[MS-OXOSMMS]:
Short Message Service (SMS) and Multimedia Messaging Service (MMS) Object Protocol Specification

Preliminary Documentation. This Open Specification provides documentation for past and current releases and/or for the pre-release (beta) version of this technology. This Open Specification is final documentation for past or current releases as specifically noted in the document, as applicable; it is preliminary documentation for the pre-release (beta) versions. Microsoft will release final documentation in connection with the commercial release of the updated or new version of this technology. As the documentation may change between this preliminary version and the final version of this technology, there are risks in relying on preliminary documentation. To the extent that you incur additional development obligations or any other costs as a result of relying on this preliminary documentation, you do so at your own risk.

Revision Summary

Date	Revision History	Revision Class	Comments
04/04/2008	0.1		Initial Availability.
04/25/2008	0.2		Revised and updated property names and other technical content.
06/27/2008	1.0		Initial Release.
08/06/2008	1.01		Revised and edited technical content.
09/03/2008	1.02		Updated references.
12/03/2008	1.03		Updated IP notice.
04/10/2009	2.0		Updated applicable product releases.
07/15/2009	3.0	Major	Revised and edited for technical content.
11/04/2009	4.0.0	Major	Updated and revised the technical content.
02/10/2010	4.0.0	None	Version 4.0.0 release
05/05/2010	5.0.0	Major	Updated and revised the technical content.
08/04/2010	5.1	Minor	Clarified the meaning of the technical content.
11/03/2010	5.2	Minor	Clarified the meaning of the technical content.
03/18/2011	5.2	No change	No changes to the meaning, language, or formatting of the technical content.
08/05/2011	5.3	Minor	Clarified the meaning of the technical content.
10/07/2011	6.0	Major	Significantly changed the technical content.
01/20/2012	7.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction

The Short Message Service (SMS) and Multimedia Messaging Service (MMS) Object Protocol enables clients and servers to create, modify, and delete **Short Message Service (SMS)** and **Multimedia Messaging Service (MMS)** messages.

This protocol extends the Message and Attachment Object Protocol, as described in [MS-OXCMSG].

Sections 1.8, 2, and 3 of this specification are normative and contain RFC 2119 language. Sections 1.5 and 1.9 are also normative but cannot contain RFC 2119 language. All other sections and examples in this specification are informative.

1.1 Glossary

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

Coordinated Universal Time (UTC)
flags
GUID
handle
Unicode

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

Attachment object
calendar
Hypertext Markup Language (HTML)
Message object
Multimedia Messaging Service (MMS)
named property
plain text
property ID
reminder
remote operation (ROP)
ROP request
ROP response
rule
Short Message Service (SMS)
store
stream

The following terms are specific to this document:

Multimedia Messaging Service (MMS) object: A Message object that represents a Multimedia Messaging Service (MMS) message in a messaging store.

SMS object: A Message object that represents an Short Message Service (SMS) message in a messaging store.

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 Specification documents do not include a publishing year because links are to the latest version of the 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-OXCDATA] Microsoft Corporation, "Data Structures".

[MS-OXCMAIL] Microsoft Corporation, "RFC2822 and MIME to E-Mail Object Conversion Algorithm".

[MS-OXCMSG] Microsoft Corporation, "Message and Attachment Object Protocol Specification".

[MS-OXCPRPT] Microsoft Corporation, "Property and Stream Object Protocol Specification".

[MS-OXOMSG] Microsoft Corporation, "E-Mail Object Protocol Specification".

[MS-OXPROPS] Microsoft Corporation, "Exchange Server Protocols Master Property List".

[MS-OXSHARE] Microsoft Corporation, "Sharing Message Object Protocol Specification".

[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

1.2.2 Informative References

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

[MS-OXCROPS] Microsoft Corporation, "Remote Operations (ROP) List and Encoding Protocol Specification".

