

[MS-OXCMMSG]: Message and Attachment 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.msp>). 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		Revised and edited technical content.
10/01/2008	1.03		Revised and edited technical content.
12/03/2008	1.04		Updated IP notice.
04/10/2009	2.0		Updated technical content and 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	5.0.0	Major	Updated and revised the technical content.
05/05/2010	6.0.0	Major	Updated and revised the technical content.
08/04/2010	7.0	Major	Significantly changed the technical content.
11/03/2010	8.0	Major	Significantly changed the technical content.
03/18/2011	9.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction	10
1.1 Glossary	10
1.2 References	11
1.2.1 Normative References	11
1.2.2 Informative References	13
1.3 Overview	13
1.3.1 Messages	13
1.3.2 FAI Messages	13
1.3.3 Message Recipients	13
1.3.4 Message Attachments	13
1.4 Relationship to Other Protocols	14
1.5 Prerequisites/Preconditions	14
1.6 Applicability Statement	14
1.7 Versioning and Capability Negotiation	14
1.8 Vendor-Extensible Fields	14
1.9 Standards Assignments	15
2 Messages	16
2.1 Transport	16
2.2 Message Syntax	16
2.2.1 Message Object Properties	16
2.2.1.1 General Properties	16
2.2.1.2 PidTagHasAttachments	16
2.2.1.3 PidTagMessageClass	17
2.2.1.4 PidTagMessageCodepage	17
2.2.1.5 PidTagMessageLocaleId	17
2.2.1.6 PidTagMessageFlags	17
2.2.1.7 PidTagMessageSize	18
2.2.1.8 PidTagMessageStatus	18
2.2.1.9 PidTagSubjectPrefix	19
2.2.1.10 PidTagNormalizedSubject	19
2.2.1.11 PidTagImportance	19
2.2.1.12 PidTagPriority	20
2.2.1.13 PidTagSensitivity	20
2.2.1.14 PidLidSmartNoAttach	20
2.2.1.15 PidLidPrivate	20
2.2.1.16 PidLidSideEffects	21
2.2.1.17 PidNameKeywords	21
2.2.1.18 PidLidCommonStart	21
2.2.1.19 PidLidCommonEnd	21
2.2.1.20 PidTagAutoForwarded	22
2.2.1.21 PidTagAutoForwardComment	22
2.2.1.22 PidLidCategories	22
2.2.1.23 PidLidClassification	22
2.2.1.24 PidLidClassificationDescription	22
2.2.1.25 PidLidClassified	22
2.2.1.26 PidTagInternetReferences	22
2.2.1.27 PidLidInfoPathFormName	23
2.2.1.28 PidTagMimeSkeleton	23
2.2.1.29 PidTagTnefCorrelationKey	23

2.2.1.30	PidTagAddressBookDisplayNamePrintable	23
2.2.1.31	PidTagCreatorEntryId	23
2.2.1.32	PidTagLastModifierEntryId	23
2.2.1.33	PidLidAgingDontAgeMe	23
2.2.1.34	PidLidCurrentVersion	24
2.2.1.35	PidLidCurrentVersionName	24
2.2.1.36	PidTagAlternateRecipientAllowed	24
2.2.1.37	PidTagResponsibility	24
2.2.1.38	PidTagRowid	24
2.2.1.39	PidTagHasNamedProperties	24
2.2.1.40	PidTagRecipientOrder	24
2.2.1.41	PidNameContentBase	25
2.2.1.42	PidNameAcceptLanguage	25
2.2.1.43	PidTagPurportedSenderDomain	25
2.2.1.44	Body Properties	25
2.2.1.44.1	PidTagBody	25
2.2.1.44.2	PidTagNativeBody	25
2.2.1.44.3	PidTagBodyHtml	26
2.2.1.44.4	PidTagRtfCompressed	26
2.2.1.44.5	PidTagRtfInSync	26
2.2.1.44.6	PidTagInternetCodepage	26
2.2.1.44.7	PidTagBodyContentId	26
2.2.1.44.8	PidTagBodyContentLocation	26
2.2.1.44.9	PidTagHtml	26
2.2.1.45	Contact Linking Properties	27
2.2.1.45.1	PidLidContactLinkEntry	27
2.2.1.45.2	PidLidContacts	27
2.2.1.45.3	PidLidContactLinkName	27
2.2.1.45.4	PidLidContactLinkSearchKey	27
2.2.1.46	Retention and Archive Properties	28
2.2.1.46.1	PidTagArchiveTag	28
2.2.1.46.2	PidTagPolicyTag	28
2.2.1.46.3	PidTagRetentionPeriod	28
2.2.1.46.4	PidTagStartDateEtc	28
2.2.1.46.5	PidTagRetentionDate	29
2.2.1.46.6	PidTagRetentionFlags	29
2.2.1.46.7	PidTagArchivePeriod	30
2.2.1.46.8	PidTagArchiveDate	30
2.2.2	Attachment Object Properties	30
2.2.2.1	General Properties	30
2.2.2.2	PidTagLastModificationTime	30
2.2.2.3	PidTagCreationTime	31
2.2.2.4	PidTagDisplayName	31
2.2.2.5	PidTagAttachSize	31
2.2.2.6	PidTagAttachNumber	31
2.2.2.7	PidTagAttachDataBinary	31
2.2.2.8	PidTagAttachDataObject	31
2.2.2.9	PidTagAttachMethod	31
2.2.2.10	PidTagAttachLongFilename	32
2.2.2.11	PidTagAttachFilename	32
2.2.2.12	PidTagAttachExtension	32
2.2.2.13	PidTagAttachLongPathname	32
2.2.2.14	PidTagAttachPathname	32

2.2.2.15	PidTagAttachTag	32
2.2.2.16	PidTagRenderingPosition	33
2.2.2.17	PidTagAttachRendering	33
2.2.2.18	PidTagAttachFlags	33
2.2.2.19	PidTagAttachTransportName	33
2.2.2.20	PidTagAttachEncoding	33
2.2.2.21	PidTagAttachAdditionalInformation	34
2.2.2.22	PidTagAttachmentLinkId	34
2.2.2.23	PidTagAttachmentFlags	34
2.2.2.24	PidTagAttachmentHidden	34
2.2.2.25	PidTagTextAttachmentCharset	34
2.2.2.26	MIME Properties	34
2.2.3	Message Object ROPS	35
2.2.3.1	RopOpenMessage Buffer Format	35
2.2.3.1.1	Request Buffer	35
2.2.3.1.1.1	CodePageId	36
2.2.3.1.1.2	FolderId	36
2.2.3.1.1.3	OpenModeFlags	36
2.2.3.1.1.4	MessageId	36
2.2.3.1.2	Response Buffer	36
2.2.3.1.2.1	HasNamedProperties	36
2.2.3.1.2.2	SubjectPrefix	36
2.2.3.1.2.3	NormalizedSubject	37
2.2.3.1.2.4	RecipientCount	37
2.2.3.1.2.5	ColumnCount	37
2.2.3.1.2.6	RecipientColumns	37
2.2.3.1.2.7	RowCount	37
2.2.3.1.2.8	RecipientRows	37
2.2.3.2	RopCreateMessage Buffer Format	38
2.2.3.2.1	Request Buffer	38
2.2.3.2.1.1	CodePageId	38
2.2.3.2.1.2	FolderId	38
2.2.3.2.1.3	AssociatedFlag	38
2.2.3.2.2	Response Buffer	38
2.2.3.2.2.1	HasMessageId	38
2.2.3.2.2.2	MessageId	39
2.2.3.3	RopSaveChangesMessage Buffer Format	39
2.2.3.3.1	Request Buffer	39
2.2.3.3.1.1	SaveFlags	39
2.2.3.3.2	Response Buffer	39
2.2.3.3.2.1	MessageId	39
2.2.3.4	RopRemoveAllRecipients Buffer Format	40
2.2.3.4.1	Request Buffer	40
2.2.3.4.1.1	Reserved	40
2.2.3.4.2	Response Buffer	40
2.2.3.5	RopModifyRecipients Buffer Format	40
2.2.3.5.1	Request Buffer	40
2.2.3.5.1.1	ColumnCount	40
2.2.3.5.1.2	RecipientColumns	40
2.2.3.5.1.3	RowCount	41
2.2.3.5.1.4	RecipientRows	41
2.2.3.5.2	Response Buffer	41
2.2.3.6	RopReadRecipients Buffer Format	41

2.2.3.6.1	Request Buffer	41
2.2.3.6.1.1	RowId	41
2.2.3.6.1.2	Reserved	41
2.2.3.6.2	Response Buffer	42
2.2.3.6.2.1	RowCount	42
2.2.3.6.2.2	RecipientRows	42
2.2.3.7	RopReloadCachedInformation Buffer Format	42
2.2.3.7.1	Request Buffer	42
2.2.3.7.1.1	Reserved	42
2.2.3.7.2	Response Buffer	42
2.2.3.8	RopSetMessageStatus Buffer Format	42
2.2.3.8.1	Request Buffer	42
2.2.3.8.1.1	MessageId	43
2.2.3.8.1.2	MessageStatusFlags	43
2.2.3.8.1.3	MessageStatusMask	43
2.2.3.8.2	Response Buffer	43
2.2.3.8.2.1	MessageStatusFlags	43
2.2.3.9	RopGetMessageStatus Buffer Format	43
2.2.3.9.1	Request Buffer	43
2.2.3.9.1.1	MessageId	43
2.2.3.9.2	Response Buffer	43
2.2.3.9.2.1	MessageStatusFlags	44
2.2.3.10	RopSetReadFlags Buffer Format	44
2.2.3.10.1	Request Buffer	44
2.2.3.10.1.1	WantAsynchronous	44
2.2.3.10.1.2	ReadFlags	44
2.2.3.10.1.3	MessageIdCount	45
2.2.3.10.1.4	MessageIds	45
2.2.3.10.2	Response Buffer	45
2.2.3.10.2.1	PartialCompletion	45
2.2.3.11	RopSetMessageReadFlag	45
2.2.3.11.1	Request Buffer	45
2.2.3.11.1.1	ReadFlags	45
2.2.3.11.1.2	ClientData	46
2.2.3.11.2	Response Buffer	46
2.2.3.11.2.1	ReadStatusChanged	46
2.2.3.11.2.2	LogonId	46
2.2.3.11.2.3	ClientData	46
2.2.3.12	RopOpenAttachment	46
2.2.3.12.1	Request Buffer	46
2.2.3.12.1.1	OpenAttachmentFlags	46
2.2.3.12.1.2	AttachmentID	47
2.2.3.12.2	Response Buffer	47
2.2.3.13	RopCreateAttachment	47
2.2.3.13.1	Request Buffer	47
2.2.3.13.2	Response Buffer	47
2.2.3.13.2.1	AttachmentID	47
2.2.3.14	RopDeleteAttachment	47
2.2.3.14.1	Request Buffer	48
2.2.3.14.1.1	AttachmentID	48
2.2.3.14.2	Response Buffer	48
2.2.3.15	RopSaveChangesAttachment Buffer Format	48
2.2.3.15.1	Request Buffer	48

2.2.3.15.1.1	SaveFlags	48
2.2.3.15.2	Response Buffer	48
2.2.3.16	RopOpenEmbeddedMessage Buffer Format.....	48
2.2.3.16.1	Request Buffer	49
2.2.3.16.1.1	CodePageId	49
2.2.3.16.1.2	OpenModeFlags	49
2.2.3.16.2	Response Buffer	49
2.2.3.16.2.1	MessageId	49
2.2.3.16.2.2	HasNamedProperties	49
2.2.3.16.2.3	SubjectPrefix.....	49
2.2.3.16.2.4	NormalizedSubject.....	49
2.2.3.16.2.5	RecipientCount	49
2.2.3.16.2.6	ColumnCount	50
2.2.3.16.2.7	RecipientColumns	50
2.2.3.16.2.8	RowCount	50
2.2.3.16.2.9	RecipientRows	50
2.2.3.17	RopGetAttachmentTable Buffer Format.....	50
2.2.3.17.1	Request Buffer	50
2.2.3.17.1.1	Table Flags	50
2.2.3.17.2	Response Buffer	50
2.2.3.18	RopGetValidAttachments	51
2.2.3.18.1	Request Buffer	51
2.2.3.18.2	Response Buffer	51
3	Protocol Details.....	52
3.1	Client Details.....	52
3.1.1	Abstract Data Model	52
3.1.1.1	Global	52
3.1.1.2	Per Mailbox.....	52
3.1.1.3	Per Message Object	52
3.1.2	Timers	52
3.1.3	Initialization	52
3.1.4	Higher-Layer Triggered Events.....	53
3.1.4.1	Opening a Message Object.....	53
3.1.4.2	Creating a Message Object.....	53
3.1.4.3	Saving Message Object Changes.....	53
3.1.4.4	Removing All Recipients	53
3.1.4.5	Adding, Deleting, and Modifying Recipients	54
3.1.4.6	Reading Recipients	54
3.1.4.7	Reload Message Object Header Info	54
3.1.4.8	Setting Message Status	54
3.1.4.9	Getting Message Status	55
3.1.4.10	Setting Message Object Read State.....	55
3.1.4.11	Opening Attachments	55
3.1.4.12	Creating Attachments.....	55
3.1.4.13	Setting Attachment Object Content	55
3.1.4.14	Saving Attachment Object Changes.....	56
3.1.4.15	Opening an Embedded Message Object.....	56
3.1.4.16	Accessing the Attachments Table	56
3.1.4.17	Creating an Embedded Message	56
3.1.4.18	Saving an Embedded Message.....	56
3.1.4.19	Linking a Contact Object.....	56
3.1.5	Message Processing Events and Sequencing Rules.....	57

3.1.6	Timer Events	57
3.1.7	Other Local Events	57
3.2	Server Details	57
3.2.1	Abstract Data Model	57
3.2.1.1	Global	57
3.2.1.2	Per Mailbox	57
3.2.1.3	Per Message Object	57
3.2.2	Timers	58
3.2.3	Initialization	58
3.2.4	Higher-Layer Triggered Events	58
3.2.5	Message Processing Events and Sequencing Rules	58
3.2.5.1	RopOpenMessage	58
3.2.5.2	RopCreateMessage	59
3.2.5.3	RopSaveChangesMessage	60
3.2.5.4	RopRemoveAllRecipients	61
3.2.5.5	RopModifyRecipients	61
3.2.5.6	RopReadRecipients	62
3.2.5.7	RopReloadCachedInformation	62
3.2.5.8	RopSetMessageStatus	63
3.2.5.9	RopGetMessageStatus	63
3.2.5.10	RopSetReadFlags	63
3.2.5.11	RopSetMessageReadFlag	64
3.2.5.12	RopOpenAttachment	64
3.2.5.13	RopCreateAttachment	65
3.2.5.14	RopSaveChangesAttachment	65
3.2.5.15	RopDeleteAttachment	66
3.2.5.16	RopOpenEmbeddedMessage	66
3.2.5.17	RopGetAttachmentTable	67
3.2.6	Timer Events	67
3.2.7	Other Local Events	67
4	Protocol Examples	68
4.1	Create Message	68
4.1.1	RopCreateMessage Request Buffer	68
4.1.2	RopCreateMessage Response Buffer	68
4.2	Name to Id Mapping	68
4.3	Get Attachment Table	68
4.3.1	RopGetAttachmentTable Request Buffer	69
4.3.2	RopGetAttachmentTable Response Buffer	69
4.4	Insert HTML Embedded Image	69
4.4.1	RopCreateAttachment Request Buffer	69
4.4.2	RopCreateAttachment Response Buffer	69
4.4.3	Setting Properties	70
4.4.4	RopSaveChangesAttachment Request Buffer	70
4.4.5	RopSaveChangesAttachment Response Buffer	71
4.4.6	Releasing Attachment Object	71
4.5	Attach Text File	71
4.5.1	RopCreateAttachment Request Buffer	71
4.5.2	RopCreateAttachment Response Buffer	71
4.5.3	Setting Properties	71
4.5.4	RopSaveChangesAttachment Request Buffer	72
4.5.5	RopSaveChangesAttachment Response Buffer	73
4.5.6	Releasing Attachment Object	73

4.6	Setting Message Properties	73
4.7	Adding Recipients	73
4.7.1	RopModifyRecipients Request Buffer	73
4.7.2	RopModifyRecipients Response Buffer	75
4.8	Save Message	75
4.8.1	RopSaveChangesMessage Request Buffer	75
4.8.2	RopSaveChangesMessage Response Buffer	76
4.9	Releasing Message Object.....	76
5	Security.....	77
5.1	Security Considerations for Implementers.....	77
5.2	Index of Security Parameters	77
6	Appendix A: Product Behavior	78
7	Change Tracking.....	80
8	Index	84

1 Introduction

The Message and Attachment Object Protocol provides the methods used within the server for manipulating **Message objects**.

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 [\[MS-GLOS\]](#):

- 8.3 name**
- ASCII**
- code page**
- Coordinated Universal Time (UTC)**
- flags**
- GUID**
- handle**
- language code identifier (LCID)**
- remote procedure call (RPC)**
- Unicode**

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

- address book**
- Archive Policy**
- Attachment object**
- attachments table**
- blind carbon copy (Bcc) recipient**
- carbon copy (Cc) recipient**
- category**
- character set**
- contact**
- Contact object**
- contents table**
- Embedded Message object**
- EntryID**
- folder associated information (FAI)**
- Folder object**
- header**
- Hypertext Markup Language (HTML)**
- Logon object**
- LogonID**
- LongTermID**
- mailbox**
- message body**
- message class**
- Message object**
- metafile**
- Multipurpose Internet Mail Extensions (MIME)**
- named property**
- non-Unicode**
- Object Linking and Embedding (OLE)**

permission
plain text
primary recipient
property tag
public folder
read receipt
recipient
recipient table
remote operation (ROP)
Retention Policy
retention tag
Rich Text Format (RTF)
ROP request
ROP request buffer
ROP response
ROP response buffer
soft delete
store
Store object
Table object
To recipient
transaction
Transport Neutral Encapsulation Format (TNEF)
Uniform Resource Identifier (URI)

The following terms are specific to this document:

archive tag: An element that contains information about the Archive Policy of a Message object or folder.

clear-signed body: A message body that was promoted from a clear-signed S/MIME message, as described in [MS-OXOSMIME].

header message object: A Message object that contains partial information about a message on a server, such as an identifier for the message, the display names of the recipients and the sender, the subject of the message, and the delivery time of the message. It allows a client to display enough information about a message to let a user choose whether to download the message.

undefined body: A body with no defined content.

