

Client applications can use the **HTTP OPTIONS** command as specified in section [2.2.1.1.2.3](#) to receive a list of ActiveSync protocol versions supported by the server. The server response to this request contains the header MS-ASProtocolVersions, whose value is a comma-separated list of the supported ActiveSync protocol versions.

The values of the MS-ASProtocolVersions header covered by this protocol specification are specified in section [1.6](#).

### 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 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 Namespace

This specification defines and references various **XML namespaces** by using the mechanisms specified in [\[MS-ASAIRS\]](#). Although this specification associates a specific XML namespace prefix for each XML namespace that is used, the choice of any particular XML namespace prefix is implementation-specific and not significant for interoperability.

### 2.2.4 Simple Types

None.

### 2.2.5 Complex Types

None.

## **2.2.6 Elements**

None.

## **2.2.7 Attributes**

None.

## **2.2.8 Groups**

None.

## **2.2.9 Attributes Groups**

None.

## **2.2.10 Messages**

None.

## **3 Protocol Details**

### **3.1 Common Details**

#### **3.1.1 Abstract Data Model**

None.

#### **3.1.2 Timers**

None.

#### **3.1.3 Initialization**

None.

#### **3.1.4 Higher Layer Triggered Events**

None.

#### **3.1.5 Message Processing Events and Sequencing Rules**

None.

#### **3.1.6 Timer Events**

None.

#### **3.1.7 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 Uniform Resource Identifier (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 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.<7> 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,ValidateCertPublic: 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
```

## 4.3 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
```

## 4.4 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:

- Microsoft® Exchange Server 2007 Service Pack 1 (SP1)
- Microsoft® Exchange Server 2010
- Microsoft® Exchange Server 2010 Service Pack 1 (SP1)

Exceptions, if any, are noted below. If a service pack number appears with the product version, behavior changed in that service pack. The new behavior also applies to subsequent service packs of the product unless otherwise specified. 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 prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that product does not follow the prescription.

[<1> Section 1.6:](#) Exchange 2007 does not support MS-ASProtocolVersions values of 14.1 and 14.0. The initial version of Exchange 2010 does not support the MS-ASProtocolVersions value of 14.1. However, Exchange 2010 SP1 supports the MS-ASProtocolVersions value of 14.1.

[<2> Section 2.2.1.1.1.1:](#) For Exchange 2003 and Exchange 2007, the other allowed values are 20, 21, 25, 120, and 121. When using these older versions, not all the current ActiveSync commands and functionality are available. The Product Behavior section in each ActiveSync protocol document defines the commands and functionality that are supported by older versions.

[<3> Section 2.2.1.1.1.1:](#) Windows Mobile devices use a GUID.

[<4> Section 2.2.1.1.1.2:](#) The GetAttachment command is only supported when the MS-ASProtocolVersion header is set to 12.1 or 12.0 in the GetAttachment command request. The GetAttachment command is not supported for MS-ASProtocolVersion 14.0 – use the Fetch element of the ItemOperations command instead.

[<5> Section 2.2.1.1.1.3:](#) The GetAttachment command is only supported when the MS-ASProtocolVersion header is set to 12.1 or 12.0 in the GetAttachment command request. The GetAttachment command is not supported for MS-ASProtocolVersion 14.0 – use the Fetch element of the ItemOperations command instead.

[<6> Section 2.2.1.1.2.2:](#) Exchange 2007 accepts a Content-Type header of either text/xml or text/html for the Autodiscover command.

[<7> Section 4.2:](#) The GetHierarchy, CreateCollection, DeleteCollection, and MoveCollection commands are not supported when the MS-ASProtocolVersions HTTP header is set to 14.0.

## 7 Change Tracking

This section identifies changes that were made to the [MS-ASHTTP] protocol document between the August 2010 and November 2010 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.
- Changes made for template compliance.
- Removal of a document from the documentation set.

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 revision type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

The changes made to this document are listed in the following table. For more information, please contact [protocol@microsoft.com](mailto:protocol@microsoft.com).

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change Type
<a href="#">2.2.2.1.1.1 HTTP Status Code Categories</a>	Added section reference for RFC2616.	N	Editorially updated.
<a href="#">2.2.2.1.1.1 HTTP Status Code Categories</a>	Removed term "table".	N	Editorially updated.
<a href="#">6 Appendix A: Product Behavior</a>	58901 Changed Exchange 2007 SP3 to Exchange 2007 SP1.	N	Content updated.

## 8 Index

### C

[Versioning](#) 8

[Capability negotiation](#) 8

[Change tracking](#) 27

### G

[Glossary](#) 5

### I

[Implementer - security considerations](#) 25

[Informative references](#) 7

[Introduction](#) 5

### M

Messages

[overview](#) 9

[syntax](#) 9

[transport](#) 9

### N

[Normative references](#) 6

### O

[Overview \(synopsis\)](#) 7

### P

[Preconditions](#) 8

[Prerequisites](#) 8

[Product behavior](#) 26

### R

References

[informative](#) 7

[normative](#) 6

[Relationship to other protocols](#) 7

### S

Security

[implementer considerations](#) 25

[overview](#) 25

Syntax

[messages - overview](#) 9

### T

[Tracking changes](#) 27

[Transport](#) 9

### V