[MS-ASMS]:
ActiveSync Short Message Service Protocol Specification

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.

- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.

- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.

- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft's Open Specification Promise (available here: http://www.microsoft.com/interop/osp) or the Community Promise (available here: http://www.microsoft.com/interop/cp/default.mspx). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.

- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.

- **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.

Table of Contents

1   Introduction

This document specifies the ActiveSync **Short Message Service (SMS)** protocol, which is 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.

1.1   Glossary

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

> **class**
> **code page**
> **conversation**
> **conversation ID**
> **conversation index**
> **Coordinated Universal Time (UTC)**
> **MIME**
> **non-delivery report (NDR)**
> **Short Message Service (SMS)**
> **WAP Binary XML (WBXML)**
> **XML**
> **XML schema**

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

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, "ActiveSync AirSyncBase Namespace Protocol Specification", December 2008.

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

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

[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.

[XML] Bray, T., et al., "Extensible Markup Language (XML) 1.0 (Fifth Edition)", November 2008, http://www.w3.org/TR/REC-xml/.

1.2.2 Informative References

None.

1.3 Protocol Overview

The ActiveSync Short Message Service (SMS) 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.

The 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 (for details about the E-mail **class**, see [MS-ASEMAIL]) such as to and from, as well as body, flag, and importance.

1.4 Relationship to Other Protocols

The ActiveSync Short Message Service (SMS) protocol consists of a series of XML elements that are embedded inside a command request or a command response. For details about command requests and responses, see [MS-ASCMD]. The **WAP Binary XML (WBXML)**, specified in [MS-ASWBXML], is used to transmit the XML markup that constitutes the request body and the response body.

The ActiveSync Short Message Service (SMS) protocol defines elements and complex types according to the data type definitions that are specified in [MS-ASDTYPE]. The E-mail **code page**, specified in [MS-ASWBXML] section 2.2.1.2.3, is used to encapsulate the SMS class data.

1.5 Prerequisites/Preconditions

None.

1.6 Applicability Statement

The ActiveSync Short Message Service (SMS) protocol is applicable for mobile devices that need to synchronize SMS messages with the server and want to allow the server to use it for relaying outbound SMS messages.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

The ActiveSync Short Message Service (SMS) 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 WAP Binary XML (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 the ActiveSync Short Message Service (SMS) protocol MUST be well-formed XML, as specified in [XML]. The XML elements are embedded inside a command request or command response. For details about command requests and responses, see [MS-ASCMD].

The complex types and elements of the ActiveSync Short Message Service (SMS) protocol are defined in four namespaces: **AirSync**, **Email**, **AirSyncBase**, and **Settings**. The **XML schema** definition for the SMS class is as follows. This schema represents the full set of data that is returned by the server in a **Sync** command response.

```
<?xml version="1.0" ?>
<xs:schema xmlns:tns="EMAIL:" attributeFormDefault="unqualified"
elementFormDefault="qualified"
targetNamespace="EMAIL:" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:A="AirSyncBase:">
    <xs:import namespace="AirSyncBase" />
    <xs:element name="To" type="xs:string" />
    <xs:element name="From" type="xs:string" />
    <xs:element name="DateReceived" type="xs:dateTime" />
    <xs:element name="Importance" type="xs:unsignedByte" />
    <xs:element name="Read" type="xs:unsignedByte" />
    <xs:element name="Body" type="A:Body" />
    <xs:element name="InternetCPID" type="xs:string" />
    <xs:element name="Flag">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Subject" type="xs:string" />
                <xs:element name="Status" type="xs:unsignedByte" />
                <xs:element name="FlagType" type="xs:string" />
                <xs:element name="DateCompleted" type="xs:dateTime" />
                <xs:element name="CompleteTime" type="xs:dateTime" />
                <xs:element name="StartDate" type="xs:dateTime" />
                <xs:element name="DueDate" type="xs:dateTime" />
                <xs:element name="UTCStartDate" type="xs:dateTime" />
                <xs:element name="UTCEndDate" type="xs:dateTime" />
                <xs:element name="ReminderSet" type="xs:unsignedByte" />
                <xs:element name="ReminderTime" type="xs:dateTime" />
                <xs:element name="OrdinalDate" type="xs:dateTime" />
                <xs:element name="SubOrdinalDate" type="xs:dateTime" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="ConversationId" type="xs:string" />
    <xs:element name="ConversationIndex" type="xs:string" />
```

```
</xs:schema>
```

2.2.1   Complex Types

The following table summarizes the set of common XML schema complex types that are defined by the SMS class in the **Email** and **AirSyncBase** namespaces. These complex types are used to specify SMS data. For more details about these complex types, see [MS-ASEMAIL] section 2.2.1.

