

[MS-OXOSMMS]: Short Message Service (SMS) and Multimedia Messaging Service (MMS) Object 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
04/04/2008	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.

Table of Contents

1 Introduction	5
1.1 Glossary	5
1.2 References	5
1.2.1 Normative References	5
1.2.2 Informative References	6
1.3 Overview	6
1.4 Relationship to Other Protocols	6
1.5 Prerequisites/Preconditions	6
1.6 Applicability Statement	6
1.7 Versioning and Capability Negotiation	6
1.8 Vendor-Extensible Fields	7
1.9 Standards Assignments	7
2 Messages	8
2.1 Transport	8
2.2 Message Syntax	8
2.2.1 Common SMS and MMS object properties	8
2.2.1.1 PidNameOMSAccountGuid	8
2.2.1.2 PidNameOMSScheduleTime	8
2.2.1.3 PidNameOMSServiceType	8
2.2.1.4 PidNameOMSSourceType	9
2.2.1.5 PidNameContentClass	9
2.2.1.6 PidNameOMSMobileModel	9
2.2.2 Additional Property Constraints	9
2.2.2.1 PidTagIconIndex	9
2.2.2.2 PidTagMessageClass	10
2.2.2.3 Body Properties	10
2.2.2.4 PidTagNormalizedSubject	10
3 Protocol Details	11
3.1 Common Details	11
3.1.1 Abstract Data Model	11
3.1.1.1 Folders	11
3.1.2 Timers	11
3.1.3 Initialization	11
3.1.4 Higher-Layer Triggered Events	11
3.1.4.1 Creation of an SMS or MMS Object	11
3.1.4.2 Modification of an SMS or MMS Object	11
3.1.4.3 Deletion of an SMS or MMS Object	11
3.1.5 Message Processing Events and Sequencing Rules	11
3.1.6 Timer Events	12
3.1.7 Other Local Events	12
4 Protocol Examples	13
4.1 Sample SMS Object	13
4.2 Sample MMS Object	14
5 Security	18
5.1 Security Considerations for Implementers	18
5.2 Index of Security Parameters	18

6	Appendix A: Product Behavior	19
7	Change Tracking.....	20
8	Index	22

1 Introduction

This document specifies the SMS and MMS Object protocol, which defines **properties** of objects that model **Short Message Service (SMS)** and **Multimedia Messaging Service (MMS) messages**.

1.1 Glossary

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

calendar
Coordinated Universal Time (UTC)
Folder object
GUID
handle
HTML
message
Message object
named property
NameID
property(1)
property ID
reminder
remote operation (ROP)
rule
special folder
Short Message Service (SMS)
store
Unicode

The following terms are specific to this document:

SMS object: A Message object that represents an SMS message in a messaging store and that adheres to the relevant property specifications in this document.

MMS: Multimedia Messaging Service, a communications protocol designed for messages containing text, images, and other multimedia content sent between mobile phones.

MMS object: A Message object that represents an MMS message in a messaging store and that adheres to the relevant property specifications in 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-OXCFOLD] Microsoft Corporation, "[Folder Object Protocol Specification](#)", April 2008.

[MS-OXCMAIL] Microsoft Corporation, "[RFC2822 and MIME to E-Mail Object Conversion Protocol Specification](#)", April 2008.

[MS-OXCMSG] Microsoft Corporation, "[Message and Attachment Object Protocol Specification](#)", April 2008.

[MS-OXCPRPT] Microsoft Corporation, "[Property and Stream Object Protocol Specification](#)", April 2008.

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

[MS-OXOMSG] Microsoft Corporation, "[E-Mail Object Protocol Specification](#)", April 2008.

[MS-OXOSFLD] Microsoft Corporation, "[Special Folders Protocol Specification](#)", April 2008.

[MS-OXPROPS] Microsoft Corporation, "[Exchange Server Protocols Master Property List](#)", April 2008.

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

1.2.2 Informative References

[SMIL] Michel, T., "Synchronized Multimedia", March 2008, <http://www.w3.org/AudioVideo/>

1.3 Overview

The SMS and MMS Object protocol specifies the representation of SMS text messages and MMS multimedia messages in a messaging **store**. This protocol extends the Message and Attachment Object protocol in that it defines new properties and adds **restrictions** to the properties that are specified in [\[MS-OXCMSG\]](#).

This document specifies the properties that are unique to **SMS objects** and **MMS objects**. An SMS object is characterized by a short unformatted text body. An MMS object is characterized by text and multimedia components. SMS and MMS objects are stored in **Folder objects**. The SMS and MMS Object protocol also specifies how an SMS or MMS object is created and manipulated.

