

[MS-OXOMSG]: E-Mail 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 [Open Specification Promise](#) or the [Community Promise](#). 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.

Preliminary Documentation. This Open Specification provides documentation for past and current releases and/or for the pre-release (beta) version of this technology. This Open Specification is final

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

Revision Summary

Date	Revision History	Revision Class	Comments
04/04/2008	0.1		Initial Availability.
04/25/2008	0.2		Revised and updated property names and other technical content.
06/27/2008	1.0		Initial Release.
08/06/2008	1.01		Updated references to reflect date of initial release.
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		Revised and edited technical content.
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	7.1	Minor	Clarified the meaning of the technical content.
03/18/2011	7.1	No change	No changes to the meaning, language, and formatting of the technical content.
08/05/2011	8.0	Major	Significantly changed the technical content.
10/07/2011	9.0	Major	Significantly changed the technical content.
01/20/2012	10.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction	9
1.1 Glossary	9
1.2 References	11
1.2.1 Normative References	11
1.2.2 Informative References	12
1.3 Overview	12
1.3.1 E-Mail Objects	12
1.3.1.1 Creating, Opening, and Saving E-Mail Objects	12
1.3.1.2 Sending Messages	13
1.3.1.3 Replying and Forwarding Messages	13
1.3.2 Report Messages	13
1.3.2.1 Read Receipt	13
1.3.2.2 Non-Read Receipt	13
1.3.2.3 Delivery Receipt	13
1.3.2.4 Non-Delivery Report	13
1.3.3 Voting and Tracking	13
1.3.4 Controlling Sending and Delivery of Mail	14
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	14
2 Messages	15
2.1 Transport	15
2.2 Message Syntax	15
2.2.1 E-Mail Object Properties	19
2.2.1.1 PidTagBlockStatus Property	19
2.2.1.2 PidTagConversationId Property	20
2.2.1.3 PidTagConversationIndex Property	20
2.2.1.4 PidTagConversationIndexTracking Property	22
2.2.1.5 PidTagConversationTopic Property	22
2.2.1.6 PidTagDeferredDeliveryTime Property	22
2.2.1.7 PidTagDisplayBcc Property	23
2.2.1.8 PidTagDisplayCc Property	23
2.2.1.9 PidTagDisplayTo Property	23
2.2.1.10 PidTagIconIndex Property	23
2.2.1.11 PidTagInternetMailOverrideFormat Property	24
2.2.1.12 PidTagInternetMessageId Property	25
2.2.1.13 PidTagInReplyToId Property	25
2.2.1.14 PidTagLastVerbExecuted Property	25
2.2.1.15 PidTagLastVerbExecutionTime Property	28
2.2.1.16 PidTagMessageClass Property	28
2.2.1.17 PidTagMessageToMe Property	28
2.2.1.18 PidTagMessageCcMe Property	28
2.2.1.19 PidTagMessageRecipientMe Property	28
2.2.1.20 PidTagOriginatorDeliveryReportRequested Property	28
2.2.1.21 PidTagOriginatorNonDeliveryReportRequested Property	29
2.2.1.22 PidTagOriginalSensitivity Property	29

2.2.1.23	PidTagReceivedRepresentingAddressType Property	29
2.2.1.24	PidTagReceivedRepresentingEmailAddress Property	29
2.2.1.25	PidTagReceivedRepresentingEntryId Property	29
2.2.1.26	PidTagReceivedRepresentingName Property	30
2.2.1.27	PidTagReceivedRepresentingSearchKey Property	30
2.2.1.28	PidTagReadReceiptRequested Property	30
2.2.1.29	PidTagReceivedByAddressType Property	30
2.2.1.30	PidTagReceivedByEmailAddress Property	30
2.2.1.31	PidTagReceivedByEntryId Property	31
2.2.1.32	PidTagReceivedByName Property	31
2.2.1.33	PidTagReceivedBySearchKey Property	31
2.2.1.34	PidTagRecipientReassignmentProhibited Property	31
2.2.1.35	PidTagReplyRecipientEntries Property	31
2.2.1.36	PidTagReplyRecipientNames Property	32
2.2.1.37	PidTagReplyRequested Property	32
2.2.1.38	PidTagResponseRequested Property	32
2.2.1.39	PidTagSendRichInfo Property	32
2.2.1.40	PidTagSenderAddressType Property	32
2.2.1.41	PidTagSenderEmailAddress Property	32
2.2.1.42	PidTagSenderEntryId Property	33
2.2.1.43	PidTagSenderName Property	33
2.2.1.44	PidTagSenderSearchKey Property	33
2.2.1.45	PidTagSenderSmtpAddress Property	33
2.2.1.46	PidTagSentRepresentingAddressType Property	33
2.2.1.47	PidTagSentRepresentingEmailAddress Property	33
2.2.1.48	PidTagSentRepresentingEntryId Property	34
2.2.1.49	PidTagSentRepresentingName Property	34
2.2.1.50	PidTagSentRepresentingSearchKey Property	34
2.2.1.51	PidTagSubjectPrefix Property	34
2.2.1.52	PidTagTransportMessageHeaders Property	34
2.2.1.53	PidLidInternetAccountName Property	35
2.2.1.54	PidLidInternetAccountStamp Property	35
2.2.1.55	PidTagPrimarySendAccount Property	35
2.2.1.56	PidTagNextSendAcct Property	35
2.2.1.57	PidLidUseTnef Property	35
2.2.1.58	Attachments	35
2.2.1.59	Categories and Keywords	36
2.2.1.60	Contacts	36
2.2.1.61	Flags	36
2.2.1.62	Reminders	36
2.2.1.63	Recipients	36
2.2.1.64	PidLidAutoProcessState Property	36
2.2.1.65	PidLidVerbStream Property	36
2.2.1.65.1	VoteOption Structure	37
2.2.1.65.2	VoteOptionExtras Structure	39
2.2.1.66	PidLidVerbResponse Property	40
2.2.1.67	PidTagTargetEntryId Property	40
2.2.1.68	PidTagAutoResponseSuppress Property	40
2.2.1.69	PidTagMessageEditorFormat Property	41
2.2.1.70	PidTagMessageSubmissionId Property	41
2.2.1.71	PidTagSenderIdStatus Property	41
2.2.1.72	PidTagListHelp Property	42
2.2.1.73	PidTagListSubscribe Property	42

2.2.1.74	PidTagListUnsubscribe Property	42
2.2.1.75	PidTagDelegatedByRule Property	42
2.2.1.76	PidTagCurrentDraftEntryId	42
2.2.2	Message Status Reports Properties	43
2.2.2.1	PidTagMessageClass Property.....	43
2.2.2.2	PidTagOriginalDeliveryTime Property	43
2.2.2.3	PidTagOriginalDisplayTo Property	43
2.2.2.4	PidTagOriginalDisplayCc Property	43
2.2.2.5	PidTagOriginalDisplayBcc Property	44
2.2.2.6	PidTagOriginalSenderAddressType Property	44
2.2.2.7	PidTagOriginalSenderEmailAddress Property	44
2.2.2.8	PidTagOriginalSenderEntryId Property	44
2.2.2.9	PidTagOriginalSenderName Property	44
2.2.2.10	PidTagOriginalSenderSearchKey Property.....	44
2.2.2.11	PidTagOriginalSentRepresentingAddressType Property.....	44
2.2.2.12	PidTagOriginalSentRepresentingEmailAddress Property	45
2.2.2.13	PidTagOriginalSentRepresentingEntryId Property	45
2.2.2.14	PidTagOriginalSentRepresentingName Property	45
2.2.2.15	PidTagOriginalSentRepresentingSearchKey Property.....	45
2.2.2.16	PidTagOriginalSubject Property.....	45
2.2.2.17	PidTagOriginalSubmitTime Property	45
2.2.2.18	PidTagParentKey Property	46
2.2.2.19	PidTagReportEntryId Property	46
2.2.2.20	PidTagReportName Property.....	46
2.2.2.21	PidTagReportSearchKey Property	46
2.2.2.22	PidTagReportTag Property	46
2.2.2.23	PidTagReportText Property.....	48
2.2.2.24	PidTagReadReceiptAddressType Property	48
2.2.2.25	PidTagReadReceiptEmailAddress Property	48
2.2.2.26	PidTagReadReceiptEntryId Property	49
2.2.2.27	PidTagReadReceiptName Property	49
2.2.2.28	PidTagReadReceiptSearchKey Property	49
2.2.3	E-Mail Submission Properties	49
2.2.3.1	PidTagRecipientType Property	49
2.2.3.2	PidTagDeferredSendNumber Property.....	50
2.2.3.3	PidTagDeferredSendUnits Property.....	50
2.2.3.4	PidTagDeferredSendTime Property	50
2.2.3.5	PidTagExpiryNumber Property	51
2.2.3.6	PidTagExpiryUnits Property	51
2.2.3.7	PidTagExpiryTime Property	51
2.2.3.8	PidTagDeleteAfterSubmit Property	51
2.2.3.9	PidTagMessageDeliveryTime Property.....	52
2.2.3.10	PidTagSentMailSvrEID Property	52
2.2.3.11	PidTagClientSubmitTime Property	52
2.2.4	Message Delivery ROPs	52
2.2.4.1	RopSubmitMessage ROP	52
2.2.4.1.1	RopSubmitMessage ROP Request Buffer	52
2.2.4.1.2	RopSubmitMessage ROP Response Buffer	53
2.2.4.2	RopAbortSubmit ROP	53
2.2.4.2.1	RopAbortSubmit ROP Request Buffer	53
2.2.4.2.2	RopAbortSubmit ROP Response Buffer	53
2.2.4.3	RopGetAddressTypes ROP.....	53
2.2.4.3.1	RopGetAddressTypes ROP Request Buffer.....	54