Complex Type	Description
Body	A description of the body text, along with its data.
Flag	The flag that is associated with the item, along with its current status.

2.2.2   Elements

These elements do not have child elements in either the command request or the command response. If a request contains incorrectly placed elements, the server returns a status value of 103 in the response.

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, or **Settings** command.

Element	Description
Class (as child of **Options**)	Specifies the class to which the options apply in a **GetItemEstimate** command request or a **Sync** command request.
Class (as child of **Add**, **Change**, or **Delete**)	Specifies the class to which the **Add**, **Change**, or **Delete** operation applies in a **Sync** command request or response.
DeviceInformation.Set.EnableOutboundSMS	Used in a **Settings** command request to enable or disable the sending of outbound SMS messages through a mobile device.
DeviceInformation.Set.PhoneNumber	Specifies the telephone number that identifies the mobile device.

The following table summarizes the set of common XML schema elements that are defined by the SMS class in the **Email** namespace. These elements are used to specify SMS data. The semantics for all these elements, except **To**, **From**, and **DateReceived**, are the same for both the SMS class and the E-mail class. The **To**, **From**, and **DateReceived** elements for the SMS class are further specified in section 2.2.2.1 of this document. For more details about all the other elements, see [MS-ASEMAIL] section 2.2.2.

Element	Description
To	The list of recipients.
From	The telephone number of the individual who sent the SMS message.
DateReceived	The date and time when the SMS message was either received by the device or sent by the device, depending on whether the SMS message is in the Sent Items

Element	Description
	folder.
Importance	The importance of the message, as determined by the sender.
Read	Specifies whether the message has been read.
InternetCPID	The original code page ID from the **MIME** message representation of the SMS item.
Flag.Subject	The subject of the flag as it would appear in a task list.
Flag.Status	The current status of the flag.
Flag.FlagType	The value of the **Flag To: follow up** field.
Flag.DateCompleted	The date on which the flagged item was completed.
Flag.CompleteTime	The time at which the flagged item was marked as finished.
Flag.StartDate	The start date of the flagged item.
Flag.DueDate	The due date of the flagged item.
Flag.UTCStartDate	The **Coordinated Universal Time (UTC)** value of the local **StartDate**.
Flag.UTCEndDate	The UTC value of the local **DueDate**.
Flag.ReminderSet	Identifies whether a reminder has been set for this flagged item.
Flag.ReminderTime	The date and time that the reminder is supposed to occur.
Flag.OrdinalDate	The time at which the client set the flag.
Flag.SubOrdinalDate	A string used to sort items.
ConversationId	Specifies the **conversation ID** for an SMS item.
ConversationIndex	Specifies the **conversation index** for an SMS item.

2.2.2.1   SMS Class

2.2.2.1.1   From

The **From** element is an optional element that specifies the telephone number of the individual who sent the SMS message.

The value of this element is a **string** type, as specified in [MS-ASDTYPE] section 2.1. 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.

2.2.2.1.2 DateReceived

The **DateReceived** element is an optional element that specifies the date and time when the SMS message was either received by the device or sent by the device, depending on whether the SMS message is in the Sent Items folder. 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 **date/time** type, as specified in [MS-ASDTYPE] section 2.6.

2.2.2.1.3 To

The **To** element is an optional element that specifies the list of primary recipients

The value of this element is a **string** type, as specified in [MS-ASDTYPE] section 2.1. The value comprises one or more e-mail addresses. If there are multiple e-mail addresses, then they are separated by commas.

2.2.2.2 Other Elements

2.2.2.2.1 Class (as Child of Options)

The Class element is an optional child element of the **Options** type in the **GetItemEstimate** command request or the **Sync** 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 2.1. For SMS items, the value of this element is "SMS".

2.2.2.2.2 Class (as Child of Add, Change, or Delete)

The Class element is an optional child element of the **Add**, **Change**, or **Delete** element in the **Sync** command. The class element is used in the **Sync** command as follows:

- In the **Sync** command response from the server to add, change, or delete an SMS item.

- In the **Sync** command request from the client to add an SMS item.

The class element specifies the class to which the **Add**, **Change**, or **Delete** operation applies in a **Sync** command request or response. For SMS items, the value is "SMS".

