[MS-ASMS]:
Exchange ActiveSync:
Short Message Service (SMS) Protocol

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 Open Specification Promise or the Community Promise. If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.

- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.

- **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
04/10/2009	0.1.0	Major	Initial Availability.
07/15/2009	1.0.0	Major	Revised and edited for technical content.
11/04/2009	2.0.0	Major	Updated and revised the technical content.
02/10/2010	3.0.0	Major	Updated and revised the technical content.
05/05/2010	4.0.0	Major	Updated and revised the technical content.
08/04/2010	5.0	Major	Significantly changed the technical content.
11/03/2010	5.0	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	5.1	Minor	Clarified the meaning of the technical content.
08/05/2011	6.0	Major	Significantly changed the technical content.
10/07/2011	6.0	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	7.0	Major	Significantly changed the technical content.
04/27/2012	7.0	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	7.0	No change	No changes to the meaning, language, or formatting of the technical content.
10/08/2012	8.0	Major	Significantly changed the technical content.
02/11/2013	8.0	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

The Exchange ActiveSync: Short Message Service (SMS) Protocol describes an **XML**-based format that provides the mechanisms for a mobile device to synchronize **SMS** messages with the server and for the server to send SMS messages through the mobile device.

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]:

> **code page**
> **XML**

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

> **conversation**
> **conversation ID**
> **conversation index**
> **Inbox folder**
> **non-delivery report**
> **Outbox folder**
> **Sent Items folder**
> **Short Message Service (SMS)**
> **Wireless Application Protocol (WAP) Binary XML (WBXML)**
> **XML namespace**
> **XML schema**
> **XML schema definition (XSD)**

The following terms are specific to this document:

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

1.2 References

References to Microsoft Open Specifications documentation do not include a publishing year because links are to the latest version of the 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-ASAIRS] Microsoft Corporation, "Exchange ActiveSync: AirSyncBase Namespace Protocol".

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

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

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

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

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

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

[XML] World Wide Web Consortium, "Extensible Markup Language (XML) 1.0 (Fourth Edition)", W3C Recommendation, August 2006, http://www.w3.org/TR/2006/REC-xml-20060816/

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

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

1.2.2   Informative References

[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 an XML-based format that is used to do the following:

- Enable a mobile device to synchronize SMS messages with the server.

- Provision the server to send outgoing SMS messages through the mobile device.

This protocol also includes XML elements to represent SMS message data. The SMS data is included in protocol command requests when SMS data is being sent from the client to the server, and is included in protocol command responses when SMS data is retrieved from the server. SMS data includes some of the same header information as e-mail data such as to and from, as well as body, flag, and importance.

1.4   Relationship to Other Protocols

This protocol consists of a series of XML elements that are embedded inside a command request or a command response. For information about command requests and responses, see [MS-ASCMD]. The **Wireless Application Protocol (WAP) Binary XML (WBXML)**, described in [MS-ASWBXML], is used to transmit the XML markup that constitutes the request body and the response body.

This protocol defines elements according to the data type definitions that are described in [MS-ASDTYPE].

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

1.5 Prerequisites/Preconditions

None.

1.6 Applicability Statement

This protocol is applicable for synchronizing SMS messages from mobile devices to the server and relaying outbound SMS messages from the server to the mobile device.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

This protocol consists of a series of XML elements that are embedded inside a command request or a command response. The XML markup that constitutes the request body or the response body is transmitted between client and server by using WBXML, as specified in [MS-ASWBXML].

The mobile device uses standard mobile network protocols, such as GSM and CDMA, to send outbound SMS messages.

2.2 Message Syntax

The XML markup that is used by this protocol MUST be well formed XML, as specified in [XML]. Any violation of the well formed XML MUST be reported by the XML processor, as specified in [XML] section 5. The XML elements are embedded inside a command request or command response. For details about command requests and responses, see [MS-ASCMD].