1.4 Relationship to Other Protocols

The SMS and MMS object 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

The SMS and MMS Object 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 defined in [\[MS-OXCPRPT\]](#) and [\[MS-OXCMSG\]](#) as its primary transport mechanism.

2.2 Message Syntax

SMS and MMS objects can be created and modified by clients and servers. Except where noted below, this section defines constraints under which both clients and servers operate.

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

Unless otherwise specified below, SMS and MMS objects adhere to all property constraints specified in [\[MS-OXPROPS\]](#) and [\[MS-OXCMSG\]](#). SMS and MMS objects can also contain other properties, which are specified in [\[MS-OXPROPS\]](#), but these properties have no impact on the SMS and MMS object protocol.

SMS and MMS object properties can be of several different data types. The following data types used by SMS and MMS object properties are specified in [\[MS-OXCDATA\]](#) section 2.12.1:

- PtypBinary
- PtypInteger32
- PtypString
- PtypString8
- PtypTime

2.2.1 Common SMS and MMS object properties

2.2.1.1 PidNameOMSAccountGuid

Type: **PtypString**

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

Type: **PtypTime**, in **UTC**

The time at which the client requests that the service provider sends the SMS or MMS message.

2.2.1.3 PidNameOMSServiceType

Type: **PtypInteger32**

Indicates the type of service used to send the SMS or MMS Message; MUST be one of the following.

Value	Meaning
0x00000001	SMS
0x00000004	MMS

2.2.1.4 PidNameOMSSourceType

Type: **PtypInteger32**

Indicates the source of the SMS or MMS Message; MUST be one of the following.

Value	Source type
0x00000000	XMS Inspector
0x00000001	Reminder
0x00000002	Calendar Summary
0x00000003	Rule
0x00000004	Unknown

2.2.1.5 PidNameContentClass

Type: **PtypString**

Set on an SMS or MMS object according to [\[MS-OXCMAIL\]](#); MUST be one of the following.

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

2.2.1.6 PidNameOMSMobileModel

Type: **PtypString**, 0x001F

A string that indicates the model of the mobile device used to send the SMS or MMS Message.

2.2.2 Additional Property Constraints

This protocol specifies additional constraints on the following properties beyond what is specified in [\[MS-OXCMSG\]](#) and [\[MS-OXOMSG\]](#).

2.2.2.1 PidTagIconIndex

Type: **PtypInteger32**

Specifies which icon is to be used by a user interface when displaying a group of SMS and/or MMS objects; SHOULD be set [<1>](#); if set, MUST be "0xFFFFFFFF".

2.2.2.2 PidTagMessageClass

Type: **PtypString**, case-insensitive

Specifies the type of the **Message object**. In addition to meeting the criteria specified in [\[MS-OXCMSG\]](#); MUST be "IPM.Note.Mobile.SMS" or begin with "IPM.Note.Mobile.SMS" for SMS objects; MUST be "IPM.Note.Mobile.MMS" or begin with "IPM.Note.Mobile.MMS." for MMS objects.

2.2.2.3 Body Properties

The contents of SMS Message objects are stored and retrieved following the **plain text** body specification in [\[MS-OXCMSG\] <2>](#).

The contents of MMS Message objects are stored and retrieved following the **HTML** body specification in [\[MS-OXCMSG\] <3>](#).

2.2.2.4 PidTagNormalizedSubject

Type: **PtypString**

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 in [\[MS-OXCMSG\]](#) apply.

3 Protocol Details

General protocol details apply, as specified in [\[MS-OXPROPS\]](#) and [\[MS-OXCMSG\]](#).

3.1 Common Details

The client and server roles are to create and operate on SMS and MMS objects, and otherwise operate in their roles as specified in [\[MS-OXCMSG\]](#).

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.

3.1.1.1 Folders

An SMS or MMS object is created in the Drafts, **Inbox** or Sent Items **special Folder**, as specified in [\[MS-OXOSFLD\]](#), unless the end user or user agent explicitly specifies another Folder Object.

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 or MMS object, the server or client sets properties in accordance with the requirements in Section 2 and [\[MS-OXCPRPT\]](#), and 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 or MMS object, the client or server modifies any of the properties in accordance with the requirements in Section 2 and [\[MS-OXCPRPT\]](#), and saves the Message object as specified in [\[MS-OXCMSG\]](#).

3.1.4.3 Deletion of an SMS or MMS Object

An SMS or MMS object has no special deletion semantics beyond what is specified in [\[MS-OXCFOLD\]](#).

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

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

Before manipulating SMS objects, the client needs to ask the server to perform a mapping from **named properties** to **property IDs**, using [RopGetPropertyIDsFromNames](#).