2.2.2.2.3 DeviceInformation.Set.EnableOutboundSMS

The **DeviceInformation.Set.EnableOutboundSMS** element 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.) This element is an optional child element of the **Set** type in the **Settings** command request.

The value of this element is an **Integer**, as specified in [MS-ASDTYPE] section 2.2. If this element is set to 1, the mobile device can be used to send outbound SMS messages. If this element is set to 0, 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 **DeviceInformation.Set.EnableOutboundSMS** element set to 0. The default value of this element is 0.

When the **DeviceInformation.Set.EnableOutboundSMS** element is set to 1, the **DeviceInformation.Set.PhoneNumber** element MUST be present.

2.2.2.2.4   DeviceInformation.Set.PhoneNumber

The **DeviceInformation.Set.PhoneNumber** element specifies the telephone number that identifies the mobile device. This element is an optional child element of the **Set** type in the **Settings** command request. This element MUST be present if the **DeviceInformation.Set.EnableOutboundSMS** element is set to 1.

The value of this element is a **string**, as specified in [MS-ASDTYPE] section 2.1.

3   Protocol Details

3.1   Client Details

3.1.1   Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

SMS class: A set of complex types and elements that specifies an SMS item. SMS class data is included in command requests that are sent to the server when SMS items need to be retrieved or synchronized.

Command request: A WBXML-formatted message that adheres to the command schemas specified in [MS-ASCMD].

Relay of outbound SMS messages: The server can use a mobile device client to relay outbound SMS messages if the mobile device client has provisioned the server to do so.

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 only on Inbox, Outbox, and Sent Items collections.

The **Sync** command request is further specified in [MS-ASCMD] section 2.2.1.19.1.

3.1.4.1.1   Options

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

A maximum of two **Options** blocks are allowed within a collection block, namely one for the default class of the given folder and one for SMS items (class is set to "SMS" to indicate SMS items).

SMS items can be synchronized only within the Inbox, the Outbox, and the Sent Items folders.

For more details about the class element for **Options**, see section 2.2.2.2.1.

3.1.4.1.1.1 Sticky Options

Sticky options allow a client to send a request that does not include an XML 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 **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 **SoftDelete** element in the **Sync** response. For more details about the **SoftDelete** element and soft-deletes, see [MS-ASCMD] section 2.2.1.19.2.10.

3.1.4.1.1.2 Filtering

A filter can be applied to the synchronization of SMS items. The **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 **FilterType** element, see [MS-ASCMD] section 2.2.1.19.1.23.

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 **Add**, **Change**, or **Delete** types, respectively.

The SMS data is specified in the SMS class complex types and elements, which MUST be children of the **ApplicationData** type, as specified in [MS-ASCMD] section 2.2.1.19.1.7. The following restrictions apply:

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

See section 2.2 for details about the SMS class complex types and elements. For more details about the **Add**, **Change**, or **Delete** types in the **Sync** command request, see [MS-ASCMD] sections 2.2.1.19.1.6, 2.2.1.19.1.8, and 2.2.1.19.1.18, respectively.

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. It SHOULD do so at regular, short intervals (about every 15 minutes) or leverage 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 2.2.1.19.2.

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 (resulting from add, change, and delete operations) 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 **Options** block with the **Class** element set to "SMS".

A maximum of two **Options** blocks are allowed within a **Collection** block, namely one for the default class of the given folder and one for SMS items (**Class** element is set to "SMS" to indicate an SMS item). An SMS **Options** block can be included only within a collection block that specifies one of the following folders:

- Inbox

- Outbox

- Sent Items

For more details about the **Class** element for **Options**, see section 2.2.2.2.1. The **GetItemEstimate** command request is further specified in [MS-ASCMD] section 2.2.1.7.1.

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 **Set** element in the **DeviceInformation** block of the **Settings** command request:

- **EnableOutboundSMS**, set to 1. See section 2.2.2.2.3.

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

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

For more details about the **EnableOutboundSMS** element, see section 2.2.2.2.3. For more details about the **Settings** command request, see [MS-ASCMD] section 2.2.1.16.1.

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. 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 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 (NDR)** and post it in the user's Inbox by sending a **Sync** request that includes the **Add** type. 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 **Delete** type. See section 3.1.4.1.2 for details about deleting SMS items from a folder.

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

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

SMS class: A set of complex types and elements that specifies an SMS item. SMS class data is included in command requests that are sent to the server when SMS items need to be retrieved or synchronized.