This protocol defines **XML schema** elements for the **SMS** class, which represents an SMS message. The **SMS** class is constituted by elements from the following three **XML namespaces**: Email, Email2, and AirSyncBase. The **XML schema definitions (XSDs)** for these three namespaces, as they apply to the **SMS** class, are provided in this section (following this paragraph). This protocol also defines XML schema elements that are necessary for various operations involving SMS messages. These elements are defined in the AirSync and Settings XML namespaces.

XSDs for the SMS class

The portion of the Email namespace constituting the **SMS** class is defined as follows. For the complete Email XSD, see [MS-ASEMAIL] section 2.2.

```xml
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:tns="Email:" xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="Email:" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import namespace="AirSyncBase:" schemaLocation="AirSyncBase.xsd"/>
  <xs:element name="To">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="32768"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="From">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:maxLength value="32768"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="DateReceived" type="xs:dateTime"/>
  <xs:element name="Importance" type="xs:unsignedByte"/>
  <xs:element name="Read" type="xs:boolean"/>
  <xs:element name="InternetCPID" type="xs:string"/>
  <xs:element name="Flag"/>
</xs:schema>
```

The portion of the Email2 namespace constituting the **SMS** class is defined as follows. For the complete Email2 XSD, see [MS-ASEMAIL] section 2.2.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:tns="Email2:" xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="Email2:" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:element name="ConversationId" type="xs:string"/>
  <xs:element name="ConversationIndex" type="xs:string"/>
</xs:schema>
```

The portion of the AirSyncBase namespace constituting the **SMS** class is defined as follows. For the complete AirSyncBase XSD, see [MS-ASAIRS] section 2.2.

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

2.2.1 Namespaces

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

Prefix	Namespace URI	Reference
airsync	AirSync	[MS-ASCMD] section 2.2.2.19
airsycnbase	AirSyncBase	[MS-ASAIRS]
email	Email	[MS-ASEMAIL]
email2	Email2	[MS-ASEMAIL]
settings	Settings	[MS-ASCMD] section 2.2.2.16
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1]

2.2.2 Elements

This protocol defines XML schema elements for the **SMS** class, which represents an SMS message. The XSDs for the **SMS** class are specified in section 2.2. The **SMS** class elements are specified in section 2.2.2.1.

This protocol also defines XML schema elements that are necessary for operations involving SMS messages. The following table summarizes the set of common XML schema elements that are defined by this specification for use in the **GetItemEstimate** command, **Sync** command, and **Settings** command.

Element name	Description
airsync:Class (for options)	Used in a **GetItemEstimate** command request or a **Sync** command request to specify the class to which the options apply. See sections 2.2.2.2.1.1 and 2.2.2.2.1.2.
airsync:Class (for adding, changing, deleting)	Used in a **Sync** command to specify the class of the item that is being added, changed, or deleted.<1> See section 2.2.2.2.1.2.
settings:EnableOutboundSMS	Used in a **Settings** command request to enable or disable the sending of outbound SMS messages through a mobile device. See section 2.2.2.2.2.
settings:PhoneNumber	Used in a **Settings** command request to specify the telephone number that identifies the mobile device. See section 2.2.2.2.3.

2.2.2.1 SMS Class Elements

This section specifies elements that constitute the **SMS** class. The **SMS** class represents an SMS message. The **SMS** class elements are children of the **airsync:ApplicationData** element in the **Sync** command.

The XSDs of the **SMS** class are specified in section 2.2. For details about the **airsync:ApplicationData** element, see [MS-ASCMD] section 2.2.3.11.

2.2.2.1.1 Body

The **airsyncbase:Body** element (as specified in [MS-ASAIRS] section 2.2.2.4) is an optional child element of the **SMS** class (represented by the **airsync:ApplicationData** element). It contains details about the body of the SMS message. This element is part of the AirSyncBase XML namespace, as specified in section 2.2 of this document.

2.2.2.1.2 ConversationId

The **email2:ConversationId** element (as specified in [MS-ASCON] section 2.2.2.3.3) is a required child element of the **SMS** class (represented by the **airsync:ApplicationData** element) in **Sync** command responses. It specifies a unique identifier for a **conversation**.

2.2.2.1.3 ConversationIndex