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-LCID] Microsoft Corporation, "[Windows Language Code Identifier \(LCID\) Reference](#)", July 2007.

[MS-OXBBODY] Microsoft Corporation, "[Best Body Retrieval Protocol Specification](#)", June 2008.

[MS-OXCADATA] Microsoft Corporation, "[Data Structures](#)", April 2008.

[MS-OXCFCFOLD] Microsoft Corporation, "[Folder Object Protocol Specification](#)", June 2008.

[MS-OXCFCXICS] Microsoft Corporation, "[Bulk Data Transfer Protocol Specification](#)", June 2008.

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

[MS-OXCMSG] Microsoft Corporation, "Message and Attachment Object Protocol Specification", June 2008.

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

[MS-OXCROPS] Microsoft Corporation, "[Remote Operations \(ROP\) List and Encoding Protocol Specification](#)", June 2008.

[MS-OXCSTOR] Microsoft Corporation, "[Store Object Protocol Specification](#)", June 2008.

[MS-OXCSYNC] Microsoft Corporation, "[Mailbox Synchronization Protocol Specification](#)", June 2008.

[MS-OXCCTABL] Microsoft Corporation, "[Table Object Protocol Specification](#)", April 2008.

[MS-OXOABK] Microsoft Corporation, "[Address Book Object Protocol Specification](#)", April 2008.

[MS-OXOCAL] Microsoft Corporation, "[Appointment and Meeting Object Protocol Specification](#)", June 2008.

[MS-OXOCFG] Microsoft Corporation, "[Configuration Information Protocol Specification](#)", June 2008.

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

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

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

[MS-OXRTEFCP] Microsoft Corporation, "[Rich Text Format \(RTF\) Compression Protocol Specification](#)", June 2008.

[MS-OXRTEFEX] Microsoft Corporation, "[Rich Text Format \(RTF\) Extensions Specification](#)", June 2008.

[MS-OXTNEF] Microsoft Corporation, "[Transport Neutral Encapsulation Format \(TNEF\) Data Structure](#)", June 2008.

[MS-WMF] Microsoft Corporation, "[Windows Metafile Format](#)", June 2007.

[RFC2045] Freed, N., and Borenstein, N., "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996, <http://ietf.org/rfc/rfc2045.txt>

[RFC2110] Palme, J., and Hopmann, A., "MIME E-mail Encapsulation of Aggregate Documents, such as HTML (MHTML)", March 1997, <http://www.rfc-editor.org/rfc/rfc2110.txt>

[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>

[RFC2822] Resnick, P., Ed., "Internet Message Format", STD 11, RFC 2822, April 2001, <http://www.ietf.org/rfc/rfc2822.txt>

[RFC3282] Alvestrand, H., "Content Language Headers", RFC 3282, May 2002, <http://www.rfc-editor.org/rfc/rfc3282.txt>

1.2.2 Informative References

[MS-GLOS] Microsoft Corporation, "[Windows Protocols Master Glossary](#)", March 2007.

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

[MS-OXOCNTC] Microsoft Corporation, "[Contact Object Protocol Specification](#)", June 2008.

[MS-OXODOC] Microsoft Corporation, "[Document Object Protocol Specification](#)", June 2008.

[MS-OXOJRN] Microsoft Corporation, "[Journal Object Protocol Specification](#)", June 2008.

[MS-OXONOTE] Microsoft Corporation, "[Note Object Protocol Specification](#)", June 2008.

[MS-OXOPOST] Microsoft Corporation, "[Post Object Protocol Specification](#)", June 2008.

[MS-OXORSS] Microsoft Corporation, "[RSS Object Protocol Specification](#)", June 2008.

[MS-OXOSMIME] Microsoft Corporation, "[S/MIME E-Mail Object Protocol Specification](#)", June 2008.

[MS-OXOSMMS] Microsoft Corporation, "[Short Message Service \(SMS\) and Multimedia Messaging Service \(MMS\) Object Protocol Specification](#)", June 2008.

[MS-OXOTASK] Microsoft Corporation, "[Task-Related Objects Protocol Specification](#)", June 2008.

[MS-OXOUM] Microsoft Corporation, "[Voice Mail and Fax Objects Protocol Specification](#)", June 2008.

1.3 Overview

1.3.1 Messages

Message objects are representations of end-users' data that store properties and are persisted in a folder hierarchy within a message **store**.

1.3.2 FAI Messages

Folder associated information (FAI) messages are a type of message that contains non-user data needed by the client or server. FAI messages are persisted in the same way as Message objects, but cannot be sent.

1.3.3 Message Recipients

Message objects allow clients to associate one or more **recipients** to a message. A recipient is a set of properties that describe where the message is to be delivered. Clients add, remove or modify recipients through the **RecipientRows**.

1.3.4 Message Attachments

An **Attachment object** is used by a client to associate files, **Object Linking and Embedding (OLE)** objects, other messages, or binary data with a particular Message object. Because

Attachment objects are created, maintained, and accessed only in the context of a message, they are considered sub-objects. Operations that affect the location of a Message object also apply to its attachments. Clients retrieve information about attachments in a message via an attachment table, which is a **Table object** as described in [\[MS-OXCTABL\]](#).

1.4 Relationship to Other Protocols

The Message and Attachment Object Protocol relies on folders, tables, and properties, as described in [\[MS-OXCFOLD\]](#), [\[MS-OXOSFLD\]](#), [\[MS-OXCTABL\]](#), and [\[MS-OXCPRPT\]](#), as well as the underlying **remote operations (ROPs)** transport, described in [\[MS-OXCROPS\]](#).

At the time of this publication, the following protocols are known to extend the Message and Attachment Object Protocol.

- Appointment and Meeting Object Protocol, [\[MS-OXOCAL\]](#)
- Contact Object Protocol, [\[MS-OXOCNTC\]](#)
- E-Mail Object Protocol, [\[MS-OXOMSG\]](#)
- Task-Related Objects Protocol, [\[MS-OXOTASK\]](#)
- Note Object Protocol, [\[MS-OXONOTE\]](#)
- Journal Object Protocol, [\[MS-OXOJRNL\]](#)
- RSS Object Protocol, [\[MS-OXORSS\]](#)
- Post Object Protocol, [\[MS-OXOPOST\]](#)
- SMS and MMS Object Protocol, [\[MS-OXOSMMS\]](#)
- Document Object Protocol, [\[MS-OXODOC\]](#)
- S/MIME E-Mail Object Protocol, [\[MS-OXOSMIME\]](#)
- Voice Mail and Fax Objects Protocol, [\[MS-OXOUM\]](#)

1.5 Prerequisites/Preconditions

The Message and Attachment Object Protocol assumes the client has previously logged on to the server and has acquired a **handle** to the **Folder object** upon which it needs to operate. For more information, see [\[MS-OXCFOLD\]](#) and [\[MS-OXCSTOR\]](#).

1.6 Applicability Statement

The Message and Attachment Object Protocol can be used as the basis for different types of personal information messages, like E-mail, **Contacts**, Appointments, Notes, etc.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

A third-party application can create its own set of **named properties** on a Message object as specified in [\[MS-OXCPRPT\]](#). A third-party application can also extend the Message and Attachment

Object Protocol to implement its own object type, by changing the **PidTagMessageClass** property (section [2.2.1.3](#)). For a simple example that extends this protocol to implement an electronic representation of a "Sticky Note", see [\[MS-OXONOTE\]](#).

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

The **ROP request buffers** and **ROP response buffers** specified by this protocol are sent to and respectively are received from the server using the underlying remote operations transport as specified in [\[MS-OXCROPS\]](#).

2.2 Message Syntax

Message objects can be created and modified by clients and servers. Except where noted below, this section defines constraints to which both clients and servers adhere when operating on Message objects.

Clients operate on Message objects using the ROPs as specified in section [2.2.3](#), and the Property and Stream Object Protocol. See [\[MS-OXCPRPT\]](#) section 2.2 for more details.

Unless otherwise specified below, all property constraints specified in [\[MS-OXPROPS\]](#) apply to Message objects. A Message object can also contain other properties defined in [\[MS-OXPROPS\]](#) but these properties have no impact on this protocol.

When a property is referred to as "read-only for the client", the server returns an error and ignores any request to change the value of that property. [<1>](#)

2.2.1 Message Object Properties

All property value types (for example, **PtypBoolean** and **PtypInteger32**) that are specified in the following sub-sections are defined in [\[MS-OXCADATA\]](#) section 2.11.1.

2.2.1.1 General Properties

The following properties exist on all Message objects. These properties are read-only for the client.

PidTagAccess ([\[MS-OXCPRPT\]](#) section 2.2.1.1)

PidTagAccessLevel ([\[MS-OXCPRPT\]](#) section 2.2.1.2)

PidTagChangeKey ([\[MS-OXCPRPT\]](#) section 2.2.1.3)

PidTagCreationTime ([\[MS-OXCPRPT\]](#) section 2.2.1.4)

PidTagLastModificationTime ([\[MS-OXCPRPT\]](#) section 2.2.1.6)

PidTagLastModifierName ([\[MS-OXCPRPT\]](#) section 2.2.1.5)

PidTagObjectType ([\[MS-OXCPRPT\]](#) section 2.2.1.7)

PidTagRecordKey ([\[MS-OXCPRPT\]](#) section 2.2.1.8)

PidTagSearchKey ([\[MS-OXCPRPT\]](#) section 2.2.1.9)

2.2.1.2 PidTagHasAttachments

Type: **PtypBoolean** ([\[MS-OXCADATA\]](#) section 2.11.1)

Indicates whether the Message object contains at least one attachment. This property is read-only for the client.

The server computes this property from the **mfHasAttach** flag of the **PidTagMessageFlags** property ([\[MS-OXPROPS\]](#) section 2.888).

2.2.1.3 PidTagMessageClass

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

Denotes the specific type of the Message object. It determines the set of properties defined for the message, the kind of information the message conveys, and how to handle the message.

All characters in this property MUST be from the **ASCII** characters 0x20 through 0x7F. It MUST NOT end with a period (ASCII character 0x2E), and its length MUST be greater than zero and less than 256 characters. Furthermore, its length SHOULD be fewer than 128 characters, because some operations require extending the value of the **PidTagMessageClass** property ([\[MS-OXPROPS\]](#) section 2.884).

The value of this property is interpreted in groups of characters separated by periods ("."). Each group specifies a type of object. If a server or client does not recognize a **message class**, it reverts to acting on all but the last group, recursively, until a recognized form remains. A message class of "IPM.Note" denotes a standard Message object, and a message class of "Remote.IPM.Note" indicates a **header** Message object.

2.2.1.4 PidTagMessageCodepage

Type: **PtypInteger32**, unsigned

Specifies the **code page** used to encode the **non-Unicode** string properties on this Message object. The Folder object code page is used if this property is set to "0x0000".

2.2.1.5 PidTagMessageLocaleId

Type: **PtypInteger32**, unsigned

Contains the **language code identifier (LCID)** of the end-user who created this message. For more details see [\[MS-LCID\]](#).

2.2.1.6 PidTagMessageFlags

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

Specifies the status of the Message object. Set to zero, or to a bitwise OR of one or more of the values from the following tables.

After the first successful call to the **RopSaveChangesMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.3), as described in section [2.2.3.3](#), these **flags** are read-only for the client.

Name	Value	Description
mfRead	0x00000001	The message is marked as having been read.
mfUnsent	0x00000008	The message is still being composed. This bit is cleared by the server when responding to the RopSubmitMessage ROP ([MS-OXCROPS] section 2.2.7.1) with a success code.

Name	Value	Description
mfResend	0x00000080	The message includes a request for a resend operation with a non-delivery report. For more details, see [MS-OXOMSG] section 3.1.4.5.

These flags are always read-only for the client.

Type	Value	Description
mfUnmodified	0x00000002	The message has not been modified since it was first saved (if unsent) or it was delivered (if sent).
mfSubmitted	0x00000004	The message is marked for sending as a result of a call to the RopSubmitMessage ROP ([MS-OXCROPS] section 2.2.7.1).
mfHasAttach	0x00000010	The message has at least one attachment. This flag corresponds to the message's PidTagHasAttachments property ([MS-OXPROPS] section 2.793).
mfFromMe	0x00000020	The user receiving the message was also the user who sent the message.
mfFAI	0x00000040	The message is an FAI message.
mfNotifyRead	0x00000100	The user who sent the message has requested notification when a recipient first reads it.
mfNotifyUnread	0x00000200	The user who sent the message has requested notification when a recipient deletes it before reading or the Message object expires.
mfEverRead	0x00000400	The message has been read at least once. This flag is set or cleared by the server whenever the mfRead flag is set or cleared. <2>
mfInternet	0x00002000	The incoming message arrived over the Internet and originated either outside the organization or from a source the gateway does not consider trusted.
mfUntrusted	0x00008000	The incoming message arrived over an external link other than X.400 or the Internet. It originated either outside the organization or from a source the gateway does not consider trusted.

The **PidTagMessageFlags** property ([\[MS-OXPROPS\]](#) section 2.888) is also modified using the **RopSetMessageReadFlag** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.11), as described in section [2.2.3.11](#), or the **RopSetReadFlags** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.10), as described in section [2.2.3.10](#).

2.2.1.7 PidTagMessageSize

Type: **PtypInteger32**, unsigned

Contains the size in bytes consumed by the Message object on the server. This property is read-only for the client.

2.2.1.8 PidTagMessageStatus

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

Specifies the status of a message in a **contents table**. Contains a bitwise OR of zero or more of the following values.

Type	Value	Description
msRemoteDownload	0x00001000	The message has been marked for downloading from the remote message store to the local client.
msInConflict	0x00000800	This is a conflict resolve message as specified in [MS-OXCSYNC] . This is a read-only value for the client.
msRemoteDelete	0x00002000	The message has been marked for deletion at the remote message store without downloading to the local client.

In order to set the **PidTagMessageStatus** property ([\[MS-OXPROPS\]](#) section 2.895), the client does not include it in a **RopSetProperties ROP request** ([\[MS-OXCROPS\]](#) section 2.2.8.6), as specified in [\[MS-OXCPRPT\]](#) section 2.2.5. Instead, the client calls the **RopGetMessageStatus ROP** ([\[MS-OXCROPS\]](#) section 2.2.6.9), as specified in section [2.2.3.9](#).

To get the **PidTagMessageStatus** property, the client does not include it in a **RopGetPropertiesSpecific ROP request** ([\[MS-OXCROPS\]](#) section 2.2.8.3). Instead, the client calls the **RopSetMessageStatus ROP** ([\[MS-OXCROPS\]](#) section 2.2.6.8), as specified in section [2.2.3.8](#). Additionally, the **PidTagMessageStatus** property can be set as a column on a contents table. For more details on tables, see [\[MS-OXCTABL\]](#).

2.2.1.9 PidTagSubjectPrefix

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

The **PidTagSubjectPrefix** property ([\[MS-OXPROPS\]](#) section 2.1145) contains the prefix for the subject of the message. Set by the client, but can be an empty string if there is no subject. The sum of the lengths of the **PidTagNormalizedSubject** property ([\[MS-OXPROPS\]](#) section 2.907) and the **PidTagSubjectPrefix** property MUST be less than 254 characters.

The client does not include the **PidTagSubjectPrefix** property in a **RopGetPropertiesSpecific ROP request** ([\[MS-OXCROPS\]](#) section 2.2.8.3). Instead, the client uses the **SubjectPrefix** field from the **RopOpenMessage ROP response buffer** ([\[MS-OXCROPS\]](#) section 2.2.6.1).

2.2.1.10 PidTagNormalizedSubject

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

The **PidTagNormalizedSubject** property ([\[MS-OXPROPS\]](#) section 2.907) contains the normalized subject of the message. Set by the client, but can be an empty string if there is no subject. The sum of the lengths of the **PidTagNormalizedSubject** property and the **PidTagSubjectPrefix** property ([\[MS-OXPROPS\]](#) section 2.1145) MUST be less than 254 characters.

The client does not include the **PidTagNormalizedSubject** property in a **RopGetPropertiesSpecific ROP request** ([\[MS-OXCROPS\]](#) section 2.2.8.3). Instead, the client uses the **NormalizedSubject** field from the **RopOpenMessage ROP response buffer** ([\[MS-OXCROPS\]](#) section 2.2.6.1).

2.2.1.11 PidTagImportance

Type: **PtypInteger32**

Indicates the level of importance assigned by the end user to the Message object; MUST be set to one of the following values.

Value	Description
0x00000000	Low importance.
0x00000001	Normal importance.
0x00000002	High importance.

2.2.1.12 PidTagPriority

Type: **PtypInteger32**

Indicates the client's request for the priority at which the message is to be sent by the messaging system. Is one of the following values.

Value	Description
0x00000001	Urgent.
0x00000000	Normal.
0xFFFFFFFF	Non-Urgent

2.2.1.13 PidTagSensitivity

Type: **PtypInteger32**

Indicates the sender's assessment of the sensitivity of the Message object. Is one of the following.

Value	Description
0x00000000	Normal
0x00000001	Personal
0x00000002	Private
0x00000003	Confidential

2.2.1.14 PidLidSmartNoAttach

Type: **PtypBoolean**

Set to TRUE (0x01) if the Message object has no end-user visible attachments. If this property is unset, a default value of FALSE (0x00) is used.

2.2.1.15 PidLidPrivate

Type: **PtypBoolean**

Set to TRUE (0x01) if the end-user wishes for this Message object to be hidden from other users who have access to the Message object.

2.2.1.16 PidLidSideEffects

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

Controls how a Message object is handled by the client when acting on end-user input. Is set to a bitwise OR of zero or more of the following flags.

Name	Value	Description
seOpenToDelete	0x0001	The client opens the Message object when deleting.
seNoFrame	0x0008	No UI is associated with the Message object.
seCoerceToInbox	0x0010	The client moves the Message object to the Inbox folder when moving or copying to a Folder object with the PidTagContainerClass property ([MS-OXOCAL] section 2.2.11.1) set to "IPF.Note". For more details about the PidTagContainerClass property, see [MS-OXOSFLD] section 2.2.5.
seOpenToCopy	0x0020	The client opens the Message object when copying to another folder.
seOpenToMove	0x0040	The client opens the Message object when moving to another folder.
seOpenForCtxMenu	0x0100	The client opens the Message object when displaying context-sensitive commands, such as a context menu, to the end-user.
seCannotUndoDelete	0x0400	Cannot undo delete operation, MUST NOT be set unless seOpenToDelete is set.
seCannotUndoCopy	0x0800	Cannot undo copy operation, MUST NOT be set unless seOpenToCopy is set.
seCannotUndoMove	0x1000	Cannot undo move operation, MUST NOT be set unless seOpenToMove is set.
seHasScript	0x2000	The Message object contains end-user script.
seOpenToPermDelete	0x4000	The client opens the Message object to permanently delete it.