2.2.4.3.2	RopGetAddressTypes ROP Response Buffer	54
2.2.4.4	RopOptionsData ROP.....	54
2.2.4.4.1	RopOptionsData ROP Request Buffer	54
2.2.4.4.2	RopOptionsData ROP Response Buffer	54
2.2.5	Spooler and Transport ROPs	55
2.2.5.1	RopSetSpooler ROP.....	55
2.2.5.1.1	RopSetSpooler ROP Request Buffer	55
2.2.5.1.2	RopSetSpooler ROP Response Buffer	55
2.2.5.2	RopGetTransportFolder ROP.....	55
2.2.5.2.1	RopGetTransportFolder ROP Request Buffer.....	56
2.2.5.2.2	RopGetTransportFolder ROP Response Buffer.....	56
2.2.5.3	RopSpoolerLockMessage ROP	56
2.2.5.3.1	RopSpoolerLockMessage ROP Request Buffer.....	56
2.2.5.3.2	RopSpoolerLockMessage ROP Response Buffer.....	56
2.2.5.4	RopTransportSend ROP	56
2.2.5.4.1	RopTransportSend ROP Request Buffer	57
2.2.5.4.2	RopTransportSend ROP Response Buffer	57
2.2.5.5	RopTransportNewMail ROP.....	57
2.2.5.5.1	RopTransportNewMail ROP Request Buffer.....	57
2.2.5.5.2	RopTransportNewMail ROP Response Buffer.....	58
3	Protocol Details.....	59
3.1	Common Details	59
3.1.1	Abstract Data Model	59
3.1.1.1	Global	59
3.1.1.2	Per Mailbox.....	59
3.1.1.3	Per Message Object	59
3.1.1.4	Per Send State	60
3.1.2	Timers	60
3.1.3	Initialization	60
3.1.4	Higher-Layer Triggered Events.....	60
3.1.5	Message Processing Events and Sequencing Rules.....	60
3.1.6	Timer Events	60
3.1.7	Other Local Events	60
3.2	Client Details.....	60
3.2.1	Abstract Data Model	60
3.2.1.1	Global	61
3.2.1.2	Per Mailbox.....	61
3.2.1.3	Per Message Object	61
3.2.2	Timers	61
3.2.3	Initialization	61
3.2.4	Higher-Layer Triggered Events.....	61
3.2.4.1	Sending a Message	61
3.2.4.1.1	Sending the Message as a Represented Sender.....	61
3.2.4.1.2	Sending the Message as the Actual Sender.....	61
3.2.4.1.3	Sending the Message as the Sender Itself	61
3.2.4.1.4	Sending the Message on Behalf of Another Person.....	62
3.2.4.2	Deferring Message Send	62
3.2.4.3	Sending a Message with Expiry Time.....	62
3.2.4.4	Optimizing Send	62
3.2.4.5	Resending a Message.....	62
3.2.4.6	Soliciting Votes from Voters	62
3.2.4.6.1	Sending a Voting Message.....	63

3.2.4.6.2	Interpreting a Voting Message	63
3.2.4.6.3	Crafting a Voting Response Message.....	63
3.2.4.6.4	Aggregating Voting Responses	63
3.2.4.7	Sending Mail Through a Specific Server	63
3.2.4.8	Processing E-mail Objects in the Spooler Queue	64
3.2.4.9	Delivering Mail to the Server	64
3.2.5	Message Processing Events and Sequencing Rules.....	64
3.2.5.1	Sending a RopSubmitMessage ROP Request.....	64
3.2.5.2	Sending a RopAbortSubmit ROP Request	64
3.2.5.3	Sending a RopLockSpoolerMessage ROP Request.....	65
3.2.5.4	Sending a RopTransportNewMail ROP Request.....	65
3.2.6	Timer Events	65
3.2.7	Other Local Events	65
3.3	Server Details	65
3.3.1	Abstract Data Model	65
3.3.1.1	Global	65
3.3.1.2	Per Mailbox.....	65
3.3.1.3	Per Message Object	66
3.3.1.4	Per E-mail Object	66
3.3.1.5	User.....	66
3.3.2	Timers	66
3.3.3	Initialization	66
3.3.4	Higher-Layer Triggered Events.....	66
3.3.5	Message Processing Events and Sequencing Rules.....	67
3.3.5.1	Receiving a RopSubmitMessage ROP Request.....	67
3.3.5.1.1	Permission Check	67
3.3.5.1.2	Delivering Mail on a RopSubmitMessage or RopTransportSend Request.....	68
3.3.5.1.3	Properties Read and/or Set Upon Submission	68
3.3.5.1.3.1	Represented Sender Properties	69
3.3.5.1.3.2	Actual Sender Properties	69
3.3.5.1.3.3	Deferred Properties.....	69
3.3.5.1.3.4	Expiry Properties	69
3.3.5.1.4	Rule Processing.....	70
3.3.5.2	Receiving a RopAbortSubmit ROP Request	70
3.3.5.3	Receiving a RopGetAddressTypes ROP Request	70
3.3.5.4	Receiving a RopSetSpooler ROP Request	71
3.3.5.5	Receiving a RopGetTransportFolder ROP Request.....	71
3.3.5.6	Receiving a RopSpoolerLockMessage ROP Request.....	71
3.3.5.7	Receiving a RopTransportSend ROP Request	72
3.3.5.8	Receiving a RopTransportNewMail ROP Request.....	72
3.3.5.9	Receiving a RopOptionsData ROP Request	72
3.3.6	Timer Events	72
3.3.7	Other Local Events	72
4	Protocol Examples.....	73
4.1	Submitting a Message	73
4.1.1	ROP Request Buffer	73
4.1.2	ROP Response Buffer	73
4.2	Submitting a Deferred Message	74
4.2.1	ROP Request Buffer	74
4.2.2	ROP Response Buffer	74
4.3	Aborting a Message Submission.....	75
4.3.1	ROP Request Buffer	75

4.3.2	ROP Response Buffer	76
4.4	Sending an E-Mail Message from a Messaging User to Another Messaging User	76
4.5	Sending a Message with Voting Options	80
4.6	Sending Mail to a Specific Server	81
4.6.1	Initialization	82
4.6.1.1	ROP Request Buffer	82
4.6.1.2	ROP Response Buffer	82
4.6.2	Submitting the Message to the Spooler Queue Folder.....	82
4.6.3	Locking the Message in the Spooler Queue Folder for Processing	82
4.6.3.1	ROP Request Buffer	82
4.6.3.2	ROP Response Buffer	83
4.6.4	Determining the Transport Folder	83
4.6.4.1	ROP Request Buffer	83
4.6.4.2	ROP Response Buffer	83
4.6.5	Sending the Message	84
4.6.5.1	ROP Request Buffer	84
4.6.5.2	ROP Response Buffer	84
4.6.6	Marking the Message as Ready for Post-Send Server Processing	87
4.6.6.1	ROP Request Buffer	87
4.6.6.2	ROP Response Buffer	87
5	Security	88
5.1	Security Considerations for Implementers.....	88
5.2	Index of Security Parameters	88
6	Appendix A: Product Behavior	89
7	Change Tracking.....	91
8	Index	93

1 Introduction

The E-Mail Object Protocol enables the creation, transmission and storage of e-mail messages by representing e-mails as **Message objects**. The E-Mail Object Protocol extends the Message and Attachment Object Protocol in that it defines new properties and adds restrictions to the properties that are described in [\[MS-OXCMMSG\]](#).