The **email2:ConversationIndex** element (as specified in [MS-ASCON] section 2.2.2.4) is a required child element of the **SMS** class (represented by the **airsync:ApplicationData** element) in **Sync** command responses. It specifies a set of timestamps that is used by a client to generate a tree view of a conversation.

2.2.2.1.4 DateReceived

The **email:DateReceived** element is an optional child element of the **SMS** class (represented by the **airsync:ApplicationData** element) in **Sync** command requests and responses. The **DateReceived** element specifies the date and time when the SMS message was either received by the device or sent by the device. If the SMS message is in the **Sent Items folder**, this element specifies the date and time when the SMS message was sent; otherwise, this element specifies the date and time when the SMS message was received.

The value of this element is a **dateTime** data type, as specified in [MS-ASDTYPE] section 2.3.

2.2.2.1.5 Flag

The **email:Flag** element (as specified in [MS-ASEMAIL] section 2.2.2.27) is an optional child element of the **SMS** class (represented by the **airsync:ApplicationData** element). It specifies the flag that is associated with the SMS message, along with the current status of the SMS message.

2.2.2.1.6 From

The **email:From** element is an optional child element of the **SMS** class (represented by the **airsync:ApplicationData** element) in **Sync** command requests and responses. It specifies the telephone number of the individual who sent the SMS message.

The value of this element is a **string** data type, as specified in [MS-ASDTYPE] section 2.6. The format of the string is as follows, including the quotes and square brackets:

"*Sender's Name*" [MOBILE:*Sender's phone number*]

Sender's Name specifies the name of the sender and is optional. *Sender's phone number* specifies the mobile telephone number of the sender.

2.2.2.1.7 Importance

The **email:Importance** element (as specified in [MS-ASEMAIL] section 2.2.2.31) is an optional child element of the **SMS** class (represented by the **airsync:ApplicationData** element). It specifies the importance of the SMS message, as determined by the sender.

2.2.2.1.8 InternetCPID

The **email:InternetCPID** element (as specified in [MS-ASEMAIL] section 2.2.2.33) is a required child element of the **SMS** class (represented by the **airsync:ApplicationData** element). It specifies the **code page** ID of the SMS message.

2.2.2.1.9 Read

The **email:Read** element (as specified in [MS-ASEMAIL] section 2.2.2.47) is an optional child element of the **SMS** class (represented by the **airsync:ApplicationData** element). It specifies whether the SMS message has been viewed by the current recipient.

2.2.2.1.10 To

The **email:To** element is an optional child element of the **SMS** class (represented by the **airsync:ApplicationData** element) in **Sync** command requests and responses. It specifies the list of primary recipients.

The value of this element is a **string** data type, as specified in [MS-ASDTYPE] section 2.6. The value comprises the name and phone number of one or more recipients. Multiple recipients are separated by commas. The format of each recipient-entry is as follows, including the quotes and square brackets:

"*Recipient's Name*" [MOBILE: *Recipient's phone number*]

Recipient's Name specifies the name of the recipient and is optional. *Recipient's phone number* specifies the mobile telephone number of the recipient.

2.2.2.2 Additional Elements

This section specifies elements that are used in ActiveSync commands involving SMS items.

2.2.2.2.1 Class

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

- **GetItemEstimate** command requests

- **Sync** command requests

- **Sync** command responses

The definition of the **airsync:Class** element differs according to the context in which it is used. For more details, see section 2.2.2.2.1.1 and section 2.2.2.2.1.2.

2.2.2.2.1.1 Class (GetItemEstimate)

The **airsync:Class** element is an optional child element of the **airsync:Options** element in the **GetItemEstimate** command request. It specifies the class to which the options apply.

The value of this element is a **string**, as specified in [MS-ASDTYPE] section 2.6. For SMS messages, the value of this element is "SMS".

2.2.2.2.1.2 Class (Sync)

For options: The **airsync:Class** element is an optional child element of the **airsync:Options** element in the **Sync** command request. It specifies the class to which the options apply.

For adding, changing, and deleting: The **airsync:Class** element is an optional child element of the **airsync:Add**, **airsync:Change**, and **airsync:Delete**<2> elements in the **Sync** command response and an optional child element of the **airsync:Add** element in the **Sync** command request. The **airsync:Class** element specifies the class of the item that is being added, changed, or deleted.

The value of this element is a **string**, as specified in [MS-ASDTYPE] section 2.6. For SMS messages, the value is "SMS".