2.2.1.17 PidNameKeywords

Type: **PtypMultipleString**

Contains keywords or categories for the Message object. The length of each string within the multi-value string is less than 256.

2.2.1.18 PidLidCommonStart

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

The **PidLidCommonStart** property ([\[MS-OXOCAL\]](#) section 2.2.1.32) indicates the start time for the Message object. Is less than or equal to the value of the **PidLidCommonEnd** property ([\[MS-OXOCAL\]](#) section 2.2.1.33). This time is interpreted as **Coordinated Universal Time (UTC)** or local depending on the specification of an extension to this protocol.

2.2.1.19 PidLidCommonEnd

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

The **PidLidCommonEnd** property ([\[MS-OXOCAL\]](#) section 2.2.1.33) indicates the end time for the Message object. MUST be greater than or equal to the value of the **PidLidCommonStart** property ([\[MS-OXOCAL\]](#) section 2.2.1.32). This time is interpreted as **UTC** or local depending on the specification of an extension to this protocol.

2.2.1.20 PidTagAutoForwarded

Type: **PtypBoolean**

Indicates that this message has been automatically generated or automatically forwarded. If this property is unset, then a default value of "0x00" is assumed.

2.2.1.21 PidTagAutoForwardComment

Type: **PtypString**

Contains a comment added by the auto-forwarding agent.

2.2.1.22 PidLidCategories

Type: **PtypMultipleString** ([\[MS-OXCDATA\]](#) section 2.11.1)

The **PidLidCategories** property ([\[MS-OXPROPS\]](#) section 2.49) contains the array of text labels assigned to this Message object.

2.2.1.23 PidLidClassification

Type: **PtypMultipleString** ([\[MS-OXCDATA\]](#) section 2.11.1)

The **PidLidClassification** property ([\[MS-OXPROPS\]](#) section 2.52) contains a list of the classification **categories** to which this Message object has been assigned.

2.2.1.24 PidLidClassificationDescription

Type: **PtypMultipleString** ([\[MS-OXCDATA\]](#) section 2.11.1)

The **PidLidClassificationDescription** property ([\[MS-OXPROPS\]](#) section 2.53) contains a human-readable summary of each of the classification categories included in the **PidLidClassification** property (section [2.2.1.23](#)).

2.2.1.25 PidLidClassified

Type: **PtypBoolean** ([\[MS-OXCDATA\]](#) section 2.11.1)

The **PidLidClassified** property ([\[MS-OXPROPS\]](#) section 2.56) indicates whether the contents of a message are regarded as classified information.

2.2.1.26 PidTagInternetReferences

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

The **PidTagInternetReferences** property ([\[MS-OXPROPS\]](#) section 2.846) contains a list of message IDs specifying the messages to which this reply is related. The format of this property is specified in [\[RFC2822\]](#).

2.2.1.27 PidLidInfoPathFormName

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

The **PidLidInfoPathFormName** property ([\[MS-OXPROPS\]](#) section 2.148) contains the name of the form associated with this message, if one exists. The relationship between this property and the **Content-Class MIME** header is specified in [\[MS-OXCMAIL\]](#) section 2.1.3.2.2 and [\[MS-OXCMAIL\]](#) section 2.2.3.2.15.

2.2.1.28 PidTagMimeSkeleton

Type: **PtypBinary** ([\[MS-OXCDATA\]](#) section 2.11.1)

The **PidTagMimeSkeleton** property ([\[MS-OXPROPS\]](#) section 2.900) contains all the top level MIME message headers, all MIME message body part headers, and body part content that is not already converted to Message object properties, including attachments.

The use of the **PidTagMimeSkeleton** property for converting between MIME messages and Message object format is specified in [\[MS-OXCMAIL\]](#) section 2.3.3.2 and [\[MS-OXCMAIL\]](#) section 2.4.3.1.

2.2.1.29 PidTagTnefCorrelationKey

Type: **PtypBinary** ([\[MS-OXCDATA\]](#) section 2.11.1)

The **PidTagTnefCorrelationKey** property ([\[MS-OXPROPS\]](#) section 2.1160) contains a value that correlates a **Transport Neutral Encapsulation Format (TNEF)** attachment with a message. This property determines whether or not an inbound TNEF file belongs to the message it is attached to. It is used primarily by transport providers and gateways.

2.2.1.30 PidTagAddressBookDisplayNamePrintable

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

The **PidTagAddressBookDisplayNamePrintable** property ([\[MS-OXPROPS\]](#) section 2.591) contains the printable string version of the display name.

2.2.1.31 PidTagCreatorEntryId

Type: **PtypBinary** ([\[MS-OXCDATA\]](#) section 2.11.1)

The **PidTagCreatorEntryId** property ([\[MS-OXPROPS\]](#) section 2.726) specifies the original author of the message according to their **address book EntryID**. The format of an address book EntryID data type is specified in [\[MS-OXCDATA\]](#) section 2.2.5.2.

2.2.1.32 PidTagLastModifierEntryId

Type: **PtypBinary** ([\[MS-OXCDATA\]](#) section 2.11.1)

The **PidTagLastModifierEntryId** property ([\[MS-OXPROPS\]](#) section 2.862) specifies the last user to modify the contents of the message according to their address book EntryID. The format of an address book EntryID data type is specified in [\[MS-OXCDATA\]](#) section 2.2.5.2.

2.2.1.33 PidLidAgingDontAgeMe

Type: **PtypBoolean** ([\[MS-OXCDATA\]](#) section 2.11.1)

The **PidLidAgingDontAgeMe** property ([\[MS-OXPROPS\]](#) section 2.4) specifies whether the message should be automatically archived. "TRUE" if the message will not be automatically archived; otherwise, "FALSE".

2.2.1.34 PidLidCurrentVersion

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

The **PidLidCurrentVersion** property ([\[MS-OXPROPS\]](#) section 2.83) specifies the build number of the client application that sent the message.

2.2.1.35 PidLidCurrentVersionName

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

The **PidLidCurrentVersionName** property ([\[MS-OXPROPS\]](#) section 2.84) specifies the name of the client application that sent the message.

2.2.1.36 PidTagAlternateRecipientAllowed

Type: **PtypBoolean** ([\[MS-OXCDATA\]](#) section 2.11.1)

The **PidTagAlternateRecipientAllowed** property ([\[MS-OXPROPS\]](#) section 2.641) specifies whether the sender permits the message to be auto-forwarded. Set to "TRUE" if auto-forwarding is allowed.

2.2.1.37 PidTagResponsibility

Type: **PtypBoolean** ([\[MS-OXCDATA\]](#) section 2.11.1)

The **PidTagResponsibility** property ([\[MS-OXPROPS\]](#) section 2.1027) specifies whether another mail agent has ensured that the message will be delivered. Set to "TRUE" if another agent has accepted responsibility; otherwise, "FALSE".

2.2.1.38 PidTagRowid

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

The **PidTagRowid** property ([\[MS-OXPROPS\]](#) section 2.1035) contains a unique identifier for a recipient in the message's **recipient table**. This is a temporary identifier that is valid only for the life of the Table object.

2.2.1.39 PidTagHasNamedProperties

Type: **PtypBoolean** ([\[MS-OXCDATA\]](#) section 2.11.1)

The **PidTagHasNamedProperties** property ([\[MS-OXPROPS\]](#) section 2.795) specifies whether a Message object supports named properties.

2.2.1.40 PidTagRecipientOrder

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

The **PidTagRecipientOrder** property ([\[MS-OXPROPS\]](#) section 2.998) specifies the location of the current recipient in the recipient table.

2.2.1.41 PidNameContentBase

Type: **PtypString** ([\[MS-OXCADATA\]](#) section 2.11.1)

The **PidNameContentBase** property ([\[MS-OXPROPS\]](#) section 2.431) specifies the value of the MIME **Content-Base** header, which defines the base **Uniform Resource Identifier (URI)** for resolving relative URLs contained within the **message body**.

2.2.1.42 PidNameAcceptLanguage

Type: **PtypString** ([\[MS-OXCADATA\]](#) section 2.11.1)

The **PidNameAcceptLanguage** property ([\[MS-OXPROPS\]](#) section 2.365) contains the value of the MIME **Accept-Language** header, which defines the natural languages in which the sender prefers to receive a response. The format of this property is specified in section 3 of [\[RFC3282\]](#). The relationship between this property and the **Accept-Language** header is specified in [\[MS-OXCMAIL\]](#) section 2.1.3.2.16 and [\[MS-OXCMAIL\]](#) section 2.2.3.2.11.

2.2.1.43 PidTagPurportedSenderDomain

Type: **PtypString** ([\[MS-OXCADATA\]](#) section 2.11.1)

The **PidTagPurportedSenderDomain** property ([\[MS-OXPROPS\]](#) section 2.973) contains the domain name of the last sender responsible for transmitting the current message.

2.2.1.44 Body Properties

Body properties are a group of related properties valid on any Message object that specify the body text format and contents and conform to the specification in [\[MS-OXBBODY\]](#).

2.2.1.44.1 PidTagBody

Type: **PtypString**

Contains the unformatted text analogous to the text/plain body of [\[RFC2822\]](#).

2.2.1.44.2 PidTagNativeBody

Type: **PtypInteger32** ([\[MS-OXCADATA\]](#) section 2.11.1)

Indicates the best available format for storing the message body. The value of this property is limited to one of the property values shown in the following table.

Meaning	Value
Undefined body	0x00
Plain text body	0x01
Rich Text Format (RTF) compressed body	0x02
HTML body	0x03
Clear-signed body	0x04

2.2.1.44.3 PidTagBodyHtml

Type: **PtypBinary**

Contains the HTML body as specified in [\[RFC2822\]](#) section 2.3.

2.2.1.44.4 PidTagRtfCompressed

Type: **PtypBinary**

Contains a Rich Text Format (RTF) body compressed as specified in [\[MS-OXRTFCP\]](#).

2.2.1.44.5 PidTagRtfInSync

Type: **PtypBoolean** ([\[MS-OXCDATA\]](#) section 2.11.1)

Set to TRUE (0x01) if the RTF body has been synchronized with the contents in the **PidTagBody** property (section [2.2.1.44.1](#)).

2.2.1.44.6 PidTagInternetCodepage

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

Indicates the code page used for the **PidTagBody** property (section [2.2.1.44.1](#)) or the **PidTagBodyHtml** property (section [2.2.1.44.3](#)).

2.2.1.44.7 PidTagBodyContentId

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

The **PidTagBodyContentId** property ([\[MS-OXPROPS\]](#) section 2.684) contains a **GUID** corresponding to the current message body.

This property corresponds to the MIME **Content-ID** header. The relationship between this property and the **Content-ID** header is further specified in [\[MS-OXCMAIL\]](#) section 2.1.3.4.2.3 and [\[MS-OXCMAIL\]](#) section 2.2.3.2.24.

2.2.1.44.8 PidTagBodyContentLocation

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

The **PidTagBodyContentLocation** property ([\[MS-OXPROPS\]](#) section 2.685) contains a globally unique URI that serves as a label for the current message body. The URI can be either absolute or relative.

This property corresponds to the MIME **Content-Location** header. The relationship between this property and the **Content-Location** header is further specified in [\[MS-OXCMAIL\]](#) section 2.1.3.4.2.3 and [\[MS-OXCMAIL\]](#) section 2.2.3.2.26. The format of this property is further specified in [\[RFC2110\]](#).

2.2.1.44.9 PidTagHtml

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

The **PidTagHtml** property ([\[MS-OXPROPS\]](#) section 2.809) contains the message body text in HTML format.

2.2.1.45 Contact Linking Properties

A group of related properties valid on any Message object containing information about the linked **Contact objects**.

2.2.1.45.1 PidLidContactLinkEntry

Type: **PtypBinary**

Contains the list of address book EntryIDs linked to by this Message object.

Size in bytes	Meaning	Notes
4	Address Entry COUNT	
4	Size of this property minus 4 (FieldSize)	
variable	Address book EntryID data	Repeats Address Entry COUNT times. Address book EntryIDs are specified in [MS-OXCDATA] section 2.2.5.2.
0 – 3	Padding to make FieldSize a multiple of 4	Each padding BYTE MUST be 0x00.

2.2.1.45.2 PidLidContacts

Type: **PtypMultipleString** ([\[MS-OXCDATA\]](#) section 2.11.1)

The **PidLidContacts** property (section [2.2.1.45.2](#)) contains the **PidTagDisplayName** property ([\[MS-OXPROPS\]](#) section 2.746) of each address book EntryID referenced in the value of the **PidLidContactLinkEntry** property (section [2.2.1.45.1](#)). Can also include names not referenced in the **PidLidContactLinkEntry** property.

2.2.1.45.3 PidLidContactLinkName

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

Contains the elements of the **PidLidContacts** property (section [2.2.1.45.2](#)), separated by a semicolon and a space ("; ").

2.2.1.45.4 PidLidContactLinkSearchKey

Type: **PtypBinary** ([\[MS-OXCDATA\]](#) section 2.11.1)

Contains the list of search keys for the Contact object linked to by this Message object. Search keys are used to find related objects. Search keys for address book data are further specified by the **PidTagSearchKey** property ([\[MS-OXPROPS\]](#) section 2.1098) in [\[MS-OXOABK\]](#) section 2.2.3.5.

Length in bytes	Meaning	Notes
2	ContactEntryCount	
variable	SearchKey data	Repeats ContactEntryCount times

2.2.1.46 Retention and Archive Properties

Properties that specify information about the **Retention Policy** or **Archive Policy**. These properties are valid on e-mail Message objects. Some of these properties are also valid on folders. The Retention Policy and the Archive Policy are independent features. The server can enable one of these policies, both of these policies, or neither of them. For details about how the Retention Policy and Archive Policy settings are communicated between client and server, see [\[MS-OXOCFG\]](#) section 2.2.4.2.3.

2.2.1.46.1 PidTagArchiveTag

Type: **PtypBinary**

Specifies the GUID of an **archive tag**. The **PidTagArchiveTag** property can be present on both Message objects and folders and can be set by both client and server.

2.2.1.46.2 PidTagPolicyTag

Type: **PtypBinary** ([\[MS-OXCADATA\]](#) section 2.11.1)

The **PidTagPolicyTag** property ([\[MS-OXPROPS\]](#) section 2.956) specifies the GUID of a **retention tag**. The **PidTagPolicyTag** property can be present on both Message objects and folders and can be set by both client and server.

2.2.1.46.3 PidTagRetentionPeriod

Type: **PtypInteger32** ([\[MS-OXCADATA\]](#) section 2.11.1)

The **PidTagRetentionPeriod** property ([\[MS-OXPROPS\]](#) section 2.1030) specifies the number of days that a Message object can be retained. The **PidTagRetentionPeriod** property can be present on both Message objects and folders and can be set by both client and server.

The presence of the **PidTagRetentionPeriod** property on a Message object indicates that the retention tag on that Message object was explicitly applied by the end-user. If the value of **PidTagRetentionPeriod** is 0, the Message object never expires.

When the **PidTagRetentionPeriod** property is present on a folder, it has no special significance; it simply specifies the retention period that corresponds to the retention tag on that folder.

2.2.1.46.4 PidTagStartDateEtc

Type: **PtypBinary** ([\[MS-OXCADATA\]](#) section 2.11.1)

The **PidTagStartDateEtc** property ([\[MS-OXPROPS\]](#) section 2.1131) is a composite property that holds two pieces of information, as follows:

- Default retention period — The first four bytes contain the retention period of the default retention tag. (A default retention tag is applied to a Message object when a regular retention tag is not present on the Message object. The absence of a regular retention tag indicates that the Message object does not have a specific Retention Policy. The application of the default tag is based on the Message object's message class.)
- Start date — The next eight bytes contain the date, in UTC, from which the Message object's age is calculated.

The **PidTagStartDateEtc** property can be present on only Message objects. The **PidTagStartDateEtc** property is read-only for the client and is set by only the server. <3>

2.2.1.46.5 PidTagRetentionDate

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

The **PidTagRetentionDate** property ([MS-OXPROPS] section 2.1028) specifies the date, in UTC, after which a Message object is expired by the server. The **PidTagRetentionDate** property can be present only on Message objects, not on folders. If the property is not present, the Message object never expires. The **PidTagRetentionDate** property can be set by both client and server.

The value of the **PidTagRetentionDate** property is calculated from the values of other properties. The values used in the calculation depend on whether the Message object has a specific Retention Policy. (A Message object will have the default Retention Policy in the absence of a specific Retention Policy.) The explicit method of calculation is as follows:

- When the Message object has a specific Retention Policy:

PidTagRetentionDate = **PidTagMessageDeliveryTime** ([MS-OXPROPS] section 2.886) + **PidTagRetentionPeriod** (section 2.2.1.46.3). If the **PidTagMessageDeliveryTime** property does not exist, the **PidTagCreationTime** property ([MS-OXPROPS] section 2.725) is used.

- When the Message object has the default Retention Policy:

PidTagRetentionDate = **PidTagMessageDeliveryTime** + default retention period. If the **PidTagMessageDeliveryTime** property does not exist, the **PidTagCreationTime** property is used.

2.2.1.46.6 PidTagRetentionFlags

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

Contains flags that specify the status or nature of an item's retention tag or archive tag. The **PidTagRetentionFlags** property can be present on both Message objects and folders and can be set by both client and server.

The value of the **PidTagRetentionFlags** property is a bitwise OR of zero or more of the values from the following table:

Name	Value	Description
ExplicitTag	0x00000001	The retention tag on the folder is explicitly set.
UserOverride	0x00000002	The retention tag was not changed by the end-user.
AutoTag	0x00000004	The retention tag on the Message object is an auto-tag, which is predicted by the system.
PersonalTag	0x00000008	The retention tag on the folder is of a personal type and can be made available to the end-user.
ExplicitArchiveTag	0x00000010	The archive tag on the folder is explicitly set.
KeepInPlace	0x00000020	The Message object remains in place and is not archived.
SystemData	0x00000040	The Message object or folder is system data.