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

- American National Standards Institute (ANSI) character set**
- ASCII**
- big-endian**
- Coordinated Universal Time (UTC)**
- Domain Name System (DNS)**
- flags**
- GUID**
- handle**
- Hypertext Transfer Protocol (HTTP)**
- language code identifier (LCID)**
- little-endian**
- remote procedure call (RPC)**
- Unicode**

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

- address book**
- address type**
- Attachment object**
- blind carbon copy (Bcc) recipient**
- body part**
- carbon copy (Cc) recipient**
- delivery receipt**
- display name**
- distribution list**
- E-mail object**
- EntryID**
- folder associated information (FAI)**
- Folder object**
- header**
- Hypertext Markup Language (HTML)**
- Internet Message Access Protocol - Version 4 (IMAP4)**
- locale**
- Logon object**
- mailbox**
- message body**
- message class**
- Message object**
- Message store**
- messaging object**

Multipurpose Internet Mail Extensions (MIME)
non-delivery report
non-read receipt
Object Linking and Embedding (OLE)
Out of Office (OOF)
permission
plain text
Post Office Protocol - Version 3 (POP3)
primary recipient
read receipt
recipient
recipient table
reminder
remote operation (ROP)
Rich Text Format (RTF)
ROP request
ROP request buffer
ROP response
ROP response buffer
search folder
search key
Sent Items folder
Server object
Simple Mail Transfer Protocol (SMTP)
spam
store
subject
To recipient
Transport Neutral Encapsulation Format (TNEF)
Uniform Resource Identifier (URI)

The following terms are specific to this document:

conversation thread: A series of messages and responses to those messages, typically related by subject.

mail spooler: A program or function that receives requests to send mail to and deliver mail for a user. It determines which mail transport handles sending or receiving mail.

message transfer agent (MTA): An SMTP server that accepts mail from a client or another MTA and delivers the mail or relays it to another MTA.

messaging transport: A networking protocol that facilitates the transfer of messages between a messaging client and a messaging server.

report message: A message that presents status information about a sent message. A report message is sent to the sender of the message.

resend message: A message that is submitted for message delivery after it failed to be sent to all or some of its recipients (1).

spooler queue: A series of outgoing messages that are ready for delivery to recipients (1).

UUEncoded attachment: A file that is attached to an e-mail message that was encoded by using the uuencode utility, as described in [IEEE1003.1].

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described in [\[RFC2119\]](#). All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

References to Microsoft Open Specification documents do not include a publishing year because links are to the latest version of the documents, which are updated frequently. References to other documents include a publishing year when one is available.

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, <http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624>, as an additional source.

[MS-DTYP] Microsoft Corporation, "[Windows Data Types](#)".

[MS-OXBBODY] Microsoft Corporation, "[Best Body Retrieval Algorithm](#)".

[MS-OXCDATA] Microsoft Corporation, "[Data Structures](#)".

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

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

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

[MS-OXCNOTIF] Microsoft Corporation, "[Core Notifications Protocol Specification](#)".

[MS-OXCPERM] Microsoft Corporation, "[Exchange Access and Operation Permissions Protocol Specification](#)".

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

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

[MS-OXCSPAM] Microsoft Corporation, "[Spam Confidence Level Protocol Specification](#)".

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

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

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

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

[MS-OXODLGT] Microsoft Corporation, "[Delegate Access Configuration Protocol Specification](#)".

[MS-OXOFLAG] Microsoft Corporation, "[Informational Flagging Protocol Specification](#)".

[MS-OXORMDR] Microsoft Corporation, "[Reminder Settings Protocol Specification](#)".

[MS-OXORULE] Microsoft Corporation, "[E-Mail Rules Protocol Specification](#)".

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

[RFC1321] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992, <http://www.ietf.org/rfc/rfc1321.txt>

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

[RFC2369] Neufeld, G., and Baer, J., "The Use of URLs as Meta-Syntax for Core Mail List Commands and their Transport through Message Header Fields", RFC 2369, July 1998, <http://www.rfc-editor.org/rfc/rfc2369.txt>

[RFC2821] Klensin, J., "Simple Mail Transfer Protocol", STD 10, RFC 2821, April 2001, <http://www.ietf.org/rfc/rfc2821.txt>

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

[RFC5234] Crocker, D., Ed., and Overell, P., "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008, <http://www.rfc-editor.org/rfc/rfc5234.txt>

1.2.2 Informative References

[IEEE1003.1] The Open Group, "IEEE Std 1003.1, 2004 Edition", 2004, http://www.unix.org/version3/ieee_std.html

Note Registration is required to view or download this specification.

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

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

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

[MS-OXOAB] Microsoft Corporation, "[Offline Address Book \(OAB\) File Format and Schema](#)".

1.3 Overview

An **E-mail object** represents a single e-mail message. The properties that are specific to an E-mail object facilitate retaining information about the e-mail message's sender, **recipients (1)**, subject, message content, and all the options associated with this e-mail that are set by the sender or recipient (1). An E-mail object is stored in a **Folder object**. This protocol also specifies how an E-mail object is used to represent a **report message**, which is a special type of message that is generated to report the status of a sent message, either at the sender's request or at the request of the system administrator.

1.3.1 E-Mail Objects

1.3.1.1 Creating, Opening, and Saving E-Mail Objects

E-mail objects adhere to the descriptions in [\[MS-OXCMSG\]](#).

1.3.1.2 Sending Messages

A client submits a request to a server to send an e-mail message to another messaging user. The server can defer or reject the request based on the properties and **permissions** that are associated with the E-mail object.

While the message is queued in the server, the client can abort the send operation.

1.3.1.3 Replying and Forwarding Messages

Replying to a message or forwarding a message is identical to sending a message except that both actions have an expanded set of properties. These properties are specified in section [2.2.1](#).

1.3.2 Report Messages

Report messages are an extension of the E-mail object. Report messages present status information about a sent message to its sender. The following are the two general types of reports:

- Read status reports. **Read receipt** reporting occurs when the sent e-mail message is read/opened by the recipient (1). **Non-read receipt** reporting occurs when the sent e-mail message is not read before it is deleted or expired.
- Delivery status reports. **Delivery receipt** reporting occurs when the sent e-mail message is delivered to the recipient (1). **Non-delivery report** reporting occurs when the sent e-mail message cannot be delivered.

1.3.2.1 Read Receipt

A read receipt report indicates that a sent e-mail message was read or opened by a recipient (1).

Read receipts are not generated automatically. Senders who want to receive read receipts explicitly request them.

1.3.2.2 Non-Read Receipt

A non-read receipt is generated during e-mail message deletion operations, as described in [\[MS-OXCFOOLD\]](#), at the expiration of a time limit or according to client-specific criteria. A non-read receipt is sent to the e-mail's sender or a designated recipient (1) by the e-mail sender's request.

1.3.2.3 Delivery Receipt

A delivery receipt is generated by an e-mail client or server and sent to the e-mail's sender or designated recipient (1) when an e-mail has reached its intended recipient (1).

1.3.2.4 Non-Delivery Report

A non-delivery report receipt is generated by an e-mail client or server and sent to the e-mail's sender when an e-mail could not reach an intended recipient (1). Non-delivery report receipts are sent automatically unless a request is made to suppress them.

1.3.3 Voting and Tracking

Voting and tracking capabilities are an extension of the E-mail object. A client can add voting options to an e-mail message through the use of voting verb properties, as specified in section [2.2.1.65](#). The client of a recipient (2) can respond to the voting survey by setting response

properties on a reply message. The sender's client processes the reply message and maintains the response tracking information in the original message's recipient (2) tracking status properties, as specified in section [2.2.1.66](#).

1.3.4 Controlling Sending and Delivery of Mail

If a client is connected to several e-mail servers at once (not necessarily using the same protocol), it can choose to control how mail is sent by manipulating the **spooler queue** of the **Message store**. If a client delivers mail into a folder on the server (such as delivering **Post Office Protocol - Version 3 (POP3)** messages), it can inform the server of the new mail through **remote operation (ROP)** requests.

1.4 Relationship to Other Protocols

The E-Mail Object Protocol has the same dependencies as the Message and Attachment Object Protocol, as described in [\[MS-OXCMMSG\]](#).

1.5 Prerequisites/Preconditions

The E-Mail Object Protocol has the same prerequisites and preconditions as the Message and Attachment Object Protocol, as described [\[MS-OXCMMSG\]](#).

1.6 Applicability Statement

The E-Mail Object Protocol is designed to facilitate the exchange of interpersonal mail and messages.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

The E-Mail Object Protocol uses the protocols specified in [\[MS-OXCPRPT\]](#) and [\[MS-OXCMSG\]](#) as its primary transport mechanism.

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

2.2 Message Syntax

An E-mail object can be created and modified by clients and servers. Except where noted, this section defines constraints to which both clients and servers adhere when operating on E-mail objects.

Clients operate on E-mail objects by using the Message and Attachment Object Protocol, which is specified in [\[MS-OXCMSG\]](#). How a server operates on E-mail objects is implementation-dependent, but the results of any such operations are to be exposed to clients in a manner that is consistent with the E-Mail Object Protocol.