2.2.2.2.2 EnableOutboundSMS

The **settings:EnableOutboundSMS** element is an optional<3> child element of the **settings:Set** element in the **Settings** command request. It is used to provision the server for sending outbound SMS messages through the mobile device. (Outbound SMS messages are sent only through mobile devices that enable it.)

The value of this element is an integer, as specified in [MS-ASDTYPE] section 2.5. If this element is set to 1, then the mobile device can be used to send outbound SMS messages. If this element is set to 0, then the mobile device will not be used to send outbound SMS messages. Devices that were previously enabled, but no longer want to act as SMS transport agents for the server, can reset the property to zero on the server by sending a **Settings** command request with the **settings:EnableOutboundSMS** element set to 0. The default value of this element is 0.<4>

2.2.2.2.3   PhoneNumber

The **settings:PhoneNumber** element is an optional child element of the **settings:Set** element in the **Settings** command request.<5> It specifies the telephone number that identifies the mobile device.

The value of this element is a **string**, as specified in [MS-ASDTYPE] section 2.6.

3   Protocol Details

3.1   Client 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

3.1.4.1   Synchronizing SMS Items

A client initiates synchronization of data with the server by sending a **Sync** command request to the server. The **Sync** command request synchronizes the client's data with the data currently stored by the server. Synchronization of SMS items is allowed on any collection that contains e-mail messages.<6>

The **Sync** command request is further specified in [MS-ASCMD] section 2.2.2.19.1.

3.1.4.1.1   Options

If the client wants to synchronize SMS messages for a given folder, the **airsync:Options** block of the **Sync** command request includes the **airsync:Class** element set to the value "SMS". If the **airsync:Class** element is not included in the **airsync:Options** block, that set of options applies to the default class of the given folder, in which case, only the items of the default class for the given folder are synchronized. If no **airsync:Options** block is included in the **Sync** command request, only items of the default class for the given folder are synchronized.

A maximum of two **airsync:Options** blocks are allowed within a **airsync:Collection** block, namely one for the default class of the given folder and one for SMS messages. The **airsync:Class** element MUST be set to "SMS" to indicate that the options apply to SMS messages. An SMS **airsync:Options** block can be included within any **airsync:Collection** block that specifies an e-mail folder.

For more details about the **airsync:Class** element as a child of the **airsync:Options** element, see section 2.2.2.2.1.2.

3.1.4.1.1.1   Sticky Options

Sticky options allow a client to send a **Sync** command request that does not include an **airsync:Options** block. In this case, the server uses the previous set of options.

Sticky options require that the client send all sets of options if any option has changed. For example, if the truncation limit has increased for SMS items and the client is synchronizing both SMS items and e-mail items for a given folder, the **Sync** request will include two **airsync:Options** blocks. If one set of options is omitted from the request, the server presumes that the item class that

corresponds to the missing set of options is no longer desired and soft-deletes are issued for the items of that class. (The server will send a soft-delete for any item of a class that is not included in the **Sync** request.) The server issues a soft-delete by including a **airsync:SoftDelete** element in the **Sync** response. For more details about the **airsync:SoftDelete** element and soft-deletes, see [MS-ASCMD] section 2.2.3.157.

3.1.4.1.1.2 Filtering

A filter can be applied to the synchronization of SMS items. The **airsync:FilterType** element specifies a time window. When synchronizing SMS items, the time-window restricts the items that the server sends to the client. The same filter values (0 through 5) that are used to filter e-mail items are also used to filter SMS items. For more details about filtering and the **airsync:FilterType** element, see [MS-ASCMD] section 2.2.3.64.2.

3.1.4.1.2 Making Changes Involving SMS Items

SMS items can be changed (updated) or deleted the same way that e-mail items are changed or deleted. SMS items can also be added. When an SMS item is added, changed, or deleted, the client sends a **Sync** command request that includes the **airsync:Add** ([MS-ASCMD] section 2.2.3.7.2), **airsync:Change**([MS-ASCMD] section 2.2.3.24), and **airsync:Delete** ([MS-ASCMD] section 2.2.3.42.2) elements, respectively.