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

[SMIL2.1] Bulterman, D., Grassel, G., Jansen, J., Eds., et al., "Synchronized Multimedia Integration Language (SMIL 2.1)", W3C Recommendation, December 2005, http://www.w3.org/TR/2005/REC-SMIL2-20051213/

1.3 Overview

This protocol enables a client or server to create, modify, and delete SMS text messages and MMS multimedia messages in a messaging **store**. SMS text messages are characterized by a short unformatted text body. MMS messages are characterized by text and multimedia components.

This protocol extends the Message and Attachment Object Protocol, as described in [MS-OXCMSG], in that it defines new properties unique to **SMS objects** and **MMS objects** and adds restrictions to the existing of the **Message object**.

A typical scenario for using this protocol is to create, modify, or delete SMS text messages or MMS multimedia messages.

1.4 Relationship to Other Protocols

This protocol has the same dependencies as the Message and Attachment Object Protocol, which it extends. For more details about the Message and Attachment Object Protocol, see [MS-OXCMSG].

1.5 Prerequisites/Preconditions

This protocol has the same prerequisites and preconditions as the Message and Attachment Object Protocol, as specified in [MS-OXCMSG].

1.6 Applicability Statement

None.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

This protocol provides no vendor-extensibility beyond what is already specified in [MS-OXCMSG].

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

The SMS and MMS Object Protocol uses the protocols specified in [MS-OXCPRPT] and [MS-OXCMSG] as its primary transport mechanism.

2.2 Message Syntax

SMS objects and MMS objects can be created and modified by clients and servers. Except where noted below, section 2.2.1 through section 2.2.2.4 define constraints under which both clients and servers operate.

Clients operate on SMS objects and MMS objects using the Message and Attachment Object Protocol, as specified in [MS-OXCMSG]. How a server operates on SMS objects and MMS objects is implementation-dependent. The results of any such operations are exposed to clients in a manner that is consistent with this protocol.

Unless otherwise specified, SMS objects and MMS objects adhere to all property constraints specified in [MS-OXPROPS] and [MS-OXCMSG]. SMS objects and MMS objects can also contain other properties, but these properties have no impact on this protocol.

2.2.1 Common SMS and MMS Object Properties

2.2.1.1 PidNameOMSAccountGuid Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidNameOMSAccountGuid** property ([MS-OXPROPS] section 2.510) encodes the **GUID** of the SMS account used to deliver the message in the following format (including the braces): {DWORD-WORD-WORD-WORD-WORD.DWORD}; for example, "{C200E360-38C5-11CE-AE62-08002B2B79EF}".

2.2.1.2 PidNameOMSScheduleTime Property

Type: **PtypTime** ([MS-OXCDATA] section 2.11.1)

The **PidNameOMSScheduleTime** property ([MS-OXPROPS] section 2.512) specifies the time, in **Coordinated Universal Time (UTC)**, that the client sends the requests instructing the service provider to send the SMS or MMS message.

2.2.1.3 PidNameOMSServiceType Property

Type: **PtypInteger32** ([MS-OXCDATA] section 2.11.1)

The **PidNameOMSServiceType** property ([MS-OXPROPS] section 2.513) indicates the type of service used to send the SMS or MMS message. The value of this property MUST be one of the values in the following table.

Value	Meaning
0x00000001	SMS
0x00000004	MMS

2.2.1.4 PidNameOMSSourceType Property

Type: **PtypInteger32** ([MS-OXCDATA] section 2.11.1)

The **PidNameOMSSourceType** property ([MS-OXPROPS] section 2.514) indicates the source of the SMS or MMS message. The value of this property MUST be one of the values in the following table.

Value	Source type
0x00000000	XMS inspector
0x00000001	**Reminder**
0x00000002	**Calendar** summary
0x00000003	**Rule (1)**
0x00000004	Unknown

2.2.1.5 PidNameContentClass Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidNameContentClass** property ([MS-OXSHARE] section 2.2.5.1) MUST be set to one of the values in the following table.

Value	Meaning
MS-OMS-SMS	SMS
MS-OMS-MMS	MMS

2.2.1.6 PidNameOMSMobileModel Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidNameOMSMobileModel** property ([MS-OXPROPS] section 2.511) indicates the model of the mobile device used to send the SMS or MMS message.