Unless otherwise specified, E-mail objects adhere to all property constraints specified in [\[MS-OXPROPS\]](#) and all property constraints specified in [\[MS-OXCMSG\]](#). An E-mail object can also contain other properties, as specified 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.

Message senders are identified by the from properties and the sender properties on an E-mail object. In general, the from properties and the sender properties will identify the same messaging user; for example, the e-mail message appears to have been sent by the actual sender of the e-mail message. In some cases, however, an e-mail message is sent by one user (the actual sender) on behalf of another user (the represented sender). In this case, the from properties identify the represented sender, and the sender properties identify the actual sender.

Message object properties can be considered as belonging to certain groups based on the type of messaging sub-object they represent. The first four groups represent actual senders, represented senders, represented recipients, and actual recipients. Other properties correspond to the Body and Subject sub-objects of a message. A separate class of properties is used to specify the **To recipients**, Carbon copy (cc) recipients, and Blind carbon copy (bcc) recipients of an e-mail message. The remaining properties that do not fall under these groups are used to specify either other **subobjects** related to message management, or to control the method or timing of message delivery.

An actual recipient is the owner of the **mailbox** that receives the e-mail message. The following properties are associated with actual recipients:

- **PidTagMessageRecipientMe** (section [2.2.1.19](#))
- **PidTagReceivedByAddressType** (section [2.2.1.29](#))
- **PidTagReceivedByEmailAddress** (section [2.2.1.30](#))
- **PidTagReceivedByEntryId** (section [2.2.1.31](#))

- **PidTagReceivedByName** (section [2.2.1.32](#))
- **PidTagReceivedBySearchKey** (section [2.2.1.33](#))
- **PidTagRecipientType** (section [2.2.3.1](#))

The represented sender of an e-mail message is the messaging user or user agent on whose behalf the e-mail message was sent (or will be sent). The following from properties are associated only with the represented sender:

- **PidTagSentRepresentingAddressType** (section [2.2.1.46](#))
- **PidTagSentRepresentingEmailAddress** (section [2.2.1.47](#))
- **PidTagSentRepresentingEntryId** (section [2.2.1.48](#))
- **PidTagSentRepresentingName** (section [2.2.1.49](#))
- **PidTagSentRepresentingSearchKey** (section [2.2.1.50](#))
- **PidTagOriginalSentRepresentingAddressType** (section [2.2.2.11](#))
- **PidTagOriginalSentRepresentingEmailAddress** (section [2.2.2.12](#))
- **PidTagOriginalSentRepresentingEntryId** (section [2.2.2.13](#))
- **PidTagOriginalSentRepresentingName** (section [2.2.2.14](#))
- **PidTagOriginalSentRepresentingSearchKey** (section [2.2.2.15](#))

The actual sender is the owner of the mailbox that sent (or will send) the e-mail message. The following from properties are associated with the actual sender:

- **PidTagSenderAddressType** (section [2.2.1.40](#))
- **PidTagSenderEmailAddress** (section [2.2.1.41](#))
- **PidTagSenderEntryId** (section [2.2.1.42](#))
- **PidTagSenderName** (section [2.2.1.43](#))
- **PidTagSenderSearchKey** (section [2.2.1.44](#))
- **PidTagOriginalSenderAddressType** (section [2.2.2.6](#))
- **PidTagOriginalSenderEmailAddress** (section [2.2.2.7](#))
- **PidTagOriginalSenderEntryId** (section [2.2.2.8](#))
- **PidTagOriginalSenderName** (section [2.2.2.9](#))
- **PidTagOriginalSenderSearchKey** (section [2.2.2.10](#))

The recipients (2) subobject is a collection of recipients (2), each of which is a messaging user to whom e-mail messages will be (or have been) delivered. As with senders, there are two types of recipients (2): represented recipients and actual recipients. Within each of these types, there are three subclasses of recipients (2) for an e-mail message: To recipients, **Cc recipients**, and **Bcc recipients**.

A represented recipient is the messaging user or user agent on whose behalf the e-mail message is being received. The following recipient properties are associated with represented recipients:

- **PidTagReceivedRepresentingAddressType** (section [2.2.1.23](#))
- **PidTagReceivedRepresentingEmailAddress** (section [2.2.1.24](#))
- **PidTagReceivedRepresentingEntryId** (section [2.2.1.25](#))
- **PidTagReceivedRepresentingName** (section [2.2.1.26](#))
- **PidTagReceivedRepresentingSearchKey** (section [2.2.1.27](#))

Another set of from properties is used to identify three subclasses of recipients (2) for an e-mail message: To recipients, Cc recipients, and Bcc recipients.

The following from properties are associated with To recipients:

- **PidTagDisplayTo** (section [2.2.1.9](#))
- **PidTagMessageToMe** (section [2.2.1.17](#))
- **PidTagOriginalDisplayTo** (section [2.2.2.3](#))

The following from properties are associated with Cc recipients:

- **PidTagDisplayCc** (section [2.2.1.8](#))
- **PidTagMessageCcMe** (section [2.2.1.18](#))
- **PidTagOriginalDisplayCc** (section [2.2.2.4](#))

The following from properties are associated with Bcc recipients:

- **PidTagDisplayBcc** (section [2.2.1.7](#))
- **PidTagOriginalDisplayBcc** (section [2.2.2.5](#))

The Subject subobject is a short text string that is intended to inform a recipient (1) as to the contents or purpose of the e-mail message. The following properties are associated with the subject:

- **PidTagNormalizedSubject** ([\[MS-OXCMSG\]](#) section 2.2.1.10)
- **PidTagSubjectPrefix** (section [2.2.1.51](#))
- **PidTagOriginalSubject** (section [2.2.2.16](#))

The Body subobject, as specified in [\[MS-OXBBODY\]](#), contains the main contents of the e-mail message. The following properties are associated with the body:

- **PidTagBlockStatus** (section [2.2.1.1](#))
- **PidTagBody** ([\[MS-OXCMSG\]](#) section 2.2.1.48.1)
- **PidTagBodyHtml** ([\[MS-OXCMSG\]](#) section 2.2.1.48.3)
- **PidTagRtfCompressed** ([\[MS-OXCMSG\]](#) section 2.2.1.48.4)
- **PidTagRtfInSync** ([\[MS-OXCMSG\]](#) section 2.2.1.48.5)

- **PidTagMessageEditorFormat** (section [2.2.1.69](#))

Many properties that are not associated with the preceding core E-mail objects are included with an e-mail message in support of other particular subobjects. The following subobjects, along with their associated properties, fall into this category:

- Conversations
 - **PidTagConversationIndex** ([\[MS-OXOCFG\]](#) section 2.2.8.7)
 - **PidTagConversationTopic** (section [2.2.1.5](#))

If an e-mail message in the **conversation thread** is given a new subject, this e-mail message starts the new conversation thread with a new value for both the **PidTagConversationTopic** and **PidTagConversationIndex** properties.

- Client Options
 - **PidTagIconIndex** (section [2.2.1.10](#))
 - **PidTagMessageClass** ([\[MS-OXOCAL\]](#) section 2.2.1.49)
 - **PidTagReadReceiptRequested** (section [2.2.1.28](#))
 - **PidTagReadReceiptEntryId** (section [2.2.2.26](#))
 - **PidTagReadReceiptSearchKey** (section [2.2.2.28](#))
 - **PidTagOriginalSensitivity** (section [2.2.1.22](#))
 - **PidTagRecipientReassignmentProhibited** (section [2.2.1.34](#))
 - **PidTagReplyRequested** (section [2.2.1.37](#))
 - **PidTagResponseRequested** (section [2.2.1.38](#))
 - **PidTagReplyRecipientEntries** (section [2.2.1.35](#))
 - **PidTagReplyRecipientNames** (section [2.2.1.36](#))
 - **PidLidAutoProcessState** (section [2.2.1.64](#))
 - **PidLidVerbStream** (section [2.2.1.65](#))
 - **PidLidVerbResponse** (section [2.2.1.66](#))

Finally, the following properties are set by an e-mail client or server to control how messages are delivered:

- **PidTagExpiryTime** (section [2.2.3.7](#))
- **PidTagInternetMessageId** (section [2.2.1.12](#))
- **PidTagOriginatorDeliveryReportRequested** (section [2.2.1.20](#))
- **PidTagOriginatorNonDeliveryReportRequested** (section [2.2.1.21](#))
- **PidTagSendRichInfo** ([\[MS-EXOABK\]](#) section 2.2.3.18)

- **PidTagTransportMessageHeaders** (section [2.2.1.52](#))
- **PidTagOriginalDeliveryTime** (section [2.2.2.2](#))
- **PidTagOriginalSubmitTime** (section [2.2.2.17](#))
- **PidTagParentKey** (section [2.2.2.18](#))
- **PidTagReportTag** (section [2.2.2.22](#))
- **PidTagReportText** (section [2.2.2.23](#))
- **PidTagMessageFlags** ([\[MS-OXCMMSG\]](#) section 2.2.1.6)
- **PidTagMessageDeliveryTime** (section [2.2.3.9](#))
- **PidTagDeferredSendNumber** (section [2.2.3.2](#))
- **PidTagDeferredSendUnits** (section [2.2.3.3](#))
- **PidTagDeferredSendTime** (section [2.2.3.4](#))
- **PidTagExpiryNumber** (section [2.2.3.5](#))
- **PidTagExpiryUnits** (section [2.2.3.6](#))

2.2.1 E-Mail Object Properties

The properties in the sub-sections of this section are specific to **E-mail objects**.