Property	Property set GUID	NameID
PidNameOMSMobileModel	{00020329-0000-0000-C00000000046}	OMSMobileModel
PidNameOMSAccountGuid	{00020329-0000-0000-C00000000046}	OMSAccountGuid
PidNameOMSServiceType	{00020329-0000-0000-C00000000046}	OMSServiceType
PidNameOMSSourceType	{00020329-0000-0000-C00000000046}	OMSSourceType

The server might respond with the following identifiers, which will be used in the example that follows. (The actual identifiers 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 [RopCreateMessage](#). The server returns a success code and a **handle** to a Message object.

After Joe has input his content for the SMS object, the client uses [RopSetProperties](#) to transmit his data to the server.

Property	Property ID	Data type	Value
PidNameOMSAccountGuid	0x84c4	0x001f (PtypString)	{01234567-0123-0123-0123-0123456789ab}
PidNameOMSMobileModel	0x84c3	0x001f (PtypString)	(null)
PidNameOMSServiceType	0x84c5	0x0003 (PtypInteger32)	0x00000001
PidNameOMSSourceType	0x84c6	0x0003 (PtypInteger32)	0x00000000
PidTagBody	0x1000	0x001f (PtypString)	What time is the meeting?

Property	Property ID	Data type	Value
PidTagInternetCodepage	0x3fde	0x0003 (PtypInteger32)	0x0000FDE9
PidTagMessageClass	0x001a	0x001a 0x001f (PtypString)	IPM.Note.Mobile. SMS
PidTagNormalizedSubject	0x0e1d	0x001f (PtypString)	What time is the meeting?
PidTagSubjectPrefix	0x003d	0x001f (PtypString)	(null)

When Joe is ready to send his Message, the client uses [RopSaveChangesMessage](#) to commit the properties on the server, and then [RopRelease](#) to release the SMS object. The client then submits the Message to an SMS provider using an appropriate messaging protocol.

The values of some properties will change during the execution of [RopSaveChangesMessage](#), but the properties specified in [MS-OXOSMMS] will not change.

4.2 Sample MMS Object

Joe creates an MMS object, gives it a subject, types in some text, attaches a picture, and sends it. The following is a description of what a client might do to accomplish Joe'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 needs to ask the server to perform a mapping from named properties to property IDs, using [RopGetPropertyIDsFromNames](#).

Property	Property set GUID	NameID
PidNameOMSMobileModel	{00020329-0000-0000-C00000000046}	OMSMobileModel
PidNameOMSAccountGuid	{00020329-0000-0000-C00000000046}	OMSAccountGuid
PidNameOMSServiceType	{00020329-0000-0000-C00000000046}	OMSServiceType
PidNameOMSSourceType	{00020329-0000-0000-C00000000046}	OMSSourceType

The server might respond with the following identifiers, which will be used in the example that follows. (The actual identifiers 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 [RopCreateMessage](#). The server returns a success code and a handle to an object.

After Joe has input his content for the MMS object, the client uses [RopSetProperties](#) to transmit his data to the server.

Property	Property ID	Data type	Value
PidNameOMSAccountGuid	0x84cf	0x001f (PtypString)	{01234567-0123-0123-0123456789abc}
PidNameOMSMobileModel	0x84ce	0x001f (PtypString)	(empty)
PidNameOMSServiceType	0x84d0	0x0003 (PtypInteger32)	0x00000004
PidNameOMSSourceType	0x84d1	0x0003 (PtypInteger32)	0x00000000
PidTagInternetCodepage	0x3fde	0x0003 (PtypInteger32)	0x0000FDE9
PidTagHtml	0x1013	0x0102 (PtypBinary)	See below
PidTagIconIndex	0x1080	0x0003 (PtypInteger32)	0xFFFFFFFF
PidTagMessageClass	0x001a	0x001a 0x001f (PtypString)	IPM.Note.Mobile.MMS
PidTagMessageFlags	0x0e07	0x0003 (PtypInteger32)	Flags: 0x00000018 MSGFLAG_UNSENT MSGFLAG_HASATTACH
PidTagNormalizedSubject	0x0e1d	0x001f (PtypString)	Here's the photo.
PidTagSubjectPrefix	0x003d	0x001f (PtypString)	(empty)

[PidTagHtml](#) 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 uses [RopCreateAttachment](#) to allocate space for a data file in the message. The server returns a success code and a handle to an Attachment object. The client then uses this handle with [RopSetProperties](#) to transmit data about the **attachment** to the server.