2.2.1.46.7 PidTagArchivePeriod

Type: **PtypInteger32**

Specifies the number of days that a Message object can remain unarchived. The **PidTagArchivePeriod** property can be present on both Message objects and folders and can be set by both client and server.

The presence of the **PidTagArchivePeriod** property on a Message object indicates that the archive tag on that Message object was explicitly applied by the end-user. If the value of **PidTagArchivePeriod** is 0, the Message object is never moved to archive by the server.

When the **PidTagArchivePeriod** property is present on a folder, it has no special significance; it simply specifies the archive period that corresponds to the archive tag on that folder.

2.2.1.46.8 PidTagArchiveDate

Type: **PtypTime**

Specifies the date, in UTC, after which a Message object is moved to archive by the server. The **PidTagArchiveDate** property can be present on only Message objects, not on folders, and can be set by both client and server. If the **PidTagArchiveDate** property is not present, the Message object is never moved to archive by the server.

The value of the **PidTagArchiveDate** property is calculated from the values of other properties as follows:

PidTagArchiveDate = start date + **PidTagArchivePeriod** (section [2.2.1.46.7](#))

The start date is obtained from the last eight bytes of the **PidTagStartDateEtc** property (section [2.2.1.46.4](#)).

2.2.2 Attachment Object Properties

All property value types (for example, **PtypBoolean** and **PtypInteger32**) that are specified in the following sub-sections are defined in [\[MS-OXCDATA\]](#) section 2.11.1.

2.2.2.1 General Properties

The following properties exist on any Attachment object. These properties are set by the server and are read-only for the client.

PidTagAccessLevel ([\[MS-OXCPRPT\]](#) section 2.2.1.2)

PidTagObjectType ([\[MS-OXCPRPT\]](#) section 2.2.1.7)

PidTagRecordKey ([\[MS-OXCPRPT\]](#) section 2.2.1.8)

2.2.2.2 PidTagLastModificationTime

Type: **PtypTime**, in UTC

Indicates the last time the file referenced by the Attachment object was modified, or the last time the Attachment object itself was modified.

2.2.2.3 PidTagCreationTime

Type: **PtypTime**, in UTC

Indicates the time the file referenced by the Attachment object was created, or the time the Attachment object itself was created.

2.2.2.4 PidTagDisplayName

Type: **PtypString**

Contains the name of the attachment as input by the end user. Set to the same value as the **PidTagAttachLongFilename** property (section [2.2.2.13](#)).

2.2.2.5 PidTagAttachSize

Type: **PtypInteger32**, unsigned

Contains the size in bytes consumed by the Attachment object on the server. This property is read-only for the client.

2.2.2.6 PidTagAttachNumber

Type: **PtypInteger32**, unsigned

Identifies the Attachment object within its Message object. MUST be unique among the Attachment objects in a message.

2.2.2.7 PidTagAttachDataBinary

Type: **PtypBinary**

Contains the contents of the file to be attached.

2.2.2.8 PidTagAttachDataObject

Type: **PtypObject**

Contains the binary representation of the Attachment object in an application-specific format.

2.2.2.9 PidTagAttachMethod

Type: **PtypInteger32**

Represents the way the contents of an attachment are accessed. Set to one of the following values.

Name	Value	Description
afNone	0x00000000	The attachment has just been created.
afByValue	0x00000001	The PidTagAttachDataBinary property (section 2.2.2.7) contains the attachment data.
afByReference	0x00000002	The PidTagAttachLongPathname property (section 2.2.2.13) contains a fully qualified path identifying the attachment To recipients with access to a common file server.

Name	Value	Description
afByReferenceOnly	0x00000004	The PidTagAttachLongPathname property contains a fully qualified path identifying the attachment.
afEmbeddedMessage	0x00000005	The attachment is an embedded message that is accessed via the RopOpenEmbeddedMessage ROP ([MS-OXCROPS] section 2.2.6.16).
afStorage	0x00000006	The PidTagAttachDataObject property (section 2.2.2.8) contains data in an application-specific format.

2.2.2.10 PidTagAttachLongFilename

Type: **PtypString**

Contains the full filename and extension of the Attachment object.

2.2.2.11 PidTagAttachFilename

Type: **PtypString**

Contains the [8.3 name](#) of the value of the **PidTagAttachLongFilename** property (section [2.2.2.10](#)).

2.2.2.12 PidTagAttachExtension

Type: **PtypString**

Contains a filename extension that indicates the document type of an attachment.

2.2.2.13 PidTagAttachLongPathname

Type: **PtypString**

Contains the fully qualified path and filename with extension.

2.2.2.14 PidTagAttachPathname

Type: **PtypString**

Contains the 8.3 name of the value of the **PidTagAttachLongPathname** property (section [2.2.2.13](#)).

2.2.2.15 PidTagAttachTag

Type: **PtypBinary**

Contains the identifier information for the application which supplied the Attachment object's data. This property can be left unset; if set, it MUST be one of the following.

Definition	Data	Comments
Transport Neutral Encapsulation Format (TNEF)	{0x2A,86,48,86,F7,14,03,0A,01}	See [MS-OXTNEF]

Definition	Data	Comments
afStorage	{0x2A,86,48,86,F7,14,03,0A,03,02,01}	Data is in an application-specific format.
MIME	{0x2A,86,48,86,F7,14,03,0A,04}	See [MS-OXCMAIL]

2.2.2.16 PidTagRenderingPosition

Type: **PtypInteger32**, unsigned

Represents an offset, in rendered characters, to use when rendering an attachment within the main message text.

The values specify a relative ordering of the rendered attachment in the text. If a message has three **attachments** with **PidTagRenderingPosition** values of 200, 100, and 500, these will be rendered in the same order as if the attachments had the values 2, 1, and 5. A detailed example of this property is provided in [\[MS-OXRTFEX\]](#) section 4.2.

The value 0xFFFFFFFF indicates a hidden attachment that is not to be rendered in the main text.

2.2.2.17 PidTagAttachRendering

Type: **PtypBinary**

Contains a Windows **metafile** as specified in [\[MS-WMF\]](#) for the Attachment object.

2.2.2.18 PidTagAttachFlags

Type: **PtypInteger32**, as a bit field

Indicates which body formats might reference this attachment when rendering data. Contains a bitwise OR of zero or more of the following flags. If this property is absent or its value is 0x00000000, then the attachment is available to be rendered in any format.

Value	Meaning
0x00000001	The Attachment object is not available to be rendered in HTML.
0x00000002	The Attachment object is not available to be rendered in Rich Text Format.
0x00000004	The Attachment object is referenced and rendered within the HTML body of the associated Message object. More details are specified in the PidTagBodyHtml property (section 2.2.1.44.3).

2.2.2.19 PidTagAttachTransportName

Type: **PtypString**

Contains the name of an attachment file, modified so that it can be correlated with TNEF messages, see [\[MS-OXTNEF\]](#).

2.2.2.20 PidTagAttachEncoding

Type: **PtypBinary**

Contains encoding information about the Attachment object. If the attachment is in MacBinary format, then this property is set to "{0x2A,86,48,86,F7,14,03,0B,01}"; otherwise, it is unset. <4> This property is used to indicate that the attachment content, which is the value of the **PidTagAttachDataBinary** property (section 2.2.2.7), MUST be encoded in the MacBinary format, as specified in [MS-OXCMAIL].

2.2.2.21 PidTagAttachAdditionalInformation

Type: **PtypString**

MUST be set to an empty string if the **PidTagAttachEncoding** property (section 2.2.2.20) is unset. If the **PidTagAttachEncoding** property is set, then it MUST be set to a string of the format ":CREA:TYPE", where ":CREA" is the four-letter Macintosh file creator code, and ":TYPE" is a four-letter Macintosh type code.

2.2.2.22 PidTagAttachmentLinkId

Type: **PtypInteger32**

The type of Message object to which this attachment is linked. MUST be "0x00000000", unless overridden by other protocols that extend the Message and Attachment Object Protocol as noted in section 1.4.

2.2.2.23 PidTagAttachmentFlags

Type: **PtypInteger32**

Indicates special handling for this Attachment object. MUST be "0x00000000", unless overridden by other protocols that extend the Message and Attachment Object Protocol as noted in section 1.4

2.2.2.24 PidTagAttachmentHidden

Type: **PtypBoolean**

Set to TRUE (0x01) if this Attachment object is hidden from the end user.

2.2.2.25 PidTagTextAttachmentCharset

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

The **PidTagTextAttachmentCharset** property ([MS-OXPROPS] section 2.1157) specifies the **character set** of messages for messages with a text body. Corresponds to the charset parameter of the **Content-Type** MIME header, as specified in [MS-OXCMAIL] section 2.2.3.4.1.2.

2.2.2.26 MIME Properties

The following properties contain MIME information and can be left unset. For details about MIME specifications, see [RFC2045]. For the specification on mapping these properties, see [MS-OXCMAIL]. The types in the following table are specified in ([MS-OXCADATA] section 2.11.1).

Type	Property	Content
PtypString	PidTagAttachMimeTag ([MS-OXPROPS] section 2.666)	The Content-Type MIME header.

Type	Property	Content
PtypString	PidTagAttachContentId ([MS-OXPROPS] section 2.651)	A content identifier unique to this Message object that matches a corresponding "cid:" Uniform Resource Identifier (URI) scheme reference in the HTML body of the Message object.
PtypString	PidTagAttachContentLocation ([MS-OXPROPS] section 2.652)	A relative or full URI that matches a corresponding reference in the HTML body of the Message object.
PtypString	PidTagAttachContentBase ([MS-OXPROPS] section 2.650)	The base of a relative URI. MUST be set if the PidTagAttachContentLocation property contains a relative URI.
PtypString	PidTagAttachPayloadClass ([MS-OXPROPS] section 2.669)	The class name of an object that can display the contents of the message.
PtypString	PidTagAttachPayloadProviderGuidString ([MS-OXPROPS] section 2.670)	The GUID of the software application that can display the contents of the message.
PtypString	PidNameAttachmentMacContentType ([MS-OXPROPS] section 2.370)	The Content-Type header of the Macintosh attachment.
PtypBinary	PidNameAttachmentMacInfo ([MS-OXPROPS] section 2.371)	The headers and resource fork data associated with the Macintosh attachment.

2.2.3 Message Object ROPS

The following sections specify the format of the ROP request buffers and ROP response buffers specific to the Message and Attachment Object Protocol. Before sending these requests to the server, the client has logged on to the server, and acquired a handle to the Message object or Folder object used in the ROP request.

2.2.3.1 RopOpenMessage Buffer Format

The **RopOpenMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.1) provides access to an existing Message object, which is identified by the message ID (MID), whose structure is specified in [\[MS-OXCADATA\]](#) section 2.2.1.2. The folder containing the Message object is identified by the folder ID (FID), whose structure is specified in [\[MS-OXCADATA\]](#) section 2.2.1.1.

For this ROP, the **InputHandleIndex** references either a **Store object** or a Folder object. If a folder is used, it is not necessary that it is the parent folder, only that it is a folder within the same store. The **OutputHandleIndex** references a Message object.

2.2.3.1.1 Request Buffer

The syntax of the **RopOpenMessage** ROP request buffer ([\[MS-OXCROPS\]](#) section 2.2.6.1) is specified in [\[MS-OXCROPS\]](#).

The fields specified in the following subsections are part of the **RopOpenMessage** ROP request buffer.

2.2.3.1.1.1 CodePageId

2 bytes specifying the code page in which the non-Unicode representation of the strings on this Message object are encoded. A value of 0x0FFF means that the code page of the **Logon object** is used.

2.2.3.1.1.2 FolderId

8 bytes containing the FID of the folder from which the message is to be opened.

2.2.3.1.1.3 OpenModeFlags

1 byte; MUST be one of the following.

Name	Value	Meaning
ReadOnly	0x00	Message will be opened as read-only.
ReadWrite	0x01	Message will be opened for both reading and writing.
BestAccess	0x03	Open for read/write if possible, read-only if not.
OpenSoftDeleted	0x04	Open a soft deleted Message object if available.

2.2.3.1.1.4 MessageId

8 bytes containing the MID for the Message object to open.

2.2.3.1.2 Response Buffer

The syntax of the **RopOpenMessage** ROP response buffer ([\[MS-OXCROPS\]](#) section 2.2.6.1) is specified in the [\[MS-OXCROPS\]](#).

The fields specified in the following sub-sections are part of the **RopOpenMessage** ROP response buffer.

2.2.3.1.2.1 HasNamedProperties

1 byte.

Value	Meaning
0x00	No named properties are included in the ROP response buffer.
Nonzero	Named properties are included.

2.2.3.1.2.2 SubjectPrefix

A **TypedString** structure ([\[MS-OXCADATA\]](#) section 2.11.7) specifying the prefix for the subject of the Message object. Contains the value of the **PidTagSubjectPrefix** property ([\[MS-OXPROPS\]](#) section 2.1146).

2.2.3.1.2.3 NormalizedSubject

A **TypedString** structure ([\[MS-OXCDATA\]](#) section 2.11.7) specifying the normalized subject of the Message object. Contains the value of the **PidTagNormalizedSubject** property ([\[MS-OXPROPS\]](#) section 2.907).

2.2.3.1.2.4 RecipientCount

A 2-byte unsigned **integer** containing the number of recipients associated with the Message object.

2.2.3.1.2.5 ColumnCount

A 2-byte unsigned **integer** containing the number of elements in the **RecipientColumns** field.

2.2.3.1.2.6 RecipientColumns

An array of **PropertyTag** structures ([\[MS-OXCDATA\]](#) section 2.9) with <ColumnCount> elements. Each **PropertyTag** value is valid for a recipient as specified in [\[MS-OXPROPS\]](#).

2.2.3.1.2.7 RowCount

A 1-byte unsigned **integer** containing the number of rows in the **RecipientRows** field. The value MUST be less than or equal to **RecipientCount**.

2.2.3.1.2.8 RecipientRows

An array of **OpenRecipientRow** structures with <RowCount> elements, as specified in [\[MS-OXCROPS\]](#) section 2.2.6.1.2.1.

The RecipientType value specified in [\[MS-OXCROPS\]](#) section 2.2.6.1.2.1 is a bitwise OR of one value from the Types table with zero or more values from the flags table.

Types are as follows.

Value	Description
0x01	Primary recipient.
0x02	Carbon copy (Cc) recipient.
0x03	Bcc recipient.

Flags are as follows.

Value	Description
0x10	When resending a previous failure, this flag indicates that this recipient did not successfully receive the message on the previous attempt.
0x80	When resending a previous failure, this flag indicates that this recipient did successfully receive the message on the previous attempt.

2.2.3.2 RopCreateMessage Buffer Format

The **RopCreateMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.2) is used to create a new Message object.

For this ROP, the **InputHandleIndex** references a folder or Logon object and the **OutputHandleIndex** references a Message object.

2.2.3.2.1 Request Buffer

The syntax of the **RopCreateMessage** ROP request buffer ([\[MS-OXCROPS\]](#) section 2.2.6.2) is specified in the [\[MS-OXCROPS\]](#).

The fields specified in the following sub-sections are part of the **RopCreateMessage** ROP request buffer.

2.2.3.2.1.1 CodePageId

2 bytes specifying the code page that the non-Unicode representation of the strings on this Message object are to be encoded; a value of "0x0FFF" means that the code page of the Logon object is used.

2.2.3.2.1.2 FolderId

8 bytes containing the FID for the Folder object in which the Message object is to be created.

2.2.3.2.1.3 AssociatedFlag

1 byte.

Value	Meaning
0x00	Not an FAI message.
Nonzero	Is an FAI message.

2.2.3.2.2 Response Buffer

The syntax of the **RopCreateMessage** ROP response buffer is specified in [\[MS-OXCROPS\]](#) section 2.2.6.2.

The fields specified in the following sub-sections are part of the **RopCreateMessage** ROP response buffer.

2.2.3.2.2.1 HasMessageId

1 byte.

Value	Meaning
0x00	This is the last byte in the buffer.
Nonzero	MessageId follows beginning with the next byte in the buffer.

2.2.3.2.2 MessageId

8 bytes containing the MID for the newly created Message object.

2.2.3.3 RopSaveChangesMessage Buffer Format

RopSaveChangesMessage commits the changes made to the Message object.

For this ROP, the **ResponseHandleIndex** references the containing Folder object or, for an embedded message, the **Embedded Message object**. The **InputHandleIndex** references a Message object.

2.2.3.3.1 Request Buffer

The syntax of the **RopSaveChangesMessage** ROP request buffer is specified in [\[MS-OXCROPS\]](#) section 2.2.6.3.

The fields specified in the following sub-sections are part of the **RopSaveChangesMessage** ROP request buffer.

2.2.3.3.1.1 SaveFlags

1 byte indicating the server save behavior; MUST be one value from the following table.

SaveType

Name	Value	Description
KeepOpenReadOnly	0x09	The client requests that the server commit the changes. The server either returns an error and leaves the Message object open with unchanged access level, or returns a success code and keeps the Message object open with read-only access. More information about access levels is specified in [MS-OXCPRPT] section 2.2.1.2.
KeepOpenReadWrite	0x0A	The client requests that the server commit the changes. The server either returns an error and leaves the Message object open with unchanged access level, or returns a success code and keeps the Message object open with read/write access.
ForceSave	0x0C	The client requests that the server commit the changes. The server either returns an error and leaves the Message object open with unchanged access level, or returns a success code and keeps the Message object open with read/write access. The ecObjectModified error code is not valid when this flag is set: the server overwrites any changes instead.

2.2.3.3.2 Response Buffer

The syntax of the **RopSaveChangesMessage** ROP response buffer is specified in [\[MS-OXCROPS\]](#) section 2.2.6.3.

The fields specified in the following sub-sections are part of the **RopSaveChangesMessage** ROP response buffer.

2.2.3.3.2.1 MessageId

8 bytes containing the MID for the saved Message object.

2.2.3.4 RopRemoveAllRecipients Buffer Format

The client sends the **RopRemoveAllRecipients** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.4) to delete all recipients from a message.

For this ROP, the **InputHandleIndex** references a Message object.

2.2.3.4.1 Request Buffer

The syntax of the **RopRemoveAllRecipients** ROP request buffer is specified in [\[MS-OXCROPS\]](#) section 2.2.6.4.

The fields specified in the following sub-sections are part of the **RopRemoveAllRecipients** ROP request buffer.

2.2.3.4.1.1 Reserved

4 bytes; unspecified value.

2.2.3.4.2 Response Buffer

The syntax of the **RopRemoveAllRecipients** ROP response buffer is specified in [\[MS-OXCROPS\]](#) section 2.2.6.4.