Command response: A WBXML-formatted message that adheres to the command schemas specified in [MS-ASCMD]. The server can return zero or more SMS class blocks in its command response, depending on how many SMS items match the criteria specified in the client's command request. The server returns an SMS class XML block for every SMS item that matches the criteria specified in the client command request.

Relay of outbound SMS messages: The server can use a mobile device client to relay outbound SMS messages if the mobile device client has provisioned the server to do so.

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 **ApplicationData** type in the **Sync** command response. The SMS class data is specified in the SMS class complex types and elements (section 2.2), which MUST be children of the **ApplicationData** type, as specified in [MS-ASCMD] section 2.2.1.19.2.2.

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 includes more than two **Options** blocks

- The **Sync** command request includes an **Options** block that specifies a class other than "SMS" or "Email"

- The **Sync** command request includes an **Options** block for a collection that is not Inbox, Outbox, or Sent Items.

The **Sync** command response is further specified in [MS-ASCMD] section 2.2.1.19.2. For more details about synchronization processing and sequencing, see [MS-ASCMD] sections 3.1.5.3 and 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 (resulting from add, change, and delete operations) for a given folder. The estimate includes the changes to items of the class types specified in the request. The estimate includes changes to SMS items for the specified e-mail folders only if the request includes an **Options** block with the class element set to "SMS".

The **GetItemEstimate** command response is further specified in [MS-ASCMD] section 2.2.1.7.2.

3.2.5.3 Processing a Settings Request

The client sends a **Settings** command request, as specified in 3.1.4.3. If the **Settings** command request includes the **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 2.2.1.16.2.

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 wants to synchronize e-mail items and SMS items.

Client's **Sync** command request:

```xml
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:" xmlns:A="AirSyncBase:" xmlns:B="POOMMAIL:" xmlns:C="POOMTASKS:">
  <Collections>
    <Collection>
      <SyncKey>601771687</SyncKey>
      <CollectionId>15</CollectionId>
      <DeletesAsMoves/>
      <GetChanges/>
      <WindowSize>100</WindowSize>
      <Options>
        <Class>SMS</Class>
        <FilterType>0</FilterType>
        <A:BodyPreference>
          <A:Type>1</A:Type>
          <A:TruncationSize>102400</A:TruncationSize>
        </A:BodyPreference>
      </Options>
      <Options>
        <FilterType>2</FilterType>
        <A:BodyPreference><A:Type>1</A:Type></A:BodyPreference>
        <A:BodyPreference><A:Type>2</A:Type></A:BodyPreference>
        <A:BodyPreference>
          <A:Type>4</A:Type>
          <A:TruncationSize>102400</A:TruncationSize>
        </A: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 a how the options are specified when the client wants to synchronize only SMS items.

Client's **Sync** command request:

```xml
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:" xmlns:A="AirSyncBase:" xmlns:B="POOMMAIL:" xmlns:C="POOMTASKS:">
  <Collections>
    <Collection>
      <SyncKey>601771687</SyncKey>
```

```
        <CollectionId>15</CollectionId>
        <DeletesAsMoves/>
        <GetChanges/>
        <WindowSize>100</WindowSize>
        <Options>
          <Class>SMS</Class>
          <FilterType>0</FilterType>
          <A:BodyPreference>
            <A:Type>1</A:Type>
            <A:TruncationSize>102400</A:TruncationSize>
          </A: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** node, whereas the class of the item is specified by the **Class** element in the **Add** node.

Server's **Sync** command response:

```
<?xml version="1.0" encoding="utf-8"?>
<Sync xmlns="AirSync:" xmlns:A="AirSyncBase:" xmlns:B="POOMMAIL:" xmlns:C="POOMMAIL2:" >
  <Collections>
    <Collection>
      <SyncKey>525665452</SyncKey>
      <CollectionId>55</CollectionId>
      <Status>1</Status>
      <Commands>
        <Add>
          <Class>SMS</Class>
          <ServerId>55:11</ServerId>
          <ApplicationData>
            <B:To>"14255550143" [MOBILE:14255550143]</B:To>
            <B:From>"+14255550123" [MOBILE:+14255550123]</B:From>
            <B:DateReceived>2009-01-08T00:14:36.000Z</B:DateReceived>
            <B:Importance>1</B:Importance>
            <B:Read>0</B:Read>
            <A:Body>
              <A:Type>1</A:Type>
              <A:EstimatedDataSize>29</A:EstimatedDataSize>
              <A:Data>Make sure you get some rest!
</A:Data>
            </A:Body>
            <B:InternetCPID>1252</B:InternetCPID>
            <B:Flag/>
            <C:ConversationId>ñòaöÏMDÿ‹döX3&#xF;0&#x0;</C:ConversationId>
            <C:ConversationIndex>Éq&amp;HÞ</C: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>
      <Language>us-EN</Language>
      <PhoneNumber>206-555-0112</PhoneNumber>
      <EnableOutboundSMS>1</EnableOutboundSMS>
      <MobileOperator>T-Mojo</MobileOperator>
    </Set>
  </DeviceInformation>
</Settings>
```

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 product versions. References to product versions include released service packs.