The SMS data is specified in the **SMS** class elements, which MUST be children of the **airsync:ApplicationData** element, as specified in [MS-ASCMD] section 2.2.3.11. The following restrictions apply:

- **Sync** request with **airsync:Add**: The **conversation ID** and **conversation index** cannot be sent to the server by clients when adding an SMS item to the server; the conversation ID and conversation index, specified in the **email2:ConversationId** and **email2:ConversationIndex** elements, respectively, are returned by the server in the **Sync** command response, along with the server ID for the SMS item that was added.

- **Sync** request with **airsync:Change**: The client is allowed to update only the read status and the flags of an SMS item. Therefore, only the **email:Read** element and child elements of the **email:Flag** element can be specified as part of the **airsync:Change** element in the **Sync** command request.

For more details about the **SMS** class elements, see section 2.2..

3.1.4.1.3 Special Case for Synchronization of Outbox

When a mobile device is enabled to send an outbound SMS message, the mobile device MUST synchronize the SMS items in the user's **Outbox folder**. It SHOULD do so at regular, short intervals (about every 15 minutes) or use hanging synchronization to provide the user with the best experience in sending SMS items.

The **Sync** command response is further specified in [MS-ASCMD] section 2.2.2.19.2.

3.1.4.2 Estimating the Number of Changes

The client sends a **GetItemEstimate** command request to the server to get an estimate of the number of changes to SMS items for a given e-mail folder. The estimate will include changes to SMS items for that e-mail folder only if the request includes an **airsync:Options** block with the **airsync:Class** element set to "SMS".

A maximum of two **airsync:Options** blocks are allowed within a **getitemestimate:Collection** block, namely one for the default class of the given folder and one for SMS items. The **airsync:Class** element MUST be set to "SMS" to indicate that the options apply to SMS items. An SMS **airsync:Options** block can be included within any **getitemestimate:Collection** block that specifies an e-mail folder.<7>

For more details about the **airsync:Class** element as a child of the **airsync:Options** element, see section 2.2.2.2.1.1. The **GetItemEstimate** command request is further specified in [MS-ASCMD] section 2.2.2.7.1.

3.1.4.3   Provisioning for Relay of Outbound SMS Messages

The client sends a **Settings** command request to provision the server for sending outbound SMS messages through the mobile device. Outbound SMS messages are sent only through mobile devices that enable it. To enable outbound SMS messages, the following elements are included as children of the **settings:Set** element in the **settings:DeviceInformation** block of the **Settings** command request:

- **settings:EnableOutboundSMS**, set to 1. See section 2.2.2.2.2.

- **settings:PhoneNumber**, specifying the mobile device's telephone number.

To disable outbound SMS, the **settings:EnableOutboundSMS** element is set to 0. In this case, the **settings:PhoneNumber** element is not needed.

For more details about the **Settings** command request, see [MS-ASCMD] section 2.2.2.16.1.

3.1.5   Message Processing Events and Sequencing Rules

3.1.5.1   Sending Outbound SMS Messages

The mobile device performs the following tasks to send an outgoing SMS message:

1. The client MUST synchronize the SMS items in the user's Outbox folder. See section 3.1.4.1 for details about synchronizing SMS items. Any SMS items that the server adds to the Outbox folder are outgoing SMS messages. It is important that the client either synchronize the Outbox folder at regular, short intervals (about every 15 minutes) or retrieve the outgoing SMS items by using hanging synchronization. Doing so allows the client to send SMS items in the shortest possible interval.

2. The client MUST send the outgoing SMS items via the mobile device's wireless network. The client sends the given SMS item to all of the mobile recipients provided by the server. The client MUST logically split the item's body as necessary to comply with its network and MUST enable the reassembling of the body on the receiver's phone.

3. If a problem occurs with the message delivery, the client SHOULD generate a **non-delivery report** and post it in the user's **Inbox folder** by sending a **Sync** request that includes the **airsync:Add** element. See section 3.1.4.1.2 for details about adding SMS items to a folder.

4. The client MUST delete the outgoing SMS items by sending a **Sync** request that includes the **airsync:Delete** element. For more details about deleting SMS items from a folder, see section 3.1.4.1.2.

For more details about synchronization processing and sequencing, see [MS-ASCMD] section 3.1.5.3 and [MS-ASCMD] section 3.1.5.4.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Server Details