2.2.2 Additional Property Constraints

This protocol makes additional constraints on the properties specified in section 2.2.2.1 through section 2.2.2.4 beyond what is specified in [MS-OXCMSG] and [MS-OXOMSG].

2.2.2.1 PidTagIconIndex Property

Type: **PtypInteger32** ([MS-OXCDATA] section 2.11.1)

The **PidTagIconIndex** property ([MS-OXOMSG] section 2.2.1.10) specifies which icon is to be used by a user interface when displaying a group of SMS objects and/or MMS objects. This property SHOULD be set;<1> if set, the value MUST be 0xFFFFFFFF.

2.2.2.2 PidTagMessageClass Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagMessageClass** property ([MS-OXCMSG] section 2.2.1.3) specifies the type of the Message object. In addition to meeting the criteria specified in [MS-OXCMSG], for SMS objects this value MUST either be set to "IPM.Note.Mobile.SMS" or begin with "IPM.Note.Mobile.SMS". For MMS objects, this property MUST either be set to "IPM.Note.Mobile.MMS" or begin with "IPM.Note.Mobile.MMS.".

2.2.2.3 Body Properties

The contents of SMS objects are stored and retrieved following the details specified for **plain text** messages in [MS-OXCMAIL] section 2.1.3.3.1.<2>

The contents of MMS objects are stored and retrieved following the details specified for **HTML** messages, as specified in [MS-OXCMAIL] section 2.1.3.3.1.<3>

2.2.2.4 PidTagNormalizedSubject Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1)

The **PidTagNormalizedSubject** property ([MS-OXCMSG] section 2.2.1.10) contains an abbreviated version of the contents of the message suitable for displaying groups of SMS objects to a user. For MMS objects, only the constraints specified in [MS-OXCMSG] section 2.2.1.10 apply.

3 Protocol Details

3.1 Common Details

Both the client and server roles create and operate on SMS objects and MMS objects.

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 Creation of an SMS or MMS Object

To create an SMS object or MMS object, the server or client sets properties in accordance with the requirements in section 2 and [MS-OXCPRPT] and then saves the resulting Message object as specified in [MS-OXCMSG].

3.1.4.2 Modification of an SMS or MMS Object

When modifying an SMS object or MMS object, the client or server modifies any of the properties in accordance with the requirements in section 2 and [MS-OXCPRPT] and then saves the Message object as specified in [MS-OXCMSG].

3.1.4.3 Deletion of an SMS or MMS Object

An SMS object or MMS object has no special deletion semantics beyond what is specified in [MS-OXCMSG].

3.1.5 Message Processing Events and Sequencing Rules

None.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

4 Protocol Examples

4.1 Sample SMS Object

A user creates an SMS text message, types in some text, and sends it. The following is a description of what a client might do to accomplish the user's intentions and the responses a server might return. For more details about **remote operations (ROPs)**, see [MS-OXCPRPT] and [MS-OXCMSG].

Before manipulating SMS objects, the client asks the server to map the **named properties** to **property IDs** by sending a **RopGetPropertyIDsFromNames ROP request** ([MS-OXCROPS] section 2.2.8.1).

Property	Property set GUID	NameID
PidNameOMSMobileModel (section 2.2.1.6)	{00020329-0000-0000-C00000000046}	OMSMobileModel
PidNameOMSAccountGuid (section 2.2.1.1)	{00020329-0000-0000-C00000000046}	OMSAccountGuid
PidNameOMSServiceType (section 2.2.1.3)	{00020329-0000-0000-C00000000046}	OMSServiceType
PidNameOMSSourceType (section 2.2.1.4)	{00020329-0000-0000-C00000000046}	OMSSourceType

The server might send a **RopGetPropertyIDsFromNames ROP response** with the following property IDs, which are used in the example that follows. (The actual property IDs are at the discretion of the server.)