This protocol adds no additional field information to the **RopRemoveAllRecipients** ROP response buffer.

2.2.3.5 RopModifyRecipients Buffer Format

The **RopModifyRecipients** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.5) modifies recipients associated with the Message object.

For this ROP, the **InputHandleIndex** references a Message object.

2.2.3.5.1 Request Buffer

The syntax of the **RopModifyRecipients** ROP request buffer is specified in [\[MS-OXCROPS\]](#) section 2.2.6.5.

The fields specified in the following sub-sections are part of the **RopModifyRecipients** ROP request buffer.

2.2.3.5.1.1 ColumnCount

2 bytes containing the number of elements in the **RecipientColumns** field. Is greater than or equal to "0x0000" and less than "0x7FEF".

2.2.3.5.1.2 RecipientColumns

An array of **PropertyTag** structures with <ColumnCount> elements. Each element is valid for a recipient as specified in [\[MS-OXPROPS\]](#). The client MUST NOT include **property tags** for any properties that are part of standard property values of the **RecipientRow**, as specified in [\[MS-OXCADATA\]](#) section 2.8.3:

PidTagAddressType ([\[MS-OXOABK\]](#) section 2.2.3.13)

- PidTagDisplayName** ([\[MS-OXPROPS\]](#) section 2.746)
- PidTagEmailAddress** ([\[MS-OXPROPS\]](#) section 2.752)
- PidTagEntryId** ([\[MS-OXPROPS\]](#) section 2.759)
- PidTagInstanceKey** ([\[MS-OXPROPS\]](#) section 2.839)
- PidTagRecipientType** ([\[MS-OXPROPS\]](#) section 2.1006)
- PidTagSearchKey** ([\[MS-OXPROPS\]](#) section 2.1098)
- PidTagSendRichInfo** ([\[MS-OXPROPS\]](#) section 2.1114)
- PidTagTransmittableDisplayName** ([\[MS-OXPROPS\]](#) section 2.1162)

2.2.3.5.1.3 RowCount

2 bytes containing the number of elements in the **RecipientRows** field. Is greater than or equal to "0x0000" and less than "0x7FEF".

2.2.3.5.1.4 RecipientRows

An array of **ModifyRecipientRow** structures ([\[MS-OXCROPS\]](#) section 2.2.6.5.1.1) with <RowCount> elements.

2.2.3.5.2 Response Buffer

The syntax of the **RopModifyRecipients** ROP response buffer is specified in [\[MS-OXCROPS\]](#) section 2.2.6.5.

This protocol adds no additional field information to the **RopModifyRecipients** ROP response buffer.

2.2.3.6 RopReadRecipients Buffer Format

The **RopReadRecipients** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.6) retrieves the recipients associated with the Message object.

For this ROP, the **InputHandleIndex** references a Message object.

2.2.3.6.1 Request Buffer

The syntax of the **RopReadRecipients** ROP request buffer is specified in [\[MS-OXCROPS\]](#) section 2.2.6.6.

The fields specified in the following sub-sections are part of the **RopReadRecipients** ROP request buffer.

2.2.3.6.1.1 RowId

4 bytes containing the starting index for the recipients to be retrieved.

2.2.3.6.1.2 Reserved

2 bytes; MUST be 0x0000.

2.2.3.6.2 Response Buffer

The syntax of the **RopReadRecipients** ROP response buffer is specified in [\[MS-OXCROPS\]](#) section 2.2.6.6.

The fields specified in the following sub-sections are part of the **RopReadRecipients** ROP response buffer.

2.2.3.6.2.1 RowCount

1 byte containing the number of elements in the **RecipientRows** field. Is greater than or equal to "0x00" and less than "0x7FEF".

2.2.3.6.2.2 RecipientRows

An array of **ReadRecipientRow** structures with <RowCount> elements, as specified in [\[MS-OXCROPS\]](#) section 2.2.6.6.2.1.

2.2.3.7 RopReloadCachedInformation Buffer Format

The **RopReloadCachedInformation** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.7) retrieves the same information as **RopOpenMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.1) but operates on an already opened Message object.

For this ROP, the **InputHandleIndex** parameter references a Message object.

2.2.3.7.1 Request Buffer

The syntax of the **RopReloadCachedInformation** ROP request buffer is specified in [\[MS-OXCROPS\]](#) section 2.2.6.7.

The fields specified in the following sub-sections are part of the RopReloadCachedInformation request buffer.

2.2.3.7.1.1 Reserved

2 bytes; MUST be "0x0000".

2.2.3.7.2 Response Buffer

The response buffer for the **RopReloadCachedInformation** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.7) is identical to the response buffer for the **RopOpenMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.1).

2.2.3.8 RopSetMessageStatus Buffer Format

The **RopSetMessageStatus** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.8) sets the **PidTagMessageStatus** property ([\[MS-OXPROPS\]](#) section 2.895) on a message in a folder without the need to open or save the Message object.

For this ROP, the **InputHandleIndex** parameter references a Folder object.

2.2.3.8.1 Request Buffer

The syntax of the **RopSetMessageStatus** ROP request buffer is specified in [\[MS-OXCROPS\]](#) section 2.2.6.8.

The fields specified in the following sub-sections are part of the **RopSetMessageStatus** ROP request buffer.

2.2.3.8.1.1 MessageId

8 bytes containing the MID for the Message object to modify.

2.2.3.8.1.2 MessageStatusFlags

4 bytes containing the **PidTagMessageStatus** property ([\[MS-OXPROPS\]](#) section 2.895).

2.2.3.8.1.3 MessageStatusMask

4 bytes indicating which status flags are to be set and which are to be cleared. Contains a bitwise OR of zero or more values from the table in section [2.2.1.8](#). The server sets all flags that are set in both **MessageStatusMask** and **MessageStatusFlags**, and clears all flags that are set in **MessageStatusMask** but clear in **MessageStatusFlags**.

2.2.3.8.2 Response Buffer

The syntax of the **RopSetMessageStatus** ROP response buffer is specified in [\[MS-OXCROPS\]](#) section 2.2.6.8.

The fields specified in the following sub-sections are part of the **RopSetMessageStatus** ROP response buffer.

2.2.3.8.2.1 MessageStatusFlags

4 bytes indicating the status flags that were set on the Message object prior to processing this request. MUST contain a bitwise OR of zero or more values from the table in section [2.2.1.8](#).

2.2.3.9 RopGetMessageStatus Buffer Format

The **RopGetMessageStatus** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.9) gets the message status of a message in a folder.

For this ROP, the **InputHandleIndex** parameter references a Folder object.

2.2.3.9.1 Request Buffer

The syntax of the **RopGetMessageStatus** ROP request buffer is specified in [\[MS-OXCROPS\]](#) section 2.2.6.9.

The fields specified in the following sub-sections are part of the **RopGetMessageStatus** ROP request buffer.

2.2.3.9.1.1 MessageId

8 bytes containing the MID for the Message object in which to operate.

2.2.3.9.2 Response Buffer

The syntax of the **RopGetMessageStatus** ROP response buffer is specified in [\[MS-OXCROPS\]](#) section 2.2.6.9.

The fields specified in the following sub-sections are part of the **RopSetMessageStatus** ROP response buffer.

2.2.3.9.2.1 MessageStatusFlags

4 bytes indicating the status of the Message object. Contains a bitwise OR of zero or more values from the table in section [2.2.1.8](#).

2.2.3.10 RopSetReadFlags Buffer Format

The **RopSetReadFlags** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.10) changes the state of the **PidTagMessageFlags** property ([\[MS-OXPROPS\]](#) section 2.888) on one or more Message objects within a Folder object. It also triggers the sending of **read receipts**, as specified in [\[MS-OXOMSG\]](#).

For this ROP the **InputHandleIndex** parameter references a Folder object.

2.2.3.10.1 Request Buffer

The syntax of the **RopSetReadFlags** ROP request buffer is specified in [\[MS-OXCROPS\]](#) section 2.2.6.10.

The fields specified in the following sub-sections are part of the **RopSetReadFlags** ROP request buffer.

2.2.3.10.1.1 WantAsynchronous

Note Some of the information in this section is subject to change because it applies to a preliminary implementation of the protocol or structure. For information about specific differences between versions, see the behavior notes that are provided in the Product Behavior appendix.

1 byte indicating whether client is prepared for the **RopSetReadFlags** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.10) to be processed asynchronously with status reported via the **RopProgress** ROP ([\[MS-OXCROPS\]](#) section 2.2.8.13).

2.2.3.10.1.2 ReadFlags

1 byte containing a bitwise OR of zero or more values from the following table. The server modifies bits on the **PidTagMessageFlags** property ([\[MS-OXCMSG\]](#) section 2.2.1.6). The flags, **rfGenerateReceiptOnly**, **rfSuppressReceipt**, and **rfClearReadFlag**, (**rfClearNotifyRead** OR **rfClearNotifyUnread**), are mutually exclusive.

Name	Value	Description
rfDefault	0x00	The server sets the read flag and sends the receipt.
rfSuppressReceipt	0x01	The user requests that any pending read report be canceled; server sets mfRead bit.
rfReserved	0x0A	Ignored by the server.
rfClearReadFlag	0x04	Server clears the mfRead bit; client MUST include rfSuppressReceipt with this flag.
rfGenerateReceiptOnly	0x10	The server sends a read report if one is pending, but does not change the mfRead bit.

Name	Value	Description
rfClearNotifyRead	0x20	The server clears the mfNotifyRead bit, but does not send a read report.
rfClearNotifyUnread	0x40	The server clears the mfNotifyUnread bit, but does not send a non-read report.

2.2.3.10.1.3 MessageIdCount

2 bytes containing the number of elements in the **MessageIds** field.

2.2.3.10.1.4 MessageIds

An array of MIDs with **MessageIdCount** elements.

2.2.3.10.2 Response Buffer

The syntax of the **RopSetReadFlags** ROP response buffer is specified in [\[MS-OXCROPS\]](#) section 2.2.6.10.

The fields specified in the following sub-sections are part of the **RopSetReadFlags** ROP response buffer.

2.2.3.10.2.1 PartialCompletion

1 byte Boolean flag indicating the server was unable to modify one or more of the Message objects represented in the **MessageIds** field.

2.2.3.11 RopSetMessageReadFlag

The **RopSetMessageReadFlag** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.11) changes the state of the **PidTagMessageFlags** property ([\[MS-OXPROPS\]](#) section 2.888) for the Message object. It also triggers the sending of read receipts, as specified in [\[MS-OXOMSG\]](#).

In this section, "in **public folder** mode" means that the logon associated with the **LogonID** from the request was created with the Private flag unset ([\[MS-OXCSTOR\]](#)).

For this ROP, the **ResponseHandleIndex** parameter references a Folder object and the **InputHandleIndex** parameter references a Message object.

2.2.3.11.1 Request Buffer

The syntax of the **RopSetMessageReadFlag** ROP request buffer is specified in ([\[MS-OXCROPS\]](#) section 2.2.6.11).

The fields specified in the following sub-sections are part of the **RopSetMessageReadFlag** ROP request buffer.

2.2.3.11.1.1 ReadFlags

1 byte containing a bitwise OR of one or more values from the table in section [2.2.3.10.1.2](#).

2.2.3.11.1.2 ClientData

The *ClientData* parameter is a 24 byte **LongTermID**, as specified in [\[MS-OXCDATA\]](#) section 2.2.1.3.1, that represents the message read when in public folder mode; 0 bytes otherwise.

2.2.3.11.2 Response Buffer

The syntax of the **RopSetMessageReadFlag** ROP response buffer is specified in ([\[MS-OXCROPS\]](#) section 2.2.6.11).

The fields specified in the following sub-sections are part of the **RopSetMessageReadFlag** ROP response buffer.

2.2.3.11.2.1 ReadStatusChanged

1 byte.

Value	Description
0x00	The read status on the Message object was unchanged or the logon is not in public folder mode.
Nonzero	The read status on the Message object changed and the logon is in public folder mode.

2.2.3.11.2.2 LogonId

1 byte; containing the LogonID from the request when the value in **ReadStatusChanged** is nonzero; 0 bytes otherwise.

2.2.3.11.2.3 ClientData

24 bytes; containing the **ClientData** from the request when the value in **ReadStatusChanged** is nonzero; 0 bytes otherwise.

2.2.3.12 RopOpenAttachment

The **RopOpenAttachment** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.12) opens an Attachment object stored on the Message object.

For this ROP, the **InputHandleIndex** parameter references a Message object and the **OutputHandleIndex** parameter references an Attachment object.

2.2.3.12.1 Request Buffer

The syntax of the **RopOpenAttachment** ROP request buffer is specified in [\[MS-OXCROPS\]](#) section 2.2.6.12.

The fields specified in the following sub-sections are part of the **RopOpenAttachment** ROP request buffer.

2.2.3.12.1.1 OpenAttachmentFlags

1 byte containing one of the following values.

Name	Value	Meaning
ReadOnly	0x00	Message will be opened as read-only.
ReadWrite	0x01	Message will be opened for both reading and writing.
BestAccess	0x03	Open for read/write if possible, read-only if not.

2.2.3.12.1.2 AttachmentID

4 bytes containing the ID of the Attachment object to be opened. See the **PidTagAttachNumber** property (section [2.2.2.6](#)).

2.2.3.12.2 Response Buffer

The syntax of the **RopOpenAttachment** ROP response buffer is specified in [\[MS-OXCROPS\]](#) section 2.2.6.12.

This protocol adds no additional field information to the **RopOpenAttachment** ROP response buffer.

2.2.3.13 RopCreateAttachment

The **RopCreateAttachment** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.13) creates a new Attachment object on the Message object.

For this ROP, the **InputHandleIndex** parameter references a Message object and the **OutputHandleIndex** parameter references an Attachment object.

2.2.3.13.1 Request Buffer

The syntax of the **RopCreateAttachment** ROP request buffer is specified in [\[MS-OXCROPS\]](#) section 2.2.6.13.

This protocol adds no additional field information to the **RopCreateAttachment** ROP request buffer.

2.2.3.13.2 Response Buffer

The syntax of the **RopCreateAttachment** ROP response buffer is specified in [\[MS-OXCROPS\]](#) section 2.2.6.13.

The fields specified in the following sub-sections are part of the **RopCreateAttachment** ROP response buffer.

2.2.3.13.2.1 AttachmentID

4 bytes containing the ID for the Attachment object that was created.

2.2.3.14 RopDeleteAttachment

The **RopDeleteAttachment** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.14) deletes an existing Attachment object from the Message object.

For this ROP, the **InputHandleIndex** parameter references a Message object.

2.2.3.14.1 Request Buffer

The syntax of the **RopDeleteAttachment** ROP request buffer is specified in [\[MS-OXCROPS\]](#) section 2.2.6.14.

The fields specified in the following sub-sections are part of the **RopDeleteAttachment** ROP request buffer.

2.2.3.14.1.1 AttachmentID

4 bytes containing the ID of the Attachment object to be deleted.

2.2.3.14.2 Response Buffer

The syntax of the **RopDeleteAttachment** ROP response buffer is specified in [\[MS-OXCROPS\]](#) section 2.2.6.14.

This protocol adds no additional field information to the **RopDeleteAttachment ROP** response buffer.

2.2.3.15 RopSaveChangesAttachment Buffer Format

The **RopSaveChangesAttachment** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.15) commits the changes made to the Attachment object.

For this ROP, the **ResponseHandleIndex** parameter references the containing Message object and **InputHandleIndex** parameter references an Attachment object.

If pending changes include changes to read-only properties, the server MAY<5> return an error.

2.2.3.15.1 Request Buffer

The syntax of the **RopSaveChangesAttachment** ROP request buffer is specified in [\[MS-OXCROPS\]](#) section 2.2.6.15.

The fields specified in the following sub-sections are part of the **RopSaveChangesAttachment** ROP request buffer.

2.2.3.15.1.1 SaveFlags

See section [2.2.3.3.1.1](#).

2.2.3.15.2 Response Buffer

The syntax of the **RopSaveChangesAttachment** response buffer is specified in [\[MS-OXCROPS\]](#) section 2.2.6.15.

This protocol adds no additional field information to the **RopSaveChangesAttachment** response buffer.

2.2.3.16 RopOpenEmbeddedMessage Buffer Format

The **RopOpenEmbeddedMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.16) retrieves a handle to a Message object from the given Attachment object.

For this ROP, the **InputHandleIndex** parameter references an Attachment object and the **OutputHandleIndex** parameter references a Message object.

2.2.3.16.1 Request Buffer

The syntax of the **RopOpenEmbeddedMessage** ROP request buffer is specified in [\[MS-OXCROPS\]](#) section 2.2.6.16. The fields specified in the following sub-sections are part of the **RopOpenEmbeddedMessage** ROP request buffer.

2.2.3.16.1.1 CodePageId

2 bytes specifying the code page in which the non-Unicode representation of the strings on this Message object MUST be encoded.

2.2.3.16.1.2 OpenModeFlags

1 byte.

Name	Value	Meaning
ReadOnly	0x00	Message will be opened as read-only.
ReadWrite	0x01	Message will be opened for both reading and writing.
Create	0x02	Create the attachment if it does not already exist and open the message for both reading and writing.

2.2.3.16.2 Response Buffer

The syntax of the **RopOpenEmbeddedMessage** ROP response buffer is specified in [\[MS-OXCROPS\]](#) section 2.2.6.16.

The fields specified in the following sub-sections are part of the **RopOpenEmbeddedMessage** ROP response buffer.

2.2.3.16.2.1 MessageId

8 bytes containing the MID for the Message object.

2.2.3.16.2.2 HasNamedProperties

See section [2.2.3.1.2.1](#).

2.2.3.16.2.3 SubjectPrefix

See section [2.2.3.1.2.2](#).

2.2.3.16.2.4 NormalizedSubject

See section [2.2.3.1.2.3](#).

2.2.3.16.2.5 RecipientCount

See section [2.2.3.1.2.4](#).

2.2.3.16.2.6 ColumnCount

See section [2.2.3.1.2.5](#).

2.2.3.16.2.7 RecipientColumns

See section [2.2.3.1.2.6](#).

2.2.3.16.2.8 RowCount

See section [2.2.3.1.2.7](#).

2.2.3.16.2.9 RecipientRows

The format of this field is specified in section [2.2.3.1.2.8](#).

The presence of data in this field is indeterminate, even when the embedded message exists.