2.2.1.1 PidTagBlockStatus Property

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

The **PidTagBlockStatus** property ([\[MS-OXPROPS\]](#) section 2.695) indicates the user's preference for viewing external content (such as links to images on a **Hypertext Transfer Protocol (HTTP)** server) in the **message body (2)**. Valid values for this property are given in the following table.

Value	Meaning
0x0	Default value. Block external content.
Variable	Allow or block external content, as described following this table. Nonzero values of this property are calculated as described following this table.

A client can ignore this value and always allow or always block external content based on other factors (such as whether the sender is on a safe list). If this property is used, the default (0x0) is to block the external content. However, if the value of this property falls within a certain range, viewing external content is allowed. The allowed value is computed from the **PidTagMessageDeliveryTime** property (section [2.2.3.9](#)): because the sender of a message does not have knowledge of this value, the sender cannot reliably set the value of the **PidTagBlockStatus** property to the allowed values.

To compute the allowed values, convert the value of the **PidTagMessageDeliveryTime** property to a **PtypFloatingTime** ([\[MS-OXCDATA\]](#) section 2.11.1) type (floatdate), where the date is represented as the number of days from 00:00:00, December 30, 1899, **Coordinated Universal Time (UTC)**. Apply the following formula.

result = ((floatdate - floor(floatdate)) * 100000000) + 3;

where floor(x) returns the largest integer $\leq x$.

Convert the value result to a 32-bit integer computed value.

When a client first receives the message, it SHOULD set the value of the **PidTagBlockStatus** property to this computed value to allow external content. However, when determining whether to accept external content, clients SHOULD allow external content if the absolute value of the difference between the computed value and the value of the **PidTagBlockStatus** property is 1 or less. After the message is received and the value of the **PidTagBlockStatus** property has been calculated, clients SHOULD persist the value of this property for future reference.

The server MUST NOT alter the value of this property.

2.2.1.2 PidTagConversationId Property

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

The **PidTagConversationId** property [<1>](#) ([\[MS-OXPROPS\]](#) section 2.728) is a computed value, derived from other conversation-related properties, that identifies a message as belonging to a specific conversation. This property is computed by the application, server or client. The computed value of the **PidTagConversationId** property SHOULD be derived from the values of the following properties.

If the value of the **PidTagConversationIndexTracking** property (section [2.2.1.4](#)) is set to **TRUE**, and the value of the **PidTagConversationIndex** property (section [2.2.1.3](#)) is at least 22 bytes long and the first byte of the value of the **PidTagConversationIndex** property is 0x01, then the value of the **PidTagConversationId** property MUST be the **GUID** portion of the **PidTagConversationIndex** property.

Otherwise, if the **PidTagConversationTopic** property (section [2.2.1.5](#)) is set, the value of the **PidTagConversationId** property MUST be computed as follows:

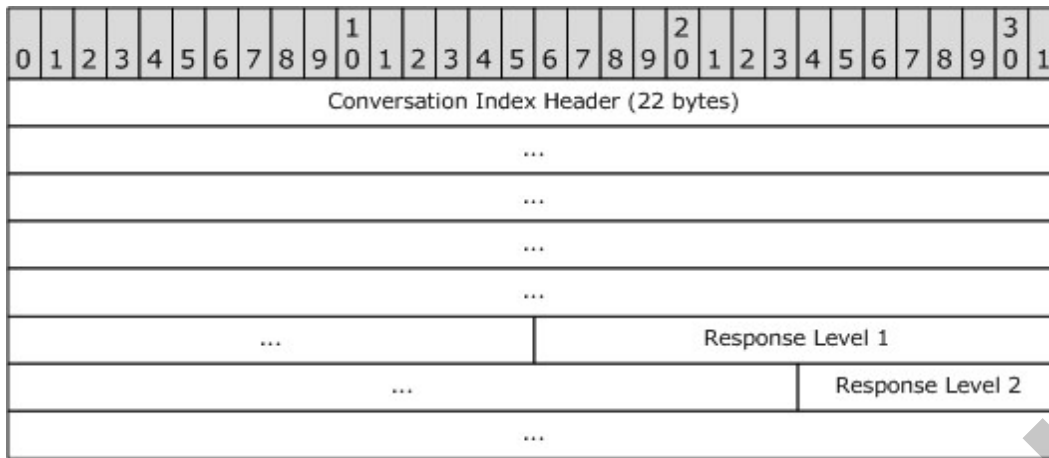
1. The application MUST use up to 255 of the first nonzero characters of the **little-endian** UTF-16 representation of the **PidTagConversationTopic** property.
2. The application MUST convert the characters to their upper-case forms, always mapping "i" to "I" regardless of the user's **locale**.
3. The application MUST perform an MD5 hash, as specified in [\[RFC1321\]](#), on the characters and use the resulting 16-byte hash as the value of the **PidTagConversationId** property.

Otherwise, if none of the above conditions were met, the **PidTagConversationId** property MUST NOT be set, in which case it will be undefined for the Message object.

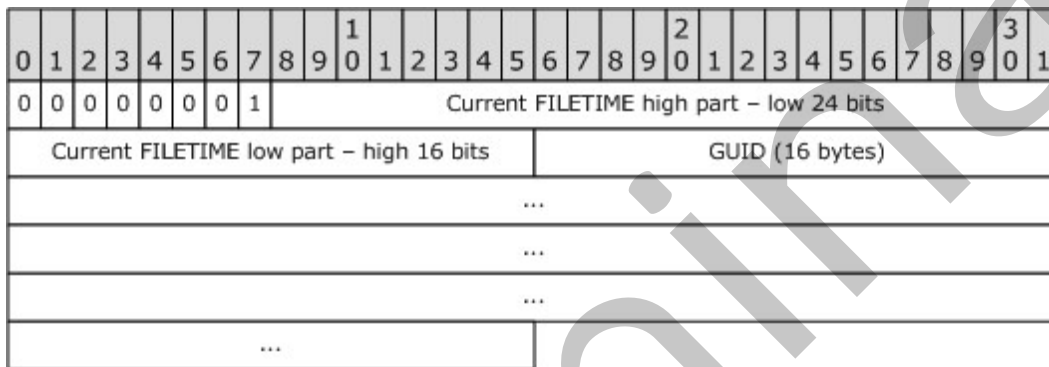
2.2.1.3 PidTagConversationIndex Property

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

The **PidTagConversationIndex** property ([\[MS-OXPROPS\]](#) section 2.729) indicates the relative position of this message within a conversation thread. It is set according to the description in the following diagram.



Conversation Index Header (22 bytes): Set according to the description in the following diagram.



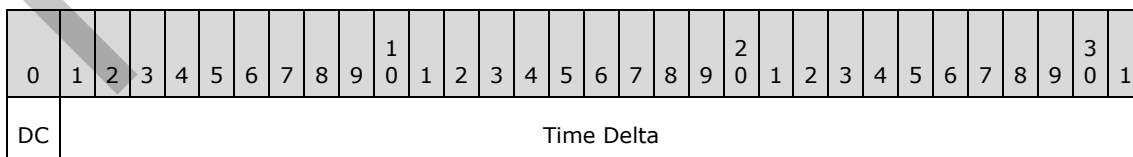
Current FILETIME (40 bits): The current time in **UTC** expressed as a **PtypTime** type ([\[MS-OXCDATA\]](#) section 2.11.1) is obtained, where only the 24 low bits of the high part and the 16 high bits of the low part of the **FILETIME** ([\[MS-DTYP\]](#)) are included in **Current FILETIME high part** and **Current FILETIME low part**, as shown in the following table.

Eight most significant bits	40 bits	16 least significant bits
Excluded	Included	Excluded

The data is stored in **big-endian** format: the five bytes of the time are written from most significant byte to least significant byte.

GUID (16 bytes, PtypGuid): Generated for each new conversation thread. The **Data1**, **Data2**, and **Data3** fields are stored in big-endian format in the packet.

Response Levels (5 bytes each): Set according to the description in the following diagram.



Random

DC (Delta code) (1 bit) and Time Delta (31 bits): Calculated based on the difference between the current time and the time stored in the conversation index header:

- If the difference is less than 1.7 years (high order part of the delta file time bitwise AND with 0x00FE0000 resulting in "0"), the **Delta Code** field is 0 and the **Time Delta** field is the least significant 31 bits of the difference remaining after the 18 least significant bits are excluded.