Property	Property ID
PidNameOMSMobileModel	0x84C3
PidNameOMSAccountGuid	0x84C4
PidNameOMSServiceType	0x84C5
PidNameOMSSourceType	0x84C6

To create an SMS object, the client uses the **RopCreateMessage** ROP ([MS-OXCROPS] section 2.2.6.2).The server returns a success code and a **handle** to a Message object.

After the user inputs the content for the SMS object, the client transmits the data to the server by using the **RopSetProperties** ROP ([MS-OXCROPS] section 2.2.8.6).

Property	Property ID	Data type	Value
PidNameOMSAccountGuid	0x84C4	**PtypString** ([MS-OXCDATA] section 2.11.1)	{01234567-0123-0123-0123-0123456789AB}
PidNameOMSMobileModel	0x84C3	**PtypString**	(null)

Property	Property ID	Data type	Value
PidNameOMSServiceType	0x84C5	**PtypInteger32** ([MS-OXCDATA] section 2.11.1)	0x00000001
PidNameOMSSourceType	0x84C6	**PtypInteger32**	0x00000000
PidTagBody ([MS-OXCMSG] section 2.2.1.48.1)	0x1000	**PtypString**	What time is the meeting?
PidTagInternetCodepage ([MS-OXCMSG] section 2.2.1.48.6)	0x3FDE	**PtypInteger32**	0x0000FDE9
PidTagMessageClass ([MS-OXCMSG] section 2.2.1.3)	0x001A	**PtypString**	IPM.Note.Mobile.SMS
PidTagNormalizedSubject ([MS-OXCMSG] section 2.2.1.10)	0x0E1D	**PtypString**	What time is the meeting?
PidTagSubjectPrefix ([MS-OXCMSG] section 2.2.1.9)	0x003D	**PtypString**	(null)

When the user sends the message, the client commits the properties on the server by using the **RopSaveChangesMessage** ROP ([MS-OXCROPS] section 2.2.6.3) and then releases the SMS object by using the **RopRelease** ROP ([MS-OXCROPS] section 2.2.15.3). The client then submits the message to an SMS provider using an appropriate messaging protocol.

The values of some properties change during the processing of the **RopSaveChangesMessage** ROP, but the properties defined in this specification do not change.

4.2 Sample MMS Object

A user creates an MMS message, gives it a subject, types some text, attaches a picture, and sends it. The following is a description of what a client might do to accomplish the user's intentions and the responses a server might return. For more details about ROPs, see [MS-OXCPRPT] and [MS-OXCMSG].

Before manipulating an MMS object, the client asks the server to map the named properties to property IDs by sending a **RopGetPropertyIDsFromNames** ROP request ([MS-OXCROPS] section 2.2.8.1).

Property	Property set GUID	NameID
PidNameOMSMobileModel (section 2.2.1.6)	{00020329-0000-0000-C00000000046}	OMSMobileModel
PidNameOMSAccountGuid (section 2.2.1.1)	{00020329-0000-0000-C00000000046}	OMSAccountGuid
PidNameOMSServiceType (section 2.2.1.3)	{00020329-0000-0000-C00000000046}	OMSServiceType
PidNameOMSSourceType (section 2.2.1.4)	{00020329-0000-0000-C00000000046}	OMSSourceType

The server might send a **RopGetPropertyIDsFromNames** ROP response with the following property IDs, which are used in the example that follows. (The actual property IDs are at the discretion of the server.)

Property	Property ID
PidNameOMSMobileModel	0x84CE
PidNameOMSAccountGuid	0x84CF
PidNameOMSServiceType	0x84D0
PidNameOMSSourceType	0x84D1

To create an MMS object, the client uses the **RopCreateMessage** ROP ([MS-OXCROPS] section 2.2.6.2). The server returns a success code and a handle to an object.

After the user inputs the content for the MMS object, the client transmits the data to the server by using the **RopSetProperties** ROP ([MS-OXCROPS] section 2.2.8.6).