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

3.2.5.1 Processing a Sync Request

Synchronization of SMS items is initiated by the client, as specified in section 3.1.4.1. The server responds with a **Sync** command response. To send **SMS** class data back to the client, the server includes an **airsync:ApplicationData** element in the **Sync** command response. The **SMS** class data is specified in the **SMS** class elements (section 2.2), which MUST be children of the **airsync:ApplicationData** element.

The server MUST NOT send SMS items to the client if the client did not request them.

The server returns status code 4 (protocol error) in the **Sync** command response for the following situations:

- The **Sync** command request specifies a filter value that is not in the allowed range of values (0 through 5). For more details about filtering, see section 3.1.4.1.1.2.

- The **Sync** command request includes more than two **airsync:Options** blocks.

- The **Sync** command request includes an **airsync:Options** block that specifies a class other than "SMS" or "Email" for a collection containing e-mail items.

- The **Sync** command request includes an **airsync:Options** block that specifies the "SMS" or "Email" class for a collection that does not contain e-mail items.

The **Sync** command response is further specified in [MS-ASCMD] section 2.2.2.19.2. For more details about synchronization processing and sequencing, see [MS-ASCMD] section 3.1.5.3 and [MS-ASCMD] section 3.1.5.4.

3.2.5.2   Processing a GetItemEstimate Request

The client sends a **GetItemEstimate** command request, as specified in section 3.1.4.2. The server sends a **GetItemEstimate** command response with an estimate of the total number of changes for a given folder. The estimate includes the changes to items of the classes specified in the request. The estimate includes changes to SMS items for the specified e-mail folders only if the request includes an **airsync:Options** block with the **airsync:Class** element set to "SMS".

The server returns status code 4 (protocol error) in the **GetItemEstimate** command response for the following situations:

- The **GetItemEstimate** command request includes more than two **airsync:Options** blocks.

- The **GetItemEstimate** command request includes an **airsync:Options** block that specifies a class other than the **SMS** or Email class for a collection containing e-mail items.

- The **GetItemEstimate** command request includes an **airsync:Options** block that specifies the **SMS** or Email class for a collection that does not contain e-mail items.

The **GetItemEstimate** command response is further specified in [MS-ASCMD] section 2.2.2.7.2.

3.2.5.3   Processing a Settings Request

The client sends a **Settings** command request, as specified in section 3.1.4.3. If the **Settings** command request includes the **settings:EnableOutboundSMS** element set to 1 and no telephone number is specified for the mobile device, the server returns status code 5 (invalid arguments) in the **Settings** command response.

If the **Settings** command request indicates that the mobile device can be used to send outbound SMS messages, the server SHOULD propagate the mobile device's outbound SMS state to all clients of the server, allowing any of these clients to send SMS items via the given mobile device.

The **Settings** command response is further specified in [MS-ASCMD] section 2.2.2.16.2.

3.2.6   Timer Events

None.

3.2.7   Other Local Events

None.

4   Protocol Examples

4.1   Synchronizing E-Mail Items and SMS Items

The following is an example of how the options are specified when the client synchronizes e-mail items and SMS items.

Client's **Sync** command request:

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:" xmlns:airsyncbase="AirSyncBase:">
  <Collections>
    <Collection>
      <SyncKey>601771687</SyncKey>
      <CollectionId>15</CollectionId>
      <DeletesAsMoves/>
      <GetChanges/>
      <WindowSize>100</WindowSize>
      <Options>
        <Class>SMS</Class>
        <FilterType>0</FilterType>
        <airsyncbase:BodyPreference>
          <airsyncbase:Type>1</airsyncbase:Type>
          <airsyncbase:TruncationSize>102400</airsyncbase:TruncationSize>
        </airsyncbase:BodyPreference>
      </Options>
      <Options>
        <FilterType>2</FilterType>

<airsyncbase:BodyPreference><airsyncbase:Type>1</airsyncbase:Type></airsyncbase:BodyPreferenc
e>

<airsyncbase:BodyPreference><airsyncbase:Type>2</airsyncbase:Type></airsyncbase:BodyPreferenc
e>
        <airsyncbase:BodyPreference>
          <airsyncbase:Type>4</airsyncbase:Type>
          <airsyncbase:TruncationSize>102400</airsyncbase:TruncationSize>
        </airsyncbase:BodyPreference>
        <MIMESupport>0</MIMESupport>
        <Conflict>1</Conflict>
      </Options>
      <Commands>
         ...
      </Commands>
    </Collection>
  </Collections>
</Sync>
```