Property	Property ID	Data type	Value
PidTagAttachmentHidden	0x7ffe	0x000b (PtypBoolean)	0x01

Property	Property ID	Data type	Value
PidTagAttachMethod	0x3705	0x0003 (PtypInteger32)	0x00000001 (ATTACH_BY_VALUE)
PidTagAttachContentId	0x3712	0x001f (PtypString)	MMS.smil@AB1B43B2B0594564.B94EF7ABB12B49BA
PidTagAttachMimeTag	0x370e	0x001f (PtypString)	application/smil
PidTagAttachLongFilename	0x3707	0x001f (PtypString)	MMS.smil

The client sets the contents of the attachment by using the attachment handle with [RopOpenStream](#), passing in [PidTagAttachDataBinary](#) as the property to open. With the handle returned from [RopOpenStream](#), the client calls [RopWriteStream](#), writing out the contents of the Synchronized Multimedia Integration Language (SMIL) file, the format of which is defined in [\[SMIL\]](#), describing the layout of the MMS message. The client follows this with [RopRelease](#) on the **stream** handle, then [RopSaveChangesAttachment](#) to commit the changes, and [RopRelease](#) to release the handle to the attachment.

The client repeats the process from [RopCreateAttachment](#) to [RopRelease](#) 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 [RopSetProperties](#).

Property	Property ID	Data type	Value
PidTagAttachmentHidden	0x7ffe	0x000b (PtypBoolean)	0x01
PidTagAttachMethod	0x3705	0x0003 (PtypInteger32)	0x00000001 (ATTACH_BY_VALUE)
PidTagAttachContentId	0x3712	0x001f (PtypString)	Att0.txt@AB1B43B2B0594564.B94EF7ABB12B49BA
PidTagAttachMimeTag	0x370e	0x001f (PtypString)	text/plain
PidTagAttachLongFilename	0x3707	0x001f (PtypString)	1.txt

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

Property	Property ID	Data type	Value
PidTagAttachmentHidden	0x7ffe	0x000b (PtypBoolean)	0x01

Property	Property ID	Data type	Value
PidTagAttachMethod	0x3705	0x0003 (PtypInteger32)	0x00000001 (ATTACH_BY_VALUE)
PidTagAttachContentId	0x3712	0x001f (PtypString)	Att1.jpg@AB1B43B2B0594564.B94EF7ABB12B49BA
PidTagAttachMimeTag	0x370e	0x001f (PtypString)	image/jpeg
PidTagAttachLongFilename	0x3707	0x001f (PtypString)	photo.jpg

The contents of [PidTagAttachDataBinary](#) on the image attachment are the binary contents of the image file.

When Joe is ready to send his message, the client uses [RopSaveChangesMessage](#) to commit the properties on the server, and then [RopRelease](#) to release the MMS object. The client then submits the message to an MMS provider using an appropriate messaging protocol.

The values of some properties will change during the execution of [RopSaveChangesMessage](#), but the properties specified in this protocol will 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 product versions. References to product versions include released service packs.

- Microsoft® Office Outlook® 2003
- Microsoft® Exchange Server 2003
- Microsoft® Office Outlook® 2007
- Microsoft® Exchange Server 2007
- Microsoft® Outlook® 2010
- 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.

[<1> Section 2.2.2.1:](#) Outlook 2007 does not set the [PidTagIconIndex](#) property on SMS or MMS objects.

[<2> Section 2.2.2.3:](#) Outlook 2007 SP1 sets both [PidTagBody](#) and [PidTagHtml](#) on SMS objects.

[<3> Section 2.2.2.3:](#) Outlook 2007 SP1 sets both [PidTagBody](#) and [PidTagHtml](#) on MMS objects.

7 Change Tracking

This section identifies changes made to [MS-OXOSMMS] protocol documentation between February 2010 and May 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
1.3 Overview	Updated the section title.	N	Content updated for template compliance.
2.2.2.1 PidTagIconIndex	51170 Changed "does not always" in the Product Behavior note to "does not."	Y	Content update.

8 Index

A

[Applicability](#) 6

C

[Capability negotiation](#) 6

[Change tracking](#) 20

Client

[overview](#) 11

E

Examples

[overview](#) 13

F

[Fields – vendor-extensible](#) 7

G

[Glossary](#) 5

I

[Implementer – security considerations](#) 18

[Index of security parameters](#) 18

[Informative references](#) 6

[Introduction](#) 5

M

Messages

[overview](#) 8

Messaging

[transport](#) 8

N

[Normative references](#) 5

O

[Overview](#) 6

P

[Parameters – security index](#) 18

[Preconditions](#) 6

[Prerequisites](#) 6

[Product behavior](#) 19

R

References

[informative](#) 6

[normative](#) 5

[Relationship to other protocols](#) 6

S

Security

[implementer considerations](#) 18

[overview](#) 18

[parameter index](#) 18

[Standards Assignments](#) 7

T

[Tracking changes](#) 20

[Transport](#) 8

V

[Vendor-extensible fields](#) 7

[Versioning](#) 6