15 most significant bits	31 bits	18 least significant bits
Excluded	Included	Excluded

- If the difference is greater than or equal to 1.7 years (high order part of the delta file time bitwise AND with 0x00FE0000 resulting in nonzero), the **Delta Code** field is 1 and the **Time Delta** field is the least significant 31 bits of the difference remaining after the 23 least significant bits are excluded.

10 most significant bits	31 bits	23 least significant bits
Excluded	Included	Excluded

For both cases, **Time Delta** is stored in big-endian format.

Random (8 bits): Random value generated by using an implementation-specific algorithm.

2.2.1.4 PidTagConversationIndexTracking Property

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

The **PidTagConversationIndexTracking** property ([\[MS-OXPROPS\]](#) section 2.730) is set to **TRUE** if the GUID portion of the value of the **PidTagConversationIndex** property (section [2.2.1.3](#)) will be used to compute the value of the **PidTagConversationId** property (section [2.2.1.2](#)), assuming the client or server application implements the **PidTagConversationId** property. Otherwise, this property is set to **FALSE**.

2.2.1.5 PidTagConversationTopic Property

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

The **PidTagConversationTopic** property ([\[MS-OXPROPS\]](#) section 2.731) contains an unchanging copy of the original subject. <2> The property is set to the same value as the **PidTagNormalizedSubject** property ([\[MS-OXCMSG\]](#) section 2.2.1.10) on an E-mail object when it is submitted.

2.2.1.6 PidTagDeferredDeliveryTime Property

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

The **PidTagDeferredDeliveryTime** property ([\[MS-OXPROPS\]](#) section 2.742) contains the date and time, in UTC, at which the sender prefers the message to be delivered. This property MAY be absent; if it is absent, the message is delivered as soon as possible. If it is present, the property SHOULD have the same value as the **PidTagDeferredSendTime** property (section [2.2.3.4](#)).

A client sets both the **PidTagDeferredDeliveryTime** property and the **PidTagDeferredSendTime** property for deferred delivery of a message before submission. <3>

2.2.1.7 PidTagDisplayBcc Property

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

The **PidTagDisplayBcc** property ([\[MS-OXPROPS\]](#) section 2.752) is set to a list of Bcc recipient **display names**, separated by semicolons, if an e-mail message has Bcc recipients. Otherwise, this property contains an empty string, as specified in [\[MS-OXCMSG\]](#) section 3.2.5.2. This property is read-only for the client.

2.2.1.8 PidTagDisplayCc Property

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

The **PidTagDisplayCc** property ([\[MS-OXPROPS\]](#) section 2.753) is set to a list of Cc recipient display names, separated by semicolons, if an e-mail message has Cc recipients. Otherwise, this property contains an empty string, as specified in [\[MS-OXCMSG\]](#) section 3.2.5.2. This property is read-only for the client.

2.2.1.9 PidTagDisplayTo Property

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

The **PidTagDisplayTo** property ([\[MS-OXPROPS\]](#) section 2.756) is set to a list of the **primary recipient** display names, separated by semicolons, if an e-mail message has primary recipients. Otherwise, this property contains an empty string, as specified in [\[MS-OXCMSG\]](#) section 3.2.5.2. This property is read-only for the client.

2.2.1.10 PidTagIconIndex Property

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

The **PidTagIconIndex** property ([\[MS-OXPROPS\]](#) section 2.813) specifies the icon to be used by the user interface when displaying a group of E-mail objects. This property, if it exists, is a hint to the client: it can ignore the value of this property and use another method of determining what icon to display to the user, such as using the values of the **PidTagMessageClass** property ([\[MS-OXCMSG\]](#) section 2.2.1.3) or the **PidTagMessageFlags** property ([\[MS-OXCMSG\]](#) section 2.2.1.6). Examples of values for the **PidTagIconIndex** property are shown in the following table.

Mail item state	Mail item icon index
New mail	0xFFFFFFFF
Read mail	0x00000100
Unread mail	0x00000101
Submitted mail	0x00000102
Unsent mail	0x00000103
Receipt mail	0x00000104
Replied mail	0x00000105

Mail item state	Mail item icon index
Forwarded mail	0x00000106
Remote mail	0x00000107
Delivery receipt	0x00000108
Read receipt	0x00000109
Non-delivery report	0x0000010A
Non-read receipt	0x0000010B
Recall_S mails	0x0000010C
Recall_F mail	0x0000010D
Tracking mail	0x0000010E
Out of Office mail	0x0000011B
Recall mail	0x0000011C
Tracked mail	0x00000130

2.2.1.11 PidTagInternetMailOverrideFormat Property

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

The **PidTagInternetMailOverrideFormat** property ([\[MS-OXPROPS\]](#) section 2.824) indicates the encoding method and **Hypertext Markup Language (HTML)** inclusion for attachments and SHOULD be set on an outgoing e-mail message. This property is broken up into subportions, as shown in the following table. Note that "X" indicates that the bit is not to be set, and if set, the bit is to be ignored; the format of the diagram is little-endian.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	
x	x	x	x	x	Format 1			x	x	x	x	x	x	x	x	x	x	x	x	x	E18	M 4	P 2	x	x	x	x	x	x	x	x	x

Format1 (3 bits): Set to one of the values listed in the following table.

Value	Meaning
0x0	Default value. The mail system chooses the default encoding scheme, based on other fields in this property value.
0x1	The message is sent in MIME format with text/plain and text/HTML body parts .
0x2	The message is sent as plain text with UUEncoded attachments .
0x4	The message is sent in MIME format with text/plain and text/HTML body parts. This value is treated the same as the 0x1 value.

E18 (2 bits): Ignored if **Format1** = 0 or **P2** = 0 or **M4** = 0. Otherwise, set to one of the following values to indicate the HTML inclusion.

Value	Meaning
0x0	Text/plain only.
0x1	Text/plain and text/HTML.
0x2	Text/plain and text/HTML. This value is treated the same as the 0x1 value.

M4 (1 bit): Ignored if **Format1** = 0 or **P2** = 0; otherwise, indicates the encoding, as shown in the following table.

Value	Meaning
0	Use the uuencode algorithm as described in [IEEE1003.1] , and ignore the value of the E18 field.
1	Use MIME encoding, and use the value of the E18 field to determine body inclusions.

P2 (1 bit): Ignored if **Format1** = 0; otherwise, indicates the preference, as shown in the following table.

Value	Meaning
0	Ignore the value of the M4 field.
1	Use the value of the M4 field to determine encoding.

2.2.1.12 PidTagInternetMessageId Property

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

The **PidTagInternetMessageId** property ([\[MS-OXPROPS\]](#) section 2.825) corresponds to the **Message-id** field, as specified in [\[RFC2822\]](#). This property SHOULD be present on all e-mail messages. More details about the conversion between this property and the **Message-id** field are specified in [\[MS-OXCMAIL\]](#) section 2.1.3.2.11.

2.2.1.13 PidTagInReplyToId Property

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

The **PidTagInReplyToId** property ([\[MS-OXPROPS\]](#) section 2.818) corresponds to the **in-reply-to** field, as specified in [\[RFC2822\]](#), and contains the value of the original message's **PidTagInternetMessageId** property (section [2.2.1.12](#)). This property is set on all message replies.

2.2.1.14 PidTagLastVerbExecuted Property

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

The **PidTagLastVerbExecuted** property ([\[MS-OXPROPS\]](#) section 2.844) specifies the last verb executed for the message item to which it is related. This property is used by the client to display the last operation performed on the item. The possible values for the **PidTagLastVerbExecuted** property are shown in the following table.