2.2.3.17 RopGetAttachmentTable Buffer Format

The **RopGetAttachmentTable** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.17) retrieves a handle to a table object that represents the attachments stored on the Message object. See [\[MS-OXCTABL\]](#) for more details on table objects.

For this ROP, the **InputHandleIndex** parameter references a Message object and the **OutputHandleIndex** parameter references a table object.

2.2.3.17.1 Request Buffer

The syntax of the **RopGetAttachmentTable** ROP request buffer is specified in [\[MS-OXCROPS\]](#) section 2.2.6.17.

The fields specified in the following sub-sections are part of the **RopGetAttachmentTable** ROP request buffer.

2.2.3.17.1.1 Table Flags

1 byte.

Name	Value	Description
Standard	0x00	Open the table.
Unicode	0x40	Open the table. Also requests that the columns containing string data be returned in Unicode format.

2.2.3.17.2 Response Buffer

The syntax of the **RopGetAttachmentTable** ROP response buffer is specified in [\[MS-OXCROPS\]](#) section 2.2.6.17.

This protocol adds no additional field information to the **RopGetAttachmentTable** ROP response buffer.

2.2.3.18 RopGetValidAttachments

RopGetValidAttachments gets the valid attachment identifiers of a message. [<6>](#)

2.2.3.18.1 Request Buffer

The request buffer for **RopGetValidAttachments** is specified in [\[MS-OXCROPS\]](#) section 2.2.6.18.1.

2.2.3.18.2 Response Buffer

The response buffer for **RopGetValidAttachments** is specified in [\[MS-OXCROPS\]](#) section 2.2.6.18.2.

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.

The following abstract types are defined in this section:

Global

Mailbox

Message Object

3.1.1.1 Global

The following abstract data elements are defined as common to both client and server:

Handle: Represents an open connection by a client to a server object.

3.1.1.2 Per Mailbox

Mailboxes are represented by the **Mailbox** abstract object type. The following abstract objects are maintained for each **Mailbox**:

Mailbox.MessageObject: An abstract representation of a **Message object**.

3.1.1.3 Per Message Object

A **Message object** is represented by the **MessageObject** abstract data type. The following abstract data elements are maintained for each **MessageObject**:

MessageObject.ReadState: A state specifying whether the message has been read by a user, as specified in section [3.1.4.10](#). The possible state values are as follows:

- **Unsent:** The message has not been sent.
- **Read:** The message has been read.
- **Resend:** The message has been marked for resending.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

3.1.4.1 Opening a Message Object

A client obtains a handle to an existing Message object by sending a **RopOpenMessage** ROP request ([MS-OXCROPS] section 2.2.6.1). In order to send this request, the client first obtains the MID for the Message object to be opened, and either the FID or the LogonID. The MID is accessible from the contents table of the Folder Object that contains the Message object by including the **PidTagMid** property ([MS-OXPROPS] section 2.898) in a **RopSetColumns** ROP request ([MS-OXCROPS] section 2.2.5.1). For more details, see [MS-OXCTABL].

To open a soft deleted Message object, the client MUST include **OpenSoftDeleted** in the **OpenModeFlag** field.

When the client receives the response buffer, it caches the data from the **NormalizedSubject** and **SubjectPrefix** fields; it also updates the cache when issuing a **RopSetProperties** ROP request ([MS-OXCROPS] section 2.2.8.6) for the **PidTagNormalizedSubject** property ([MS-OXPROPS] section 2.907) and the **PidTagSubjectPrefix** property ([MS-OXPROPS] section 2.1146) and uses the cached values.

The client uses the opened Message object in subsequent ROPs; it MUST eventually send a **RopRelease** ROP request ([MS-OXCROPS] section 2.2.15.3) on the Message object, and after doing so, MUST NOT use the Message object for any subsequent ROPs.

The client is responsible for maintaining the privacy of the properties on the Message object when the **PidLidPrivate** property (section 2.2.1.15) is set to "0x01".

3.1.4.2 Creating a Message Object

A client creates a new Message object by sending a **RopCreateMessage** ROP request ([MS-OXCROPS] section 2.2.6.2). The client sends a **RopSaveChangesMessage** ROP request ([MS-OXCROPS] section 2.2.6.3) to commit the new Message object, and uses the opened Message object in subsequent ROPs. It MUST eventually send a **RopRelease** ROP request ([MS-OXCROPS] section 2.2.15.3) on the Message object and after doing so MUST NOT use the Message object for any subsequent ROPs.

3.1.4.3 Saving Message Object Changes

A client saves all the changes to a Message object by sending a **RopSaveChangesMessage** ROP request ([MS-OXCROPS] section 2.2.6.3).

The client controls the access level of the Message object after saving changes by setting the proper flags as specified in section 2.2.3.3.1.1.

3.1.4.4 Removing All Recipients

A client clears all recipients from a Message object by sending a **RopRemoveAllRecipients** ROP request ([MS-OXCROPS] section 2.2.6.4).

The client sends a **RopSaveChangesMessage** ROP ([MS-OXCROPS] section 2.2.6.3) for the Message object associated with the removed recipients in order to commit the changes.

3.1.4.5 Adding, Deleting, and Modifying Recipients

A client modifies recipients of the Message object by sending a **RopModifyRecipients** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.5).

To modify an existing recipient the client sets the **RowId** field to the **RowId** of the recipient to be modified and sets all of the **ModifyRecipientRow** data to the desired values for that recipient, including any additional property information for the recipients. Additional property information is set by adding **PropertyTag** values to the **RecipientColumns** field and including the property values in the **RecipientProperties** field.

To delete an existing recipient the client sets the **RowId** field to the **RowId** of the recipient to be deleted and sets the **RecipientRowSize** field to "0x0000".

To add a new recipient the client sets the **RowId** field to a value greater than the largest **RowId** for any recipient that already exists on the Message object. The client sets all of the **ModifyRecipientRow** data to the desired values for that recipient, including any additional property information.

The client sends a **RopSaveChangesMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.3) for the Message object associated with the added recipients in order to commit the changes.

3.1.4.6 Reading Recipients

A client retrieves a list of all recipients on the Message object by sending a **RopReadRecipients** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.6).

If the COUNT of recipients and the COUNT of recipient rows in the **RopOpenMessage** ROP response buffer ([\[MS-OXCROPS\]](#) section 2.2.6.1) are the same, then the client can use the **RecipientRow** information from the **RopOpenMessage** ROP instead of sending a **RopReadRecipients** ROP request. If the counts are not equal, the client MUST issue a series of **RopReadRecipients** ROP requests to retrieve all the recipients associated with the Message object.

A client accesses the information for all recipients in the message by setting the **RowId** field to "0x00000000", and then iteratively sending **RopReadRecipients** ROP requests with an increasing **RowId** value to obtain the recipients that did not fit in the previous request.

3.1.4.7 Reload Message Object Header Info

A client retrieves the current state of the data returned in a **RopOpenMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.1) by sending a **RopReloadCachedInformation** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.7).

3.1.4.8 Setting Message Status

A client changes the status on a **header message object** (that is, marks or unmarks it for download or delete) by sending a **RopSetMessageStatus** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.8).

To modify the status of a header Message object:

1. Obtain the message's MID, see section [3.1.4.1](#).
2. Send the **RopSetMessageStatus** ROP request, setting the mask and status appropriately.

3.1.4.9 Getting Message Status

A client checks the status of a header Message object by sending a **RopGetMessageStatus** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.9).

To retrieve the status of a header message object:

1. Obtain the message's MID, see section [3.1.4.1](#).
2. Sent the **RopGetMessageStatus** ROP request; if the request succeeds then the value of the header message object's **PidTagMessageStatus** property ([\[MS-OXPROPS\]](#) section 2.895) is returned in the response buffer.

3.1.4.10 Setting Message Object Read State

A client marks one or more Message objects as read or unread without opening the Message objects by sending a **RopSetReadFlags** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.10). The client obtains a list of MIDs using a contents table; see section [3.1.4.1](#), and uses the list of MIDs in the **RopSetReadFlags** ROP request.

When a higher layer wants to mark or unmark a single opened Message object as read, the client sends a **RopSetMessageReadFlag** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.11).

The client controls whether the Message object is marked as read or unread, as well as the sending of read receipts by setting the appropriate flags as specified in section [2.2.3.10.1.2](#).

3.1.4.11 Opening Attachments

A client opens and manipulates an existing Attachment object to a Message object by sending a **RopOpenAttachment** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.12).

The client MUST use a valid AttachmentID (see section [2.2.3.12.1.2](#)) when requesting to open an attachment. See section [3.1.4.16](#).

The client uses the opened Attachment object in subsequent ROPs. It eventually sends a **RopRelease** ROP request ([\[MS-OXCROPS\]](#) section 2.2.15.3) on the Attachment object and after doing so, MUST NOT use the Attachment object for any subsequent ROPs.

3.1.4.12 Creating Attachments

A client adds a new Attachment object to a Message object by sending a **RopCreateAttachment** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.13).

The client sends a **RopSaveChangesAttachment** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.15) to commit the new Attachment object, and uses the newly created Attachment object in subsequent ROPs. The client eventually sends a **RopRelease** ROP request ([\[MS-OXCROPS\]](#) section 2.2.15.3) on the Attachment object and after doing so, MUST NOT use the Attachment object for any subsequent ROPs.

The client sends a **RopSaveChangesMessage** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.3) to commit the Attachment object change to the Message object.

3.1.4.13 Setting Attachment Object Content

A client adds the contents of a file to an Attachment object by sending a **RopSetProperties** ROP request ([\[MS-OXCROPS\]](#) section 2.2.8.6) as specified in [\[MS-OXCPRPT\]](#) section 2.2.5.

Depending on the type of Attachment object the higher layer intends to use, the client sets the appropriate value for the **PidTagAttachMethod** property as specified in section [2.2.2.9](#).

The client sends a **RopSaveChangesAttachment** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.15) to commit the change to the Attachment object and a **RopSaveChangesMessage** ROP request to commit the Attachment object change to the Message object.

3.1.4.14 Saving Attachment Object Changes

A client saves changes to an Attachment object by sending a **RopSaveChangesAttachment** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.15). It sends a **RopSaveChangesMessage** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.3) to commit the Attachment object changes to the Message object.

3.1.4.15 Opening an Embedded Message Object

A client opens an existing Attachment object and manipulates it as if it were a Message object, by sending a **RopOpenEmbeddedMessage** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.16).

The client uses the opened Message object in subsequent ROPs; it eventually sends a **RopRelease** ROP request ([\[MS-OXCROPS\]](#) section 2.2.15.3) on the Message object and after doing so, MUST NOT use the Message object for any subsequent ROPs.

3.1.4.16 Accessing the Attachments Table

A client retrieves information about all Attachment objects associated with a Message object without opening each Attachment object by sending a **RopGetAttachmentTable** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.17).

The server returns a table of properties for each Attachment object associated with the Message object as specified in [\[MS-OXCTABL\]](#). To retrieve the **AttachmentID**, the client includes the **PidTagAttachNumber** property (section [2.2.2.6](#)) when sending a **RopSetColumns** ROP request ([\[MS-OXCROPS\]](#) section 2.2.5.1).

3.1.4.17 Creating an Embedded Message

A client creates an embedded message by sending a **RopCreateAttachment** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.13) to create an attachment on a message. The client uses the **RopSetProperties** ROP ([\[MS-OXCROPS\]](#) section 2.2.8.6) to set the value of the **PidTagAttachMethod** property (section [2.2.2.9](#)) to **afEmbeddedMessage**. Finally the client sends a **RopOpenEmbeddedMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.16) on the attachment to get a Message object handle.

3.1.4.18 Saving an Embedded Message

A client saves an embedded message by sending a **RopSaveChangesMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.3) on the embedded message. Then the client sends a **RopSaveChangesAttachment** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.15) on the attachment from which the embedded message was opened. Finally, the client sends a **RopSaveChangesMessage** ROP on the enclosing message.

3.1.4.19 Linking a Contact Object

To link a Contact object with another Message object, the client sets the following properties. For more details, see section [2.2.1.45](#).

- **PidLidContactLinkEntry** (section [2.2.1.45.1](#))
- **PidLidContactLinkName** (section [2.2.1.45.3](#))
- **PidLidContactLinkSearchKey** (section [2.2.1.45.4](#))
- **PidLidContacts** (section [2.2.1.45.2](#))

3.1.5 Message Processing Events and Sequencing Rules

None.

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.

The following abstract data types are defined in this section.

Global

Mailbox

Message Object

3.2.1.1 Global

The following abstract data elements are defined as common to both client and server.

Handle: Represents an open connection by a client to a server object.

3.2.1.2 Per Mailbox

Mailboxes are represented by the **Mailbox** abstract data type. The following abstract data elements are maintained for each **Mailbox**.

Mailbox.MessageObject: An abstract representation of a **Message object**.

3.2.1.3 Per Message Object

A Message object is represented by the **MessageObject** abstract data type. The following abstract data elements are maintained for each **MessageObject**.

MessageObject.AttachmentTable: The set of attachments associated with a message.

MessageObject.Transactions: A record of the open **Handles** against a **MessageObject**. Each **Handle** is given its own transaction. If a change to a **MessageObject** is committed on one **Handle**, the server prevents changes from being saved against other open **Handles** on the same **MessageObject**, as specified in section [3.2.5.3](#).

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 RopOpenMessage

Provides access to existing Message objects stored by the server. The Message object returned by this ROP is used in subsequent ROPs, such as a **RopGetPropertiesSpecific** ROP request ([\[MS-OXCROPS\]](#) section 2.2.8.3). See specific ROPs for information of which operate on Message objects, as specified in [\[MS-OXCROPS\]](#).

When the server receives multiple requests to open the same Message object, it returns a different handle and maintains a separate **transaction** for each.

A **RopOpenMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.1) MUST NOT succeed if a Message object with the specified ID does not exist or the client has insufficient access rights to view the Message object.

If the **OpenModeFlag** field includes the **OpenSoftDeleted** flag, then the **RopOpenMessage** ROP provides access to all Message objects, including soft deleted Message objects. If **OpenSoftDeleted** is not included, then the server MUST NOT provide access to soft deleted Message objects.

The response field **RecipientCount** indicates the current number of recipients in the message. In addition, the server returns data for as many recipients as will fit in the response buffer, in order of **RowId** value. The data for each recipient is encoded as an **OpenRecipientRow** structure in the **RecipientRows** field. The response field **RowCount** indicates how many recipients are present in **RecipientRows**.

The following are specific error codes which apply to this ROP.

Name	Value	Meaning
ecNotFound	0x8004010F	The MID does not correspond to a message in the database. The user does not have rights to the message. The folder corresponding to the FID entered in the ROP request buffer does not contain a message with the entered MID. The InputHandleIndex on which this ROP was called does not refer to a folder or Logon object.

Name	Value	Meaning
		The message is soft deleted and the client has not specified the OpenSoftDeleted flag as part of the OpenModeFlag field.

3.2.5.2 RopCreateMessage

Creates a new Message object on the server and provides access to it by returning a Message object handle for use in subsequent ROPs. The server MUST NOT commit the new Message object until it receives a **RopSaveChangesMessage** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.3).

The server MUST initialize the following properties before responding.

Property name	Initial Data
PidTagImportance ([MS-OXPROPS] section 2.821)	"0x00000001"
PidTagMessageClass ([MS-OXPROPS] section 2.884)	"IPM.Note"
PidTagSensitivity ([MS-OXPROPS] section 2.1115)	"0x00000000"
PidTagDisplayBcc ([MS-OXPROPS] section 2.744)	""
PidTagDisplayCc ([MS-OXPROPS] section 2.745)	""
PidTagDisplayTo ([MS-OXPROPS] section 2.748)	""
PidTagMessageFlags ([MS-OXPROPS] section 2.888)	"0x00000009"
PidTagMessageSize ([MS-OXPROPS] section 2.893)	See PidTagMessageSize in section 2.2.1.7
PidTagHasAttachments ([MS-OXPROPS] section 2.793)	"0x00"
PidTagSecurityDescriptor ([MS-OXPROPS] section 2.1100)	See PidTagSecurityDescriptor
PidTagUrlCompNameSet ([MS-OXPROPS] section 2.1167)	"0x00"
PidTagTrustSender ([MS-OXPROPS] section 2.1164)	"0x00000001"
PidTagAccess ([MS-OXCPRPT] section 2.2.1.1)	"0x00000003"
PidTagAccessLevel ([MS-OXCPRPT] section 2.2.1.2)	"0x00000001"
PidTagUrlCompName ([MS-OXPROPS] section 2.1166)	"No Subject.EML"

Property name	Initial Data
PidTagCreationTime (section 2.2.2.3)	The time the RopCreateMessage ROP ([MS-OXCROPS] section 2.2.6.2) was processed
PidTagLastModificationTime ([MS-OXPROPS] section 2.861)	Same as PidTagCreationTime
PidTagSearchKey ([MS-OXPROPS] section 2.1098)	Server generated SearchKey.
PidTagMessageLocaleId ([MS-OXPROPS] section 2.890)	The Logon object LocaleID.
PidTagCreatorName ([MS-OXPROPS] section 2.727)	Name of the creator.
PidTagCreatorEntryId ([MS-OXPROPS] section 2.726)	Address Book EntryID of the creator
PidTagLastModifierName ([MS-OXPROPS] section 2.863)	Same as PidTagCreatorName
PidTagLastModifierEntryId ([MS-OXPROPS] section 2.862)	Same as PidTagCreatorEntryId
PidTagHasNamedProperties ([MS-OXPROPS] section 2.795)	"0x00"
PidTagLocaleId ([MS-OXPROPS] section 2.871)	Same as PidTagMessageLocaleId
PidTagLocalCommitTime ([MS-OXPROPS] section 2.870)	Same as PidTagCreationTime

The following specific error code applies to this ROP.

Name	Value	Meaning
ecAccessDenied	0x80000009	The user does not have permissions to create this message.

3.2.5.3 RopSaveChangesMessage

Commits the changes made to a Message object on the server. The status of the Message object after the commit is determined by the value of the **SaveFlags** flag as documented in section [2.2.3.3.1.1](#).

The response contains the message ID of the committed message.

The **RopSaveChangesMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.3) commits the changes made to the Message object.

For this ROP, the the index in the **ResponseHandleIndex** field references the containing Folder object or, for an embedded message, the Embedded Message object. The index in the **InputHandleIndex** field references a Message object.