- Microsoft Exchange Server 2010

Exceptions, if any, are noted below. If a service pack number appears with the product version, behavior changed in that service pack. The new behavior also applies to subsequent service packs of the product unless otherwise specified.

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

7 Change Tracking

This section identifies changes made to [MS-ASMS] protocol documentation between November 2009 and February 2010 releases. Changes are classed as major, minor, or editorial.

Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.

- An extensive rewrite, addition, or deletion of major portions of content.

- A protocol is deprecated.

- The removal of a document from the documentation set.

- Changes made for template compliance.

Minor changes do not affect protocol interoperability or implementation. Examples are updates to fix technical accuracy or ambiguity at the sentence, paragraph, or table level.

Editorial changes apply to grammatical, formatting, and style issues.

No changes means that the document is identical to its last release.

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

- New content added.

- Content update.

- Content removed.

- New product behavior note added.

- Product behavior note updated.

- Product behavior note removed.

- New protocol syntax added.

- Protocol syntax updated.

- Protocol syntax removed.

- New content added due to protocol revision.

- Content updated due to protocol revision.

- Content removed due to protocol revision.

- New protocol syntax added due to protocol revision.

- Protocol syntax updated due to protocol revision.

- Protocol syntax removed due to protocol revision.

- New content added for template compliance.

- Content updated for template compliance.

- Content removed for template compliance.

- Obsolete document removed.

Editorial changes always have the revision type "Editorially updated."

Some important terms used in revision type descriptions are defined as follows:

Protocol syntax refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.

Protocol revision refers to changes made to a protocol that affect the bits that are sent over the wire.

Changes are listed in the following table. If you need further information, please contact protocol@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
2.2.2 Elements	51479 Specified how the server responds if the elements include child elements in the request.	N	Content update.
2.2.2 Elements	51480 Updated the descriptions of the From element and DateReceived element.	N	Content update.
2.2.2 Elements	54196 Added the DeviceInformation.Set.PhoneNumber element to the table of elements used by the GetItemEstimate command and Sync command.	Y	New content added.
2.2.2.1.1 From	51480 Updated the description of the From element to match the description in the table in section 2.2.2.	N	Content update.
2.2.2.1.2 DateReceived	51480 Updated the description of the DateReceived element to match the description in the table in section 2.2.2.	N	Content update.
2.2.2.2.1 Class (as Child of Options)	Added statement to specify the data type and include a reference to [MS-ASDTYPE].	N	Content update.
2.2.2.2.3 DeviceInformation.Set.EnableOutboundSMS	51483 Added statement to specify the default value.	N	Content update.
2.2.2.2.3 DeviceInformation.Set.EnableOutboundS	Added statement to specify the data type and include a reference to [MS-	N	Content update.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Revision Type
MS	ASDTYPE].		
2.2.2.2.3 DeviceInformation.Set.EnableOutboundSMS	Added statement to specify that the DeviceInformation.Set.PhoneNumber element MUST be present when the DeviceInformation.Set.EnableOutboundSMS is set to 1.	N	Content update.
2.2.2.2.4 DeviceInformation.Set.PhoneNumber	54196 Added new topic.	Y	New content added.
3.1.4.1.1 Options	51484 Specified the folders within which SMS items can be synchronized.	N	Content update.
3.1.4.1.1.1 Sticky Options	51485 Added details about soft-deletes.	N	Content update.
3.1.4.1.1.1 Sticky Options	51486 Clarified and added details about sticky options.	N	Content update.
3.1.4.1.1.2 Filtering	51491 Added details about how filtering is applied.	N	Content update.
3.1.4.1.2 Making Changes Involving SMS Items	51495 Added section references for Add, Change, and Delete.	N	Content update.
3.2.1 Abstract Data Model	Re-worded from "The server MUST return an SMS class XML block" to "The server returns an SMS class XML block".	N	Content update.

Release: Friday, February 5, 2010

8 Index

A