Verb name	Alternate name	Value
Open	NOTEIVERB_OPEN	0x00000000
ReplyToSender	NOTEIVERB_REPLYTOSENDER	0x00000102
ReplyToAll	NOTEIVERB_REPLYTOALL	0x00000103
Forward	NOTEIVERB_FORWARD	0x00000104
Print	NOTEIVERB_PRINT	0x00000105
Save as	NOTEIVERB_SAVEAS	0x00000106
ReplyToFolder	NOTEIVERB_REPLYTOFOLDER	0x00000108
Save	NOTEIVERB_SAVE	0x00000500
Properties	NOTEIVERB_PROPERTIES	0x00000510
Followup	NOTEIVERB_FOLLOWUP	0x00000511
Accept	NOTEIVERB_ACCEPT	0x00000512
Tentative	NOTEIVERB_TENTATIVE	0x00000513
Reject	NOTEIVERB_REJECT	0x00000514
Decline	NOTEIVERB_DECLINE	0x00000515
Invite	NOTEIVERB_INVITE	0x00000516
Update	NOTEIVERB_UPDATE	0x00000517
Cancel	NOTEIVERB_CANCEL	0x00000518
SilentInvite	NOTEIVERB_SILENTINVITE	0x00000519
SilentCancel	NOTEIVERB_SILENTCANCEL	0x00000520
RecallMessage	NOTEIVERB_RECALL_MESSAGE	0x00000521
ForwardResponse	NOTEIVERB_FORWARD_RESPONSE	0x00000522
ForwardCancel	NOTEIVERB_FORWARD_CANCEL	0x00000523
FollowupClear	NOTEIVERB_FOLLOWUPCLEAR	0x00000524
ForwardAppointment	NOTEIVERB_FORWARD_APPT	0x00000525
OpenResend	NOTEIVERB_OPENRESEND	0x00000526
StatusReport	NOTEIVERB_STATUSREPORT	0x00000527
JournalOpen	NOTEIVERB_JOURNALOPEN	0x00000528
JournalOpenLink	NOTEIVERB_JOURNALOPENLINK	0x00000529
ComposeReplace	NOTEIVERB_COMPOSEREPLACE	0x00000530
Edit	NOTEIVERB_EDIT	0x00000531

Verb name	Alternate name	Value
DeleteProcess	NOTEIVERB_DELETEPROCESS	0x00000532
TentativeAppointmentTime	NOTEIVERB_TENTPNTIME	0x00000533
EditTemplate	NOTEIVERB_EDITTEMPLATE	0x00000534
FindInCalendar	NOTEIVERB_FINDINCALENDAR	0x00000535
ForwardAsFile	NOTEIVERB_FORWARDASFILE	0x00000536
ChangeAttendees	NOTEIVERB_CHANGE_ATTENDEES	0x00000537
RecalculateTitle	NOTEIVERB_RECALC_TITLE	0x00000538
PropertyChange	NOTEIVERB_PROP_CHANGE	0x00000539
ForwardAsVcal	NOTEIVERB_FORWARD_AS_VCAL	0x00000540
ForwardAsIcal	NOTEIVERB_FORWARD_AS_ICAL	0x00000541
ForwardAsBusinessCard	NOTEIVERB_FORWARD_AS_BCARD	0x00000542
DeclineAppointmentTime	NOTEIVERB_DECLPNTIME	0x00000543
Process	NOTEIVERB_PROCESS	0x00000544
OpenWithWord	NOTEIVERB_OPENWITHWORD	0x00000545
OpenInstanceOfSeries	NOTEIVERB_OPEN_INSTANCE_OF_SERIES	0x00000546
FilloutThisForm	NOTEIVERB_FILLOUT_THIS_FORM	0x00000547
FollowupDefault	NOTEIVERB_FOLLOWUP_DEFAULT	0x00000548
ReplyWithMail	NOTEIVERB_REPLY_WITH_MAIL	0x00000549
ToDoToday	NOTEIVERB_TODO_TODAY	0x00000566
ToDoTomorrow	NOTEIVERB_TODO_TOMORROW	0x00000567
ToDoThisWeek	NOTEIVERB_TODO_THISWEEK	0x00000568
ToDoNextWeek	NOTEIVERB_TODO_NEXTWEEK	0x00000569
ToDoThisMonth	NOTEIVERB_TODO_THISMONTH	0x00000570
ToDoNextMonth	NOTEIVERB_TODO_NEXTMONTH	0x00000571
ToDoNoDate	NOTEIVERB_TODO_NODATE	0x00000572
FollowupComplete	NOTEIVERB_FOLLOWUPCOMPLETE	0x00000573
CopyToPostFolder	NOTEIVERB_COPYTOPOSTFOLDER	0x00000574
SeriesInvitationUpdateToPartialAttendeeList	NOTEIVERB_PARTIALRECIP_SILENTINVITE	0x00000575
SeriesCancellationUpdateToPartialAttendeeList	NOTEIVERB_PARTIALRECIP_SILENTCANCEL	0x00000576

2.2.1.15 PidTagLastVerbExecutionTime Property

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

The **PidTagLastVerbExecutionTime** property ([\[MS-OXPROPS\]](#) section 2.845) contains the date and time, in UTC, during which the operation represented by the value of the **PidTagLastVerbExecuted** property (section [2.2.1.14](#)) took place.

2.2.1.16 PidTagMessageClass Property

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

The **PidTagMessageClass** property ([\[MS-OXCMMSG\]](#) section 2.2.1.3) contains the object type classification. This property is set to "IPM.Note" on E-mail objects. The value of the **PidTagMessageClass** property for report objects is specified in section [2.2.2.1](#) of this document.

2.2.1.17 PidTagMessageToMe Property

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

The **PidTagMessageToMe** property ([\[MS-OXPROPS\]](#) section 2.876) is an optional property indicating that the receiving mailbox owner is one of the primary recipients of an e-mail message. If this property is present, it is set to either 0x01, in which case, the receiving mailbox owner is specifically named as a primary recipient of an e-mail message and is not part of a **distribution list**; or 0x00, in which case the receiving mailbox owner is not a primary recipient of an e-mail message. The default is 0x00.

2.2.1.18 PidTagMessageCcMe Property

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

The **PidTagMessageCcMe** property ([\[MS-OXPROPS\]](#) section 2.862) is an optional property indicating that the receiving mailbox owner is a Cc recipient of an e-mail message. If this property is present, it is set to either 0x01, in which case the receiving mailbox owner is specifically named as a Cc recipient of an e-mail message and is not part of a distribution list; or 0x00, in which case the receiving mailbox owner is not a Cc recipient of an e-mail message. The default is 0x00.

2.2.1.19 PidTagMessageRecipientMe Property

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

The **PidTagMessageRecipientMe** property ([\[MS-OXPROPS\]](#) section 2.870) is an optional property indicating that the receiving mailbox owner is a primary or a Cc recipient of an e-mail message. If this property is present, it is set to either 0x01, in which case the receiving mailbox owner is specifically named as a primary or a Cc recipient of an e-mail message and is not part of a distribution list, or 0x00, in which case the receiving mailbox owner is not a primary and not a Cc recipient of an e-mail message. The default is 0x00.

2.2.1.20 PidTagOriginatorDeliveryReportRequested Property

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

The **PidTagOriginatorDeliveryReportRequested** property ([\[MS-OXPROPS\]](#) section 2.916) indicates whether an e-mail sender requests an e-mail delivery receipt from an e-mail client or server. This property is set to either 0x01, in which case the sender requests the delivery report be

sent to the e-mail sender or designated report receiver when the e-mail message is delivered, or 0x00 if the e-mail sender does not want to receive the delivery receipt.

2.2.1.21 PidTagOriginatorNonDeliveryReportRequested Property

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

The **PidTagOriginatorNonDeliveryReportRequested** property ([\[MS-OXPROPS\]](#) section 2.917) specifies whether an e-mail sender requests suppression of non-delivery reports. If this property is absent, the server automatically generates and sends a non-delivery report to the e-mail sender. If this property is present, it is set to either 0x00, in which case the e-mail sender requests suppression of non-delivery reports, or 0x01, in which case the non-delivery report is generated and sent.

2.2.1.22 PidTagOriginalSensitivity Property

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

The **PidTagOriginalSensitivity** property ([\[MS-OXPROPS\]](#) section 2.908) contains the sensitivity value of the original e-mail message. This property is set on replying and forwarding e-mail messages by using the value of the **PidTagSensitivity** property ([\[MS-OXCMSG\]](#) section 2.2.1.13) of the original message.

2.2.1.23 PidTagReceivedRepresentingAddressType Property

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

The **PidTagReceivedRepresentingAddressType** property ([\[MS-OXPROPS\]](#) section 2.965) contains the e-mail **address type** for the end user represented by the receiving mailbox owner, as specified in the **AddressType** field of the **RecipientRow** structure (section 2.2.4.3 and [\[MS-OXCDATA\]](#) section 2.8.3.2). If the receiving mailbox owner receives the e-mail message on his or her own behalf, this property is set to the value of the **PidTagReceivedByAddressType** property (section 2.2.1.29).

2.2.1.24 PidTagReceivedRepresentingEmailAddress Property

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

The **PidTagReceivedRepresentingEmailAddress** property ([\[MS-OXPROPS\]](#) section 2.966) contains the e-mail address for the end user represented by the receiving mailbox owner, as specified in the **EmailAddress** field of the **RecipientRow** structure ([\[MS-OXCDATA\]](#) section 2.8.3.2). If the receiving mailbox owner receives the e-mail message on his or her own behalf, this property is set to the value of the **PidTagReceivedByEmailAddress** property (section 2.2.1.30).