When the server receives multiple requests to open the same Message object, it returns a different handle and maintains a separate transaction for each. Any changes made on one transaction MUST

NOT be visible to another transaction until the changes are committed via the **RopSaveChangesMessage** ROP. Once a transaction on one handle has been committed, the server MUST return "ecObjectModified" for **RopSaveChangesMessage** ROP requests on other handles and MUST NOT allow those transactions to be committed, unless the client instructs the server to override previous changes with the ForceSave flag from section [2.2.3.3.1.1.<7>](#)

If pending changes include changes to read-only properties, the server MAY<8> return an error.

The following specific error code applies to this ROP.

Name	Value	Meaning
ecNotSupported	0x80040102	The values of the SaveFlags flags are not a supported combination as documented in section 2.2.3.3.1.1 .
ecObjectModified	0x80040109	The underlying data for this Message object was changed through another transaction context.

3.2.5.4 RopRemoveAllRecipients

Removes all the recipients from a Message object.

The server ignores the value of Reserved.

Until the server receives a **RopSaveChangesMessage** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.3) from the client, the server adheres to the following:

- The **RowIds** and associated data of removed recipients MUST NOT be returned as part of any subsequent handling of ROPs for the opened Message object on the same Message object handle.
- The changes made to the recipients MUST NOT be included in the response buffer returned for ROP requests that apply to recipients on different Message object handles.

The following specific error code applies to this ROP.

Name	Value	Meaning
ecNullObject	0x000004b9	The InputHandleIndex on which this ROP was called does not refer to a Message object.

3.2.5.5 RopModifyRecipients

Modifies the recipients on a Message object according to the data in the request buffer. The format of the request buffer is specified in section [2.2.3.5.1](#).

For each recipient provided the server locates its representation of the recipient based on **RowId**. If the recipient indicated by **RowId** does not exist then the server creates a new recipient with that **RowId** and applies the data from the request.

If the recipient currently exists on the Message object and the value of **RecipientRowSize** in the request buffer is nonzero, the server replaces all existing properties of the recipient with the property values supplied in the request. If the value of **RecipientRowSize** in the request buffer is "0x0000" then the server deletes the recipient from the Message object.

Until the server receives a **RopSaveChangesMessage** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.3) from the client, the server adheres to the following:

- If a recipient was deleted, its **RowId** and associated data MUST NOT be returned as part of any subsequent handling of ROPs for the opened Message object.
- Any changes made to the recipients MUST be included in the response buffer for any subsequent ROP requests that apply to recipients for the same Message object handle.
- The changes made to the recipients MUST NOT be included in the response buffer returned for ROP requests that apply to recipients on different Message object handles.

The following specific error code applies to this ROP.

Name	Value	Meaning
ecNullObject	0x000004b9	The InputHandleIndex on which this ROP was called does not refer to a Message object.

3.2.5.6 RopReadRecipients

Provides the recipient information for the Message object in a tabular form, where each row has information for a single recipient. The **RopReadRecipients** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.6) is used to obtain information for all recipients in the Message object, regardless of the number of recipients on the message.

The server provides the recipient information starting with the recipient specified by the **RowId** field. If there is a recipient with the given **RowId**, the server provides the information for that recipient and as many recipients as possible, limited by the number of actual recipients in the message and the amount of recipient information that fits in the response buffer.

When **RowId** is "0x00000000", the server returns all recipients for the message, beginning with the first recipient and filling the response buffer with as many **RecipientRow** structures ([\[MS-OXCDATA\]](#) section 2.8.3) as will fit. If the message does not have recipients, the server returns **ecNotFound**.

The following specific error codes apply to this ROP.

Name	Value	Meaning
ecNotFound	0x8004010F	Recipient row RowId does not exist on the message.
ecBufferTooSmall	0x0000047D	Unable to fit at least one recipient in the response buffer.
ecNullObject	0x000004b9	The InputHandleIndex on which this ROP was called does not refer to a Message object.

3.2.5.7 RopReloadCachedInformation

Returns the same information as would be returned by the **RopOpenMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.1), except that it is updated with the information that has been modified on the Message object.

The following specific error code applies to this ROP.

Name	Value	Meaning
ecNullObject	0x000004b9	The InputHandleIndex on which this ROP was called does not refer to a

Name	Value	Meaning
		Message object.

3.2.5.8 RopSetMessageStatus

Modifies the **PidTagMessageStatus** property ([\[MS-OXPROPS\]](#) section 2.895) of a single message.

The server modifies the bits on this property specified by the **MessageStatusMask**.

The server immediately commits the changes to the Message object as if the Message object had been opened and the **RopSaveChangesMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.3) had been called, except that it only changes the **PidTagMessageStatus** property, not the **PidTagChangeKey** property ([\[MS-OXCFXICS\]](#) section 2.2.1.2.7), the **PidTagLastModificationTime** property ([\[MS-OXPROPS\]](#) section 2.861), or any other property that is modified during the **RopSaveChangesMessage** ROP request.

The following specific error code applies to this ROP.

Name	Value	Meaning
ecNullObject	0x000004b9	The InputHandleIndex on which this ROP was called does not refer to a Folder object.

3.2.5.9 RopGetMessageStatus

Provides the value of the **PidTagMessageStatus** property ([\[MS-OXPROPS\]](#) section 2.895) of a single message; MUST NOT require the Message object to be opened.

The following specific error code applies to this ROP.

Name	Value	Meaning
ecNullObject	0x000004b9	The InputHandleIndex on which this ROP was called does not refer to a Folder object.

3.2.5.10 RopSetReadFlags

Note Some of the information in this section is subject to change because it applies to a preliminary implementation of the protocol or structure. For information about specific differences between versions, see the behavior notes that are provided in the Product Behavior appendix.

Modifies the **PidTagMessageFlags** property ([\[MS-OXPROPS\]](#) section 2.888) of several messages.

The server immediately commits the changes to the Message objects as if the Message objects had been opened and RopSaveMessageChanges had been called, except that it only changes the **PidTagMessageFlags** property, not the **PidTagChangeKey** property ([\[MS-OXCFXICS\]](#) section 2.2.1.2.7), the **PidTagLastModificationTime** property ([\[MS-OXPROPS\]](#) section 2.861), or any other property that is modified during a **RopSaveChangesMessage** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.3).

If the **WantAsynchronous** flag is nonzero, the server SHOULD return a **RopProgress ROP response** ([\[MS-OXCROPS\]](#) section 2.2.8.13) but MAY return a **RopSetReadFlags** ROP response ([\[MS-OXCROPS\]](#) section 2.2.6.10) instead. If the **SUPPORT_PROGRESS** flag, as specified in [\[MS-OXCSTOR\]](#) section 2.2.1.1.1.2, is not set by the client in the **OpenFlags** field in the **RopLogon** ROP

([\[MS-OXCROPS\]](#) section 2.2.3.1), then the server MUST disable asynchronous processing of the **RopSetReadFlags** ROP and MUST NOT return the **RopProgress** ROP whether or not the **WantAsynchronous** flag is set [<9>](#)

If the server has not received a SUPPORT_PROGRESS flag in the request buffer of the **RopLogon** ROP ([\[MS-OXCROPS\]](#) section 2.2.3.1), as specified in [\[MS-OXCSTOR\]](#) section 2.2.1.1.1.2, then the server MUST disable asynchronous processing for the **RopSetReadFlags** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.10), overriding any value of the **WantAsynchronous** flag (section [2.2.3.10.1.1](#)). In this case a **RopProgress** ROP MUST NOT be sent [<10>](#). If the client does not pass SUPPORT_PROGRESS, the server will process the entire **RopSetReadFlags** ROP request before returning a response to the client. If the client does pass SUPPORT_PROGRESS, and the client also passes the **WantAsynchronous** flag, the server performs asynchronously and returns the **RopProgress** ROP to inform the client of status processing the **RopSetReadFlags** ROP.

If the server is unable to modify one or more of the Message objects that are specified in the **MessageIds** field (section [2.2.3.10.1.4](#)) of the request buffer, then the server returns the **PartialCompletion** flag (section [2.2.3.10.2.1](#)) in the response buffer.

The following specific error code applies to this ROP.

Name	Value	Meaning
ecNullObject	0x000004b9	The InputHandleIndex on which this ROP was called does not refer to a Folder object.

3.2.5.11 RopSetMessageReadFlag

Modifies the **PidTagMessageFlags** property ([\[MS-OXPROPS\]](#) section 2.888) of a single message.

If this operation is performed in public folder mode as specified in section [2.2.3.11](#), the server finds the message associated with the LongTermID contained in the *ClientData* request parameter using the method specified in [\[MS-OXCSTOR\]](#) section 3.2.5.9.

The server immediately commits the changes to the Message object as if the Message object had been opened and the **RopSaveChangesMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.3) had been called, except that it only changes the **PidTagMessageFlags** property, not the **PidTagChangeKey** property ([\[MS-OXCFCICS\]](#) section 2.2.1.2.7), the **PidTagLastModificationTime** property ([\[MS-OXPROPS\]](#) section 2.861), or any other property that is modified during a **RopSaveChangesMessage** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.3).

The following specific error code applies to this ROP.

Name	Value	Meaning
ecNullObject	0x000004b9	The InputHandleIndex on which this ROP was called does not refer to a Message object.

3.2.5.12 RopOpenAttachment

Provides access to a single existing Attachment object. The handle returned by this ROP is used in subsequent ROPs, such as the **RopGetPropertiesSpecific** ROP ([\[MS-OXCROPS\]](#) section 2.2.8.3). For details about which ROPs operate on Attachment objects, see the sections for the ROPs in [\[MS-OXCROPS\]](#).

The following specific error codes apply to this ROP.

Name	Value	Meaning
ecNotFound	0x8004010F	The AttachmentID does not correspond to an attachment on the Message object.
ecAccessDenied	0x80000009	The user has insufficient privileges.
ecNullObject	0x000004b9	The InputHandleIndex on which this ROP was called does not refer to a Message object.

3.2.5.13 RopCreateAttachment

Creates a new Attachment object and provides a handle to it for use in subsequent ROPs. The server does not commit the new Attachment object until it receives a call to the **RopSaveChangesAttachment** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.15).

The server MUST initialize the following properties before responding.

PropertyName	Initial Data
PidTagAttachNumber (section 2.2.2.6)	Varies, depending on the number of existing attachments on the Message object.
PidTagAttachSize (section 2.2.2.5)	"0x00000000"
PidTagAccessLevel ([MS-OXCPRPT] section 2.2.1.2)	"0x00000001"
PidTagRenderingPosition ([MS-OXPROPS] section 2.1012)	"0x00000000"
PidTagCreationTime (section 2.2.2.3)	The time the RopCreateAttachment ROP ([MS-OXCROPS] section 2.2.6.13) was processed.
PidTagLastModificationTime ([MS-OXPROPS] section 2.861)	Same as the PidTagCreationTime property.

The following specific error codes apply to this ROP.

Name	Value	Meaning
ecAccessDenied	0x80000009	The user does not have permissions to create an attachment on this message.
ecMaxAttachmentExceeded	0x000004DB	The (server defined) maximum number of attachments for a message has been exceeded.
ecNullObject	0x000004b9	The InputHandleIndex on which this ROP was called does not refer to a Message object.

3.2.5.14 RopSaveChangesAttachment

Commits the changes made to an Attachment object. The status of the Attachment object after the commit is determined by the values of the **SaveFlags** as documented in section [2.2.3.3.1.1](#).

Although the server commits any pending changes to the Attachment object in the context of its containing Message object, the changes MUST NOT be committed to the database until the

RopSaveChangesMessage ROP ([\[MS-OXCROPS\]](#) section 2.2.6.3) has been executed on the handle of the Message object.

The following specific error code applies to this ROP.

Name	Value	Meaning
ecNotSupported	0x80040102	The value of SaveFlags is not a supported combination as documented in section 2.2.3.3.1.1 . The InputHandleIndex on which this ROP was called does not refer to an Attachment object.

3.2.5.15 RopDeleteAttachment

Removes an Attachment object from a message. The server recalculates the **PidTagHasAttachments** property ([\[MS-OXPROPS\]](#) section 2.793) during this ROP.

The attachment is not permanently removed from the message until the client calls the **RopSaveChangesMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.3).

The following specific error codes apply to this ROP.

Name	Value	Meaning
ecNotFound	0x8004010F	The AttachmentID does not correspond to an attachment on the Message object.
ecAccessDenied	0x80000009	The user has insufficient privileges.
ecNullObject	0x000004b9	The InputHandleIndex on which this ROP was called does not refer to a Message object.

3.2.5.16 RopOpenEmbeddedMessage

Provides access to an Embedded Message object stored in an Attachment object. If the embedded object does not exist, then the client creates an Attachment object following the process defined in section [3.1.4.17](#). Once the attachment is created and its properties initialized, as specified in section [3.1.4.17](#), the client sends a **RopOpenEmbeddedMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.16) on the attachment to get a Message object handle. The returned handle is used in subsequent ROPs (similar to the one returned by the **RopOpenMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.1)). The server MUST NOT commit the Message object to the containing Attachment object until **RopSaveChangesMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.3) is called with the Embedded Message object's handle.

The following specific error codes apply to this ROP.

Name	Value	Meaning
ecAccessDenied	0x80000009	The user does not have permission to open or create this message.
ecNullObject	0x000004b9	The InputHandleIndex on which this ROP was called does not refer to an Attachment object.
ecUnknownCodePage	0x000003ef	The code page is unknown

3.2.5.17 RopGetAttachmentTable

Returns a handle to a table object for use in subsequent ROPs as specified in [\[MS-OXCTABL\]](#). The Table object returned allows access to the properties of Attachment objects.

The following specific error codes apply to this ROP.

Name	Value	Meaning
ecNullObject	0x000004b9	The InputHandleIndex parameter on which this ROP was called does not refer to a Message object.
ecBusy	0x80040108	The server is too busy to complete the request.

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

4 Protocol Examples

A user creates a new HTML-format e-mail, sets its subject to "abc123sample" and its body to "This is a sample body text". The user also adds two attachments: an HTML embedded image and a text file, adds a recipient, then saves and closes the message.

4.1 Create Message

The client first creates a new Message object by sending a **RopCreateMessage** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.2).

4.1.1 RopCreateMessage Request Buffer

```
0000: 06 00 00 01 ff 0f 01 00-00 00 00 f0 79 93 00
```

RopId: 0x06

LogonId: 0x00

InputHandleIndex: 0x00

OutputHandleIndex: 0x01

CodePageId: 0x0FFF

FolderId: 01 00 00 00 00 f0 79 93

AssociatedFlag: 0x00

4.1.2 RopCreateMessage Response Buffer

```
0000: 06 01 00 00 00 00 00 00
```

RopId: 0x06

OutputHandleIndex: 0x01

ReturnValue: 0x00000000

HasMessageId: 0x00

4.2 Name to Id Mapping

Before manipulating named properties on Message objects, the client needs to ask the server to perform a mapping from the named properties to property identifiers, using the **RopGetPropertyIdsFromNames** ROP ([\[MS-OXCROPS\]](#) section 2.2.8.1) as described in [\[MS-OXCPRPT\]](#) section 2.2.12.

4.3 Get Attachment Table

The client sends a **RopGetAttachmentTable** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.17) to retrieve the **attachments table** for a Message object.

4.3.1 RopGetAttachmentTable Request Buffer

```
0000:21 00 00 01 00
```

RopId: 0x21

LogonId: 0x00

InputHandleIndex: 0x00

OutputHandleIndex: 0x01

TableFlags: 0x00 (Standard)

4.3.2 RopGetAttachmentTable Response Buffer

```
0000:21 01 00 00 00 00
```

RopId: 0x21

OutputHandleIndex: 0x01

ReturnValue: 0x00000000

4.4 Insert HTML Embedded Image

The client first creates the Attachment object on the Message object, then sets its properties and commits the changes.

4.4.1 RopCreateAttachment Request Buffer

```
0000: 23 00 00 01
```

RopId: 0x23

LogonId: 0x00

InputHandleIndex: 0x00

OutputHandleIndex: 0x01

4.4.2 RopCreateAttachment Response Buffer

```
0000: 23 01 00 00 00 00 00 00-00 00
```

RopId: 0x23

OutputHandleIndex: 0x01

ReturnValue: 0x00000000

AttachmentID: 0x00000000

4.4.3 Setting Properties

At this point the client uses the **RopSetProperties** ROP ([\[MS-OXCROPS\]](#) section 2.2.8.6) as described in [\[MS-OXCPRPT\]](#) section 2.2.5 to set properties on the Attachment objects.

Property tag	Property name	Data
0x37050003	PidTagAttachMethod (section 2.2.2.9)	"0x00000001"
0x370b0003	PidTagRenderingPosition (section 2.2.2.16)	"0xFFFFFFFF"
0x7ffd0003	PidTagAttachmentFlags (section 2.2.2.23)	"0x00000000"
0x3001001f	PidTagDisplayName ([MS-OXCFOLD] section 2.3.2.2.3)	"image001.PNG"
0x3712001f	PidTagAttachContentId (section 2.2.2.26)	"image001.PNG@01C86E1C.F1954390"
0x370e001f	PidTagAttachMimeTag (section 2.2.2.26)	"image/PNG"
0x7ffa0003	PidTagAttachmentLinkId (section 2.2.2.22)	"0x00000000"
0x37140003	PidTagAttachFlags (section 2.2.2.18)	"0x00000004"
0x7ffe000b	PidTagAttachmentHidden (section 2.2.2.24)	"0x01"
0x3707001f	PidTagAttachLongFilename (section 2.2.2.10)	"image001.PNG"
0x3704001f	PidTagAttachFilename (section 2.2.2.11)	"image001.PNG"
0x3703001f	PidTagAttachExtension (section 2.2.2.12)	".PNG"

To set the contents of the embedded image, the client uses four ROPs.

The **RopOpenStream** ROP ([\[MS-OXCROPS\]](#) section 2.2.9.1) with the **PidTagAttachDataBinary** property (section [2.2.2.7](#)).

The **RopSetStreamSize** ROP ([\[MS-OXCROPS\]](#) section 2.2.9.6) with the size of image file data.

The **RopWriteStream** ROP request ([\[MS-OXCROPS\]](#) section 2.2.9.3) with the actual file contents.

The **RopRelease** ROP ([\[MS-OXCROPS\]](#) section 2.2.15.3) for the handle returned from the **RopOpenStream** ROP.