Property	Property ID	Data type	Value
PidNameOMSAccountGuid	0x84CF	**PtypString** ([MS-OXCDATA] section 2.11.1)	{01234567-0123-0123-0123456789ABC}
PidNameOMSMobileModel	0x84CE	**PtypString**	(empty)
PidNameOMSServiceType	0x84D0	**PtypInteger32** ([MS-OXCDATA] section 2.11.1)	0x00000004
PidNameOMSSourceType	0x84D1	**PtypInteger32**	0x00000000
PidTagInternetCodepage ([MS-OXCMSG] section 2.2.1.48.6)	0x3FDE	**PtypInteger32**	0x0000FDE9
PidTagHtml ([MS-OXCMSG] section 2.2.1.48.9)	0x1013	**PtypBinary** ([MS-OXCDATA] section 2.11.1)	(The content of this property is given following the table.)
PidTagIconIndex ([MS-OXOMSG] section 2.2.1.10)	0x1080	**PtypInteger32**	0xFFFFFFFFFF
PidTagMessageClass ([MS-OXCMSG] section 2.2.1.3)	0x001A	**PtypString**	IPM.Note.Mobile.MMS
PidTagMessageFlags ([MS-OXCMSG] section 2.2.1.6)	0x0E07	**PtypInteger32**	**Flags**: 0x00000018 MSGFLAG_UNSENT MSGFLAG_HASATTACH
PidTagNormalizedSubject ([MS-OXCMSG] section 2.2.1.10)	0x0E1d	**PtypString**	Here's the photo.
PidTagSubjectPrefix ([MS-OXCMSG] section 2.2.1.9)	0x003d	**PtypString**	(empty)

The **PidTagHtml** property is a binary property containing the following text.

```
<HTML>
<BODY>
<IMG SRC="cid:Att1.jpg@AB1B43B2B0594564.B94EF7ABB12B49BA" border="0">
<BR>
This is the photo you asked for.
<BR>
<A HREF="cid:Att0.txt@AB1B43B2B0594564.B94EF7ABB12B49BA"></A>
</BODY>
</HTML>
```

The client allocates space for a data file in the message by using the **RopCreateAttachment** ROP ([MS-OXCROPS] section 2.2.6.13). In response, the server returns a success code and a handle to an **Attachment object**. The client uses this handle with the **RopSetProperties** ROP ([MS-OXCROPS] section 2.2.8.6) to transmit data about the attachment to the server.

Property	Property ID	Data type	Value
PidTagAttachmentHidden ([MS-OXCMSG] section 2.2.2.24)	0x7FFE	**PtypBoolean** ([MS-OXCDATA] section 2.11.1)	0x01
PidTagAttachMethod ([MS-OXCMSG] section 2.2.2.9)	0x3705	**PtypInteger32**	0x00000001 (ATTACH_BY_VALUE)
PidTagAttachContentId ([MS-OXCMSG] section 2.2.2.26)	0x3712	**PtypString**	MMS.smil@AB1B43B2B0594564.B94EF7ABB12B49BA
PidTagAttachMimeTag ([MS-OXCMSG] section 2.2.2.26)	0x370E	**PtypString**	application/smil
PidTagAttachLongFilename ([MS-OXCMSG] section 2.2.2.10)	0x3707	**PtypString**	MMS.smil

The client sets the contents of the attachment by using the attachment handle with the **RopOpenStream** ROP ([MS-OXCROPS] section 2.2.9.1), passing in the **PidTagAttachDataBinary** property ([MS-OXPROPS] section 2.667) as the property to open. With the handle returned from the **RopOpenStream** ROP, the client calls the **RopWriteStream** ROP ([MS-OXCROPS] section 2.2.9.3), writing out the contents of the Synchronized Multimedia Integration Language (SMIL) file, the format of which is described in [SMIL2.1], describing the layout of the MMS message. The client follows this with the **RopRelease** ROP ([MS-OXCROPS] section 2.2.15.3) on the **stream (2)** handle, then commits the changes by using the **RopSaveChangesAttachment** ROP ([MS-OXCROPS] section 2.2.6.15), releases the handle to the attachment by using and the **RopRelease** ROP.