4.2   Synchronizing Only SMS Items

The following is an example of how the options are specified when the client synchronizes only SMS items.

Client's **Sync** command request:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<Sync xmlns="AirSync:" xmlns:airsyncbase="AirSyncBase:">
  <Collections>
    <Collection>
      <SyncKey>601771687</SyncKey>
      <CollectionId>15</CollectionId>
      <DeletesAsMoves/>
      <GetChanges/>
      <WindowSize>100</WindowSize>
      <Options>
        <Class>SMS</Class>
        <FilterType>0</FilterType>
        <airsyncbase:BodyPreference>
          <airsyncbase:Type>1</airsyncbase:Type>
          <airsyncbase:TruncationSize>102400</airsyncbase:TruncationSize>
        </airsyncbase:BodyPreference>
      </Options>
      <Commands>
          ...
      </Commands>
    </Collection>
  </Collections>
</Sync>
```

4.3 SMS Message Added by the Server

The following is an example of an SMS item being added by the server to the client. Note that the **SMS** class data is included within the **ApplicationData** element, whereas the class of the item is specified by the **Class** element in the **Add** element.

Server's **Sync** command response:

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:" xmlns:airsyncbase="AirSyncBase:" xmlns:email="Email:"
xmlns:email2="Email2:" >
  <Collections>
    <Collection>
      <SyncKey>525665452</SyncKey>
      <CollectionId>55</CollectionId>
      <Status>1</Status>
      <Commands>
        <Add>
          <Class>SMS</Class>
          <ServerId>55:11</ServerId>
          <ApplicationData>
            <email:To>"14255550143" [MOBILE:14255550143]</email:To>
            <email:From>"+14255550123" [MOBILE:+14255550123]</email:From>
            <email:DateReceived>2009-01-08T00:14:36.000Z</email:DateReceived>
            <email:Importance>1</email:Importance>
            <email:Read>0</email:Read>
            <airsyncbase:Body>
              <airsyncbase:Type>1</airsyncbase:Type>
              <airsyncbase:EstimatedDataSize>29</airsyncbase:EstimatedDataSize>
              <airsyncbase:Data>Make sure you get some rest!
</airsyncbase:Data>
            </airsyncbase:Body>
            <email:InternetCPID>1252</email:InternetCPID>
            <email:Flag/>
```

```
                    <email2:ConversationId>ñòaöÏMDÿ‹döX3&#xF;0&#x0;</email2:ConversationId>
                    <email2:ConversationIndex>Éq&amp;HÞ</email2:ConversationIndex>
                </ApplicationData>
            </Add>
        </Commands>
    </Collection>
  </Collections>
</Sync>
```