4.4.4 RopSaveChangesAttachment Request Buffer

```
0000: 25 00 01 00 02
```

RopId: 0x25

LogonId: 0x00

ResponseHandleIndex: 0x01

InputHandleIndex: 0x00

SaveFlags: 0x0A (**KeepOpenReadWrite**)

4.4.5 RopSaveChangesAttachment Response Buffer

```
0000: 25 01 00 00 00 00
```

RopId: 0x25

ResponseHandleIndex: 0x01

ReturnValue: 0x00000000

4.4.6 Releasing Attachment Object

Finally, the client releases the Attachment object by using the **RopRelease** ROP ([\[MS-OXCROPS\]](#) section 2.2.15.3).

4.5 Attach Text File

The client first creates the Attachment object on the Message object, then sets its properties and commits the changes.

4.5.1 RopCreateAttachment Request Buffer

```
0000:23 00 00 03
```

RopId: 0x23

LogonId: 0x00

InputHandleIndex: 0x00

OutputHandleIndex: 0x03

4.5.2 RopCreateAttachment Response Buffer

```
0000: 23 03 00 00 00 00 01 00-00 00
```

RopId: 0x23

OutputHandleIndex: 0x03

ReturnValue: 0x00000000

AttachmentID: 0x00000001

4.5.3 Setting Properties

At this point the client uses the **RopSetProperties** ROP ([\[MS-OXCROPS\]](#) section 2.2.8.6) as described in [\[MS-OXCPRPT\]](#) section 2.2.5 to set properties on the Attachment objects.

Property tag	Property name	Data
0x37050003	PidTagAttachMethod (section 2.2.2.9)	"0x00000001"
0x370b0003	PidTagRenderingPosition (section 2.2.2.16)	"0xFFFFFFFF"
0x7ffd0003	PidTagAttachmentFlags (section 2.2.2.23)	"0x00000000"
0x3001001f	PidTagDisplayName ([MS-OXCFOOLD] section 2.3.2.2.3)	"test.txt"
0x7ffa0003	PidTagAttachmentLinkId (section 2.2.2.22)	"0x00000000"
0x37140003	PidTagAttachFlags (section 2.2.2.18)	"0x00000000"
0x7ffe000b	PidTagAttachmentHidden (section 2.2.2.24)	"0x00"
0x3707001f	PidTagAttachLongFilename (section 2.2.2.10)	"test.txt"
0x3704001f	PidTagAttachFilename (section 2.2.2.11)	"test.txt"
0x3703001f	PidTagAttachExtension (section 2.2.2.12)	".txt"
0x30070040	PidTagCreationTime (section 2.2.2.3)	"2008/02/1222:28:34.636"
0x30080040	PidTagLastModificationTime (section 2.2.2.2)	"2008/02/1222:28:50.112"
0x37090102	PidTagAttachRendering (section 2.2.2.17)	3,512 bytes of Windows metafile

To set the contents of the embedded image, the client uses four ROPs:

1. The **RopOpenStream** ROP ([\[MS-OXCROPS\]](#) section 2.2.9.1) with **PidTagAttachDataBinary** (section [2.2.2.7](#)).
2. The **RopSetStreamSize** ROP ([\[MS-OXCROPS\]](#) section 2.2.9.6) with the size of image file data.
3. The **RopWriteStream** ROP request ([\[MS-OXCROPS\]](#) section 2.2.9.3) with the actual file contents.
4. The **RopRelease** ROP ([\[MS-OXCROPS\]](#) section 2.2.15.3) for the handle returned from the **RopOpenStream** ROP.

4.5.4 RopSaveChangesAttachment Request Buffer

```
0000: 25 00 02 01 02
```

RopId: 0x25

LogonId: 0x00

ResponseHandleIndex: 0x02

InputHandleIndex: 0x01

SaveFlags: 0x0A (**KeepOpenReadWrite**)

4.5.5 RopSaveChangesAttachment Response Buffer

```
0000: 25 02 00 00 00 00
```

RopId: 0x25

ResponseHandleIndex: 0x02

ReturnValue: 0x00000000

4.5.6 Releasing Attachment Object

Finally, the client releases the Attachment object by using the **RopRelease** ROP ([\[MS-OXCROPS\]](#) section 2.2.15.3).

4.6 Setting Message Properties

The client sets all the necessary properties using the **RopSetProperties** ROP ([\[MS-OXCROPS\]](#) section 2.2.8.6).

The HTML body, stored in the **PidTagBodyHtml** property (section [2.2.1.44.3](#)), is the following:

```
<html>
<head>
<meta http-equiv=Content-Type content="text/html; charset=us-ascii">
</head>
<body lang=EN-US link=blue vlink=purple>
<p>This is a sample body text</p></p>
<p ></p></p>
</div>
</body>
</html>
```

4.7 Adding Recipients

4.7.1 RopModifyRecipients Request Buffer

```
0000:0e 00 08 0c 00 03 00 fe-0f 03 00 00 39 1f 00 ff
0010:39 1f 00 fe 39 03 00 71-3a 03 00 05 39 1f 00 ff
0020:5f 03 00 fd 5f 03 00 ff-5f 03 00 de 5f 03 00 df
0030:5f 02 01 f7 5f 01 00 00-00 00 00 01 27 01 51 06
0040:5a 00 55 73 65 72 32 00-75 00 73 00 65 00 72 00
0050:32 00 00 00 75 00 73 00-65 00 72 00 32 00 00 00
0060:0c 00 00 06 00 00 00 00-00 00 00 75 00 73 00 65
0070:00 72 00 32 00 00 00 75-00 73 00 65 00 72 00 32
0080:00 40 00 73 00 7a 00 66-00 6b 00 75 00 6b 00 2d
0090:00 64 00 6f 00 6d 00 2e-00 65 00 78 00 74 00 65
00a0:00 73 00 74 00 2e 00 6d-00 69 00 63 00 72 00 6f
00b0:00 73 00 6f 00 66 00 74-00 2e 00 63 00 6f 00 6d
00c0:00 00 00 00 00 00 00 00-00 00 40 75 00 73 00 65
00d0:00 72 00 32 00 00 00 01-00 00 00 00 00 00 00 00
00e0:00 00 00 00 00 00 00 7c-00 00 00 00 00 dc a7 40
00f0:c8 c0 42 10 1a b4 b9 08-00 2b 2f e1 82 01 00 00
```

```
0100:00 00 00 00 00 2f 6f 3d-46 69 72 73 74 20 4f 72
0110:67 61 6e 69 7a 61 74 69-6f 6e 2f 6f 75 3d 45 78
0120:63 68 61 6e 67 65 20 41-64 6d 69 6e 69 73 74 72
0130:61 74 69 76 65 20 47 72-6f 75 70 20 28 46 59 44
0140:49 42 4f 48 46 32 33 53-50 44 4c 54 29 2f 63 6e
0150:3d 52 65 63 69 70 69 65-6e 74 73 2f 63 6e 3d 75
0160:73 65 72 32 00
```

RopId: 0x0e

LogonId: 0x00

InputHandleIndex: 0x08

ColumnCount: 0x000C (COUNT of following **RecipientColumns**)

PidTagObjectType ([\[MS-OXCPRPT\]](#) section 2.2.1.7): 0x0ffe0003

PidTagDisplayType ([\[MS-OXOABK\]](#) section 2.2.3.11): 0x39000003

PidTagAddressBookDisplayNamePrintable (section [2.2.1.30](#)): 0x39ff001f

PidTagSmtppAddress ([\[MS-OXOABK\]](#) section 2.2.3.21): 0x39fe001f

PidTagSendInternetEncoding ([\[MS-OXPROPS\]](#) section 2.1113): 0x3a710003

PidTagDisplayTypeEx ([\[MS-OXOABK\]](#) section 2.2.3.12): 0x39050003

PidTagRecipientDisplayName ([\[MS-OXPROPS\]](#) section 2.995): 0x5ff6001f

PidTagRecipientFlags ([\[MS-OXPROPS\]](#) section 2.997): 0x5ffd0003

PidTagRecipientTrackStatus ([\[MS-OXPROPS\]](#) section 2.1004): 0x5fff0003

PidTagRecipientResourceState ([\[MS-OXPROPS\]](#) section 2.1003): 0x5fde0003

PidTagRecipientOrder ([\[MS-OXPROPS\]](#) section 2.998): 0x5fdf0003

PidTagRecipientEntryId ([\[MS-OXPROPS\]](#) section 2.996): 0x5ff70102

RowCount: 0x0001 (COUNT of following **ModifyRecipientRows**)

RowId: 0x00000000

RecipientType: 0x01 (primary recipient)

RecipientRowSize: 0x0127 (bytes in following **RecipientRow**)

RecipientFlags: 0101000100000110 (S,D,Type=X500DN,I,U)

AddressPrefixUsed: 0x5A (present because Type=X500DN)

DisplayType: 0x00 (present because Type=X500DN)

EmailAddress: User2 (present because Type=X500DN)

DisplayName: user2 (present because D is set)

SimpleDisplayName: user2 (present because I is set)
RecipientColumnCount: 0x000C (matches **ColumnCount**)
StandardPropertyRow:
Flag: 0x00
ValueArray: (property order defined by **RecipientColumns**)
PidTagObjectType: 0x00000006
PidTagDisplayType: 0x00000000
PidTag7BitDisplayName ([\[MS-OXOABK\]](#) section 2.2.3.7): user2
PidTagSmtAddress: user2@szfkuk-dom.extest.microsoft.com
PidTagSendInternetEncoding: 0
PidTagDisplayTypeEx: 0x40000000
PidTagRecipientDisplayName: user2
PidTagRecipientFlags: 0x00000001
PidTagRecipientTrackStatus: 0x00000000
PidTagRecipientResourceState: 0x00000000
PidTagRecipientOrder: 0x00000000
PidTagRecipientEntryId: 0x007c and the subsequent 124 (0x7c) bytes

4.7.2 RopModifyRecipients Response Buffer

```
0000: 0e 08 00 00 00 00
```

RopId: 0x0e
InputHandleIndex: 0x08
ReturnValue: 0x000000

4.8 Save Message

After all necessary properties were set for the message it was saved. The client sends a **RopSaveChangesMessage** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.3).

4.8.1 RopSaveChangesMessage Request Buffer

```
0000: 0c 00 00 01 0a
```

RopId: 0x0c
LogonId: 0x00

ResponseHandleIndex: 0x00

InputHandleIndex: 0x01

SaveFlags: 0x0A (KeepOpenReadWrite)

4.8.2 RopSaveChangesMessage Response Buffer

```
0000: 0c 00 00 00 00 00 01 01-00 00 00 00 f0 86 39
```

RopId: 0x0c

ResponseHandleIndex: 0x00

ReturnValue: 0x00000000

InputHandleIndex: 0x01

MessageId: 01 00 00 00 00 f0 86 39

4.9 Releasing Message Object

Finally, the client releases the Message object by using the **RopRelease** ROP ([\[MS-OXCROPS\]](#) section 2.2.15.3).

5 Security

5.1 Security Considerations for Implementers

There are no special security considerations specific to this protocol. General security considerations pertaining to the underlying **remote procedure call (RPC)**-based transport apply (see [\[MS-OXCROPS\]](#)).

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® Office Outlook® 2003
- Microsoft® Exchange Server 2003
- Microsoft® Office Outlook® 2007
- Microsoft® Exchange Server 2007
- Microsoft® Outlook® 2010
- Microsoft® Exchange Server 2010
- Microsoft® Outlook® 2010 Service Pack 1 (SP1) Beta
- Microsoft® Exchange Server 2010 Service Pack 1 (SP1)

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:](#) Exchange 2010 returns Success when attempting to save read-only properties; the values, however, are not persisted to the server store.

[<2> Section 2.2.1.6:](#) Office Outlook 2003, Office Outlook 2007, and Outlook 2010 ignore the **mfEverRead** flag.

[<3> Section 2.2.1.46.4:](#) Outlook 2010 sets the **PidTagStartDateEtc** property (section [2.2.1.46.4](#)) on a message if it was not defined by the server, using the value of the **PidTagMessageDeliveryTime** property ([\[MS-OXPROPS\]](#) section 2.886) if it exists and the value of the **PidTagCreationTime** property ([\[MS-OXPROPS\]](#) section 2.725) otherwise.

[<4> Section 2.2.2.20:](#) Office Outlook 2007 correctly detects MacBinary I, MacBinary II, and MacBinary III formats. Office Outlook 2003 only correctly detects MacBinary I and MacBinary II.

[<5> Section 2.2.3.15:](#) Exchange 2010 returns a GeneralFailure error if pending changes include changes to read-only properties.

[<6> Section 2.2.3.18:](#) Exchange 2010 does not support **RopGetValidAttachments**.

[<7> Section 3.2.5.3:](#) Exchange 2010 returns Success for **RopSaveChangesMessage** ROP requests ([\[MS-OXCROPS\]](#) section 2.2.6.3) when a previous request has already been committed against the Message object, even though the changes to the object are not actually committed to the server store.

[<8> Section 3.2.5.3:](#) Exchange 2010 returns a GeneralFailure error if pending changes include changes to read-only properties.

<9> [Section 3.2.5.10](#): Exchange 2003, Exchange 2007, Exchange 2010, Office Outlook 2003, Office Outlook 2007, and Outlook 2010 do not support this behavior. Exchange 2010 SP1 and Outlook 2010 SP1 Beta support this behavior.

<10> [Section 3.2.5.10](#): Exchange 2003, Exchange 2007, Exchange 2010, Office Outlook 2003, Office Outlook 2007, and Outlook 2010 do not support this behavior. Exchange 2010 SP1 and Outlook 2010 SP1 Beta support this behavior.

7 Change Tracking

This section identifies changes that were made to the [MS-OXCMSG] protocol document between the November 2010 and March 2011 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
1 Introduction	Added information about which sections of the specification are normative and can contain RFC 2119 language.	Y	New content added for template compliance.
2.2.1.22 PidLidCategories	New topic.	Y	New content added.
2.2.1.22 PidLidCategories	New topic.	Y	New content added.
2.2.1.23 PidLidClassification	New topic.	Y	New content added.
2.2.1.24 PidLidClassificationDescription	New topic.	Y	New content added.
2.2.1.25 PidLidClassified	New topic.	Y	New content added.
2.2.1.26 PidTagInternetReferences	New topic.	Y	New content

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
			added.
2.2.1.27 PidLidInfoPathFormName	New topic.	Y	New content added.
2.2.1.28 PidTagMimeSkeleton	New topic.	Y	New content added.
2.2.1.29 PidTagTnefCorrelationKey	New topic.	Y	New content added.
2.2.1.30 PidTagAddressBookDisplayNamePrintable	New topic.	Y	New content added.
2.2.1.31 PidTagCreatorEntryId	New topic.	Y	New content added.
2.2.1.32 PidTagLastModifierEntryId	New topic.	Y	New content added.
2.2.1.33 PidLidAgingDontAgeMe	New topic.	Y	New content added.
2.2.1.34 PidLidCurrentVersion	New topic.	Y	New content added.
2.2.1.35 PidLidCurrentVersionName	New topic.	Y	New content added.
2.2.1.36 PidTagAlternateRecipientAllowed	New topic.	Y	New content added.
2.2.1.37 PidTagResponsibility	New topic.	Y	New content added.
2.2.1.38 PidTagRowid	New topic.	Y	New content added.
2.2.1.39 PidTagHasNamedProperties	New topic.	Y	New content added.
2.2.1.40	New topic.	Y	New

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
PidTagRecipientOrder			content added.
2.2.1.41 PidNameContentBase	New topic.	Y	New content added.
2.2.1.42 PidNameAcceptLanguage	New topic.	Y	New content added.
2.2.1.43 PidTagPurportedSenderDomain	New topic.	Y	New content added.
2.2.2.26 MIME Properties	Added the PidTagAttachPayloadClass, PidTagAttachPayloadProviderGuidString, PidNameAttachmentMacContentType, and PidNameAttachmentMacInfo properties to the table.	Y	New content added.
2.2.3.1 RopOpenMessage Buffer Format	Moved server processing behavior to section 3.2.5.	Y	Content removed.
2.2.3.3 RopSaveChangesMessage Buffer Format	Moved server processing behavior to section 3.2.5.	Y	Content removed.
2.2.3.10.1.1 WantAsynchronous	Moved server processing behavior to section 3.2.5.	Y	Content removed.
3.2.5.1 RopOpenMessage	Moved server processing behavior from section 2 into this section.	Y	Content removed.
3.2.5.3 RopSaveChangesMessage	Moved server processing behavior from section 2 into this section.	Y	Content removed.
3.2.5.10 RopSetReadFlags	Moved server processing behavior from section 2 into this section.	Y	Content removed.

8 Index

A

Abstract data model
 [client](#) 52
 [server](#) 57
[Applicability](#) 14
[Attachment Object Properties message](#) 30

C

[Capability negotiation](#) 14
[Change tracking](#) 80
Client
 [abstract data model](#) 52
 [initialization](#) 52
 [message processing](#) 57
 [other local events](#) 57
 [sequencing rules](#) 57
 [timer events](#) 57
 [timers](#) 52

D

Data model - abstract
 [client](#) 52
 [server](#) 57

F

[Fields - vendor-extensible](#) 14

G

[Glossary](#) 10

H

Higher-layer triggered events
 [server](#) 58

I

[Implementer - security considerations](#) 77
[Index of security parameters](#) 77
[Informative references](#) 13
Initialization
 [client](#) 52
 [server](#) 58
[Introduction](#) 10

M

[Message Object Properties message](#) 16
[Message Object ROPS message](#) 35
Message processing
 [client](#) 57
Messages
 [Attachment Object Properties](#) 30

[Message Object Properties](#) 16
[Message Object ROPS](#) 35
[transport](#) 16

N

[Normative references](#) 11

O

Other local events
 [client](#) 57
 [server](#) 67

P

[Parameters - security index](#) 77
[Preconditions](#) 14
[Prerequisites](#) 14
[Product behavior](#) 78

R

References
 [informative](#) 13
 [normative](#) 11
[Relationship to other protocols](#) 14

S

Security
 [implementer considerations](#) 77
 [parameter index](#) 77
Sequencing rules
 [client](#) 57
Server
 [abstract data model](#) 57
 [higher-layer triggered events](#) 58
 [initialization](#) 58
 [other local events](#) 67
 [timer events](#) 67
 [timers](#) 58
[Standards assignments](#) 15

T

Timer events
 [client](#) 57
 [server](#) 67
Timers
 [client](#) 52
 [server](#) 58
[Tracking changes](#) 80
[Transport](#) 16
Triggered events - higher-layer
 [server](#) 58

V

[Vendor-extensible fields](#) 14
[Versioning](#) 14