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

Revision Summary

Date	Revision History	Revision Class	Comments
04/10/2009	0.1		Initial Availability.
07/15/2009	1.0	Major	Revised and edited for 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
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 Protocol Specification", 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)", 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)**, as specified in [MS-ASWBXML], is used to transmit the XML markup that constitutes the request body and the response body.

The E-mail **code page**, as specified in [MS-ASWBXML], is used to encode 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 Localization

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 that are used by the ActiveSync Short Message Service (SMS) protocol 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

Elements MUST NOT have child elements in either the command request or 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.

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 **From** and **DateReceived**, are the same for both the SMS class and the E-mail class. The **From** and **DateReceived** elements for the SMS class are further specified in 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 e-mail address of the individual who sent the message.
DateReceived	The date and time that the message was received on the server.
Importance	The importance of the message, as determined by the sender.
Read	Specifies whether the message has been read.

Element	Description
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 individual who sent the SMS item.

The value of this element is a string that contains one sender. The format of the string is as follows, including the quotes and angle 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 item is sent, if the item is in the Sent Items folder.

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

2.2.2.2 Other Elements

2.2.2.2.1 Class (as Child of Options)

The options Class element is an optional child element of the **Options** element in the **GetItemEstimate** command request or the **Sync** command request. It specifies the class to which the options apply. For SMS items, the value is "SMS".

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

The operations 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 item.
- In the **Sync** command request from the client to add an 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 optional. It is used in 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.

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, can reset the property to zero on the server by sending a **Settings** command request with the **DeviceInformation.Set.EnableOutboundSMS** element set to 0.

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.

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 only be synchronized within folders that natively contain e-mail messages.

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, which allows a client to send requests that direct the server to use the previous set of options, are supported. As a result of this, the client MUST 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 MUST include two **Options** blocks. If one set of options is omitted from the request, the server presumes that the item type that corresponds to the missing set of options is no longer desired and soft-deletes are issued for the items of that type.

3.1.4.1.1.2 Filtering

SMS items can be filtered. The same filter values (0 through 5) that are used to filter e-mail items are also used to filer SMS items.

3.1.4.1.2 Making Changes Involving SMS Items

SMS items can be added, changed, or deleted the same way that e-mail items are added, changed, or deleted. When an SMS item is added, changed, or deleted, the client sends a **Sync** command request that includes the **Add**, **Change**, or **Delete** element, 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 2.2.1.19.1.6.

See 2.2 for details about the SMS class complex types and elements. For more details about the **Add**, **Change**, or **Delete** elements in the **Sync** command request, see [MS-ASCMD] section 2.2.1.19.1.

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 is set to "SMS" to indicate SMS items). An SMS **Options** block can only be included within a collection block that specifies a folder that natively contains e-mail messages.

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 are added to the Outbox are outgoing SMS messages. It is important that the client either synchronize this folder at regular, short intervals or retrieve the Outbox SMS items by using hanging synchronization. Doing so allows the client to send SMS items in the shortest possible interval.

2. The client MUST delete the Outbox SMS items by sending a **Sync** request that includes the **Delete** element. See section 3.1.4.1.2 for details about deleting SMS items from a folder.

3. The client MUST send the outgoing SMS items via the mobile device's wireless network. The client sends the given SMS item to all the 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.

4. After the SMS message is sent, the client MUST add the SMS item to the Sent Items folder by sending a **Sync** request that includes the **Add** element. See section 3.1.4.1.2 for details about adding SMS items to a folder.

5. If a problem occurs with the message delivery, the client MUST generate an NDR and post it in the user's Inbox by sending a **Sync** request that includes the **Add** element. See section 3.1.4.1.2 for details about adding SMS items to a folder.

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 MUST return 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** element 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 2.2.1.19.1.6.

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

If the **Sync** command request includes more than two **Options** blocks or if the **Sync** command request includes an **Options** block that specifies a class other than "SMS" or "Email", the server returns a status code 4 (protocol error) in the **Sync** command response.

The **Sync** command response is further specified in [MS-ASCMD] section 2.2.1.19.2.

3.2.5.1.1 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, as specified in section 3.1.5.1. It SHOULD do so at regular, short intervals 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.

3.2.5.2 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 propagates the mobile device's outbound SMS state and telephone number 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 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 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.

Client's **Sync** command request:

```
<?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>6</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 enables outbound SMS messages to be sent through a 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:

- 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 the product does not follow the prescription.

7 Change Tracking

This section will report content and/or editorial changes, beginning with the next release.

8 Index