4.4 Enabling Outbound SMS Messages

The following example shows how a mobile-device client enables the server to use that mobile-device client for relaying outbound SMS messages.

Mobile-device client's **Settings** command request:

```
<?xml version="1.0" encoding="utf-8"?>
<Settings xmlns="Settings:">
  <DeviceInformation>
    <Set>
      <Model>Manufacturer-Name-Number</Model>
      <IMEI>123456789012345</IMEI>
      <FriendlyName>My PPC Phone</FriendlyName>
      <OS>PPC</OS>
      <OSLanguage>us-EN</OSLanguage>
      <PhoneNumber>206-555-0112</PhoneNumber>
      <EnableOutboundSMS>1</EnableOutboundSMS>
      <MobileOperator>T-Mojo</MobileOperator>
    </Set>
  </DeviceInformation>
</Settings>
```

Release: February 11, 2013

5 Security

5.1 Security Considerations for Implementers

None.

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 2010

- Microsoft Exchange Server 2013

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.

<1> Section 2.2.2: The **airsync:Class** and **airsync:Delete** elements are not returned in a **Sync** response when the MS-ASProtocolVersion header is set to 14.0.

<2> Section 2.2.2.2.1.2: The **airsync:Class** and **airsync:Delete** elements are not returned in a **Sync** response when the MS-ASProtocolVersion header is set to 14.0.

<3> Section 2.2.2.2.2: The **settings:EnableOutboundSMS** element is not supported when the MS-ASProtocolVersion header is set to 12.1.

<4> Section 2.2.2.2.2: The initial release version of Exchange 2010 requires the **settings:PhoneNumber** element to have a value when the **settings:EnableOutboundSMS** element is set to 1. Under these conditions, if the **settings:PhoneNumber** element does not have a value, a Status value of 5 is returned by the server. Exchange 2010 SP1 and Exchange 2013 do not require the **settings:PhoneNumber** element to have a value when the **settings:EnableOutboundSMS** element is set to 1.

<5> Section 2.2.2.2.3: The initial release version of Exchange 2010 requires the **settings:PhoneNumber** element to have a value when the **settings:EnableOutboundSMS** element is set to 1. Under these conditions, if the **settings:PhoneNumber** element does not have a value, a Status value of 5 is returned by the server. Exchange 2010 SP1 and Exchange 2013 do not require the **settings:PhoneNumber** element to have a value when the **settings:EnableOutboundSMS** element is set to 1.

<6> Section 3.1.4.1: When the MS-ASProtocolVersion header is set to 14.0, synchronization of SMS items is only supported in the Inbox folder, Outbox folder, and Sent Items folder.

<7> Section 3.1.4.2: When the MS-ASProtocolVersion header is set to 14.0, synchronization of SMS items is only supported in the Inbox folder, Outbox folder, and Sent Items folder.

7   Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

8   Index