The client repeats the process from the **RopCreateAttachment** ROP ([MS-OXCROPS] section 2.2.6.13) for the **RopRelease** ROP with the attachment handle twice more, once for a plain-text version of the body, and once for the image. The attachment containing the body uses the following properties and values with the **RopSetProperties** ROP.

Property	Property ID	Data type	Value
PidTagAttachmentHidden	0x7FFE	**PtypBoolean**	0x01
PidTagAttachMethod	0x3705	**PtypInteger32**	0x00000001 (ATTACH_BY_VALUE)
PidTagAttachContentId	0x3712	**PtypString**	Att0.txt@AB1B43B2B0594564.B94EF7ABB12B49BA
PidTagAttachMimeTag	0x370E	**PtypString**	text/plain
PidTagAttachLongFilename	0x3707	**PtypString**	1.txt

The **RopOpenStream** ROP for the plain-text body is also on the **PidTagAttachDataBinary** property, but the contents written are **Unicode** text. The last attachment the client creates contains the image, and the **RopSetProperties** ROP sends the following data.

Property	Property ID	Data type	Value
PidTagAttachmentHidden	0x7FFE	**PtypBoolean**	0x01
PidTagAttachMethod	0x3705	**PtypInteger32**	0x00000001 (ATTACH_BY_VALUE)
PidTagAttachContentId	0x3712	**PtypString**	Att1.jpg@AB1B43B2B0594564.B94EF7ABB12B49BA
PidTagAttachMimeTag	0x370E	**PtypString**	image/jpeg
PidTagAttachLongFilename	0x3707	**PtypString**	photo.jpg

The contents of the **PidTagAttachDataBinary** property on the image attachment are the binary contents of the image file.

When the user sends the message, the client commits the properties on the server by using the **RopSaveChangesMessage** ROP ([MS-OXCROPS] section 2.2.6.3) and then releases the MMS object by using the **RopRelease** ROP. The client then submits the message to an MMS provider using an appropriate messaging protocol.

The values of some properties change during the processing of the **RopSaveChangesMessage** ROP, but the properties specified in this protocol do not change.

5 Security

5.1 Security Considerations for Implementers

There are no special security considerations specific to the SMS and MMS Object Protocol. General security considerations pertaining to the underlying transport apply, as specified in [MS-OXCMSG] and [MS-OXCPRPT].

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 2003
- Microsoft® Exchange Server 2007
- Microsoft® Exchange Server 2010
- Microsoft® Exchange Server 15 Technical Preview
- Microsoft® Office Outlook® 2003
- Microsoft® Office Outlook® 2007
- Microsoft® Outlook® 2010
- Microsoft® Outlook® 15 Technical Preview

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

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

<1> Section 2.2.2.1: Office Outlook 2007 does not set the **PidTagIconIndex** property ([MS-OXOMSG] section 2.2.1.10) on SMS objects or MMS objects.

<2> Section 2.2.2.3: Office Outlook 2007 SP1 sets both the **PidTagBody** property ([MS-OXCMSG] section 2.2.1.48.1) and the **PidTagHtml** property ([MS-OXCMSG] section 2.2.1.48.9) on SMS objects.

<3> Section 2.2.2.3: Office Outlook 2007 SP1 sets both the **PidTagBody** property ([MS-OXCMSG] section 2.2.1.48.1) and the **PidTagHtml** property ([MS-OXCMSG] section 2.2.1.48.9) on MMS objects.

7 Change Tracking

This section identifies changes that were made to the [MS-OXOSMMS] protocol document between the October 2011 and January 2012 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

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

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- The removal of a document from the documentation set.
- Changes made for template compliance.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

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

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

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

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

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

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

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

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

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
6 Appendix A: Product Behavior	Added Exchange 15 Technical Preview and Outlook 15 Technical Preview to the list of applicable product versions.	Y	Content updated.

8 Index

T

V