2.2.1.25 PidTagReceivedRepresentingEntryId Property

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

The **PidTagReceivedRepresentingEntryId** property ([\[MS-OXPROPS\]](#) section 2.967) contains an **address book EntryID** that identifies the end user represented by the receiving mailbox owner, as specified in the **EntryID** field of the **RecipientRow** structure ([\[MS-OXCDATA\]](#) section 2.8.3.2). If the receiving mailbox owner receives the e-mail message on his or her own behalf, this property is set to the value of the **PidTagReceivedByEntryId** property (section 2.2.1.31).

2.2.1.26 PidTagReceivedRepresentingName Property

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

The **PidTagReceivedRepresentingName** property ([\[MS-OXPROPS\]](#) section 2.969) contains the display name for the end user represented by the receiving mailbox owner, as specified by the **DisplayName** field of the **RecipientRow** structure ([\[MS-OXCDATA\]](#) section 2.8.3.2). If the receiving mailbox owner receives the e-mail on his or her own behalf, this property is set to the value of the **PidTagReceivedByName** property (section [2.2.1.32](#)).

2.2.1.27 PidTagReceivedRepresentingSearchKey Property

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

The **PidTagReceivedRepresentingSearchKey** property ([\[MS-OXPROPS\]](#) section 2.970) identifies an address book **search key** that contains a binary-comparable key of the end user represented by the receiving mailbox owner, as specified by the **SearchKey** field of the **RecipientRow** structure ([\[MS-OXCDATA\]](#) section 2.8.3.2). This property is computed in the same way that the value of the **PidTagReceivedBySearchKey** property (section [2.2.1.33](#)) is computed. If the receiving mailbox owner receives the e-mail message on his or her own behalf, this property is set to a value that is identical to the value of the **PidTagReceivedBySearchKey** property.

2.2.1.28 PidTagReadReceiptRequested Property

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

The **PidTagReadReceiptRequested** property ([\[MS-OXPROPS\]](#) section 2.957) specifies whether the e-mail sender requests a read receipt from all recipients (1) when this e-mail message is read or opened. If this property is absent, no read receipt is sent to the e-mail's sender. If the property is present, it is set to either 0x01, in which case the e-mail message's sender requests the read receipt from an e-mail client or server, or 0x00, in which case no read receipt is requested by the e-mail message's sender.

If an E-mail object that has its **PidTagReadReceiptRequested** property set to 0x01 is deleted, or it expires due to the time limit set by the **PidTagExpiryTime** property (section [2.2.3.7](#)) before the read receipt for this e-mail is generated, a non-read receipt is generated and sent to the e-mail message's sender or designated receipt recipient (1).

2.2.1.29 PidTagReceivedByAddressType Property

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

The **PidTagReceivedByAddressType** property ([\[MS-OXPROPS\]](#) section 2.959) contains the e-mail message receiver's e-mail address type, as specified by the **AddressType** field of the **RecipientRow** structure ([\[MS-OXCDATA\]](#) section 2.8.3.2).

2.2.1.30 PidTagReceivedByEmailAddress Property

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

The **PidTagReceivedByEmailAddress** property ([\[MS-OXPROPS\]](#) section 2.960) contains the e-mail message receiver's e-mail address, as specified by the **EmailAddress** field of the **RecipientRow** structure ([\[MS-OXCDATA\]](#) section 2.8.3.2).

2.2.1.31 PidTagReceivedByEntryId Property

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

The **PidTagReceivedByEntryId** property ([\[MS-OXPROPS\]](#) section 2.961) identifies an address book EntryID that contains the e-mail message receiver of the E-mail object. The address book EntryID data format is specified by the **EntryID** field of the **RecipientRow** structure ([\[MS-OXCDATA\]](#) section 2.8.3.2).

2.2.1.32 PidTagReceivedByName Property

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

The **PidTagReceivedByName** property ([\[MS-OXPROPS\]](#) section 2.963) contains the e-mail message receiver's display name, as specified by the **DisplayName** field of the **RecipientRow** structure ([\[MS-OXCDATA\]](#) section 2.8.3.2).

2.2.1.33 PidTagReceivedBySearchKey Property

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

The **PidTagReceivedBySearchKey** property ([\[MS-OXPROPS\]](#) section 2.964) identifies an address book search key that contains a binary-comparable key that is used to identify correlated objects for a search. This property is computed and set by concatenating the message receiver's **AddressType** and **EmailAddress** with a colon in between (for example, <TYPE>:<E-MAIL ADDRESS>), as specified by the **SearchKey** field of the **RecipientRow** structure ([\[MS-OXCDATA\]](#) section 2.8.3.2).

2.2.1.34 PidTagRecipientReassignmentProhibited Property

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

The **PidTagRecipientReassignmentProhibited** property ([\[MS-OXPROPS\]](#) section 2.978) specifies whether adding additional or different recipients (1), when forwarding the message, is prohibited for the e-mail message. This property is set based on the value of the e-mail message's **PidTagSensitivity** property ([\[MS-OXCMSG\]](#) section 2.2.1.13). If the **PidTagSensitivity** property is set to 0x00000000 (normal) or 0x00000003 (confidential), this property is set to 0x00 or is absent, meaning that adding additional or different recipients (1) to the e-mail message is allowed. If the **PidTagSensitivity** property of the E-mail object is set to 0x00000001 (personal) or 0x00000002 (private), this property is set to 0x01 to prevent adding additional or different recipients (1) of this e-mail message through forwarding.

2.2.1.35 PidTagReplyRecipientEntries Property

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

The **PidTagReplyRecipientEntries** property ([\[MS-OXPROPS\]](#) section 2.988) identifies a **FlatEntryList** structured array of address book EntryIDs for recipients (2) that will receive a reply. When the **PidTagReplyRecipientEntries** property and the **PidTagReplyRecipientNames** property (section [2.2.1.36](#)) are defined, the reply is sent to all the recipients (2) identified by these two properties. If this property is absent, a reply is sent only to the user identified by the **PidTagSenderEntryId** property (section [2.2.1.42](#)). If present, the property is set to a **FlatEntryList** structure of recipient (2) EntryIDs, as specified in [\[MS-OXCDATA\]](#) section 2.3.3.

The **PidTagReplyRecipientEntries** property and the **PidTagReplyRecipientNames** property MUST be set in a way that they contain the same number of recipients (2) in the same order.

2.2.1.36 PidTagReplyRecipientNames Property

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

The **PidTagReplyRecipientNames** property ([\[MS-OXPROPS\]](#) section 2.989) contains a list of display names for recipients (1) that are to get a reply. If this property is absent, a reply is sent only to the user identified by the **PidTagSenderName** property (section [2.2.1.43](#)). If present, the property is set to one string containing the address book entry's recipient (2) display names separated by semicolons.

2.2.1.37 PidTagReplyRequested Property

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

The **PidTagReplyRequested** property ([\[MS-OXPROPS\]](#) section 2.990) specifies whether a reply to the e-mail message is requested by the e-mail message's sender. If this property is absent, the reply to the e-mail message is not requested. If the property is present, it is set to either 0x01 if an e-mail sender requests a reply to the e-mail from recipients (1) or 0x00, which is the same handling as if the property is absent.

2.2.1.38 PidTagResponseRequested Property

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

The **PidTagResponseRequested** property ([\[MS-OXPROPS\]](#) section 2.1001) specifies whether an e-mail sender requests a response to a meeting request, as specified in [\[MS-OXOCAL\]](#) section 2.2.1.36, or requests a voting response (section [2.2.1.66](#)). If present, this property is set to either 0x01, in which case the response to the e-mail message is requested, or 0x00, in which case the response to the e-mail message is not requested. The default is 0x00.

2.2.1.39 PidTagSendRichInfo Property

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

The **PidTagSendRichInfo** property ([\[MS-OXOABK\]](#) section 2.2.3.18) specifies whether the sender can receive all message content, including **Rich Text Format (RTF)** and **Object Linking and Embedding (OLE)** objects. If this property is present, this property is set to either 0x01, indicating that the sender can receive all message contents, or 0x00, indicating that the sender of the e-mail message is using a different type of e-mail client. The default is 0x00.

2.2.1.40 PidTagSenderAddressType Property

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

The **PidTagSenderAddressType** property ([\[MS-OXPROPS\]](#) section 2.1073) contains the sending mailbox owner's e-mail address type, as specified by the **AddressType** field of the **RecipientRow** structure ([\[MS-OXCDATA\]](#) section 2.8.3.2 and section [2.2.4.3](#)).

2.2.1.41 PidTagSenderEmailAddress Property

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

The **PidTagSenderEmailAddress** property ([\[MS-OXPROPS\]](#) section 2.1074) contains the sending mailbox owner's e-mail address, as specified by the **EmailAddress** field of the **RecipientRow** structure ([\[MS-OXCDATA\]](#) section 2.8.3.2).

2.2.1.42 PidTagSenderEntryId Property

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

The **PidTagSenderEntryId** property ([\[MS-OXPROPS\]](#) section 2.1075) identifies an address book EntryID that contains the sending mailbox owner's address book EntryID, as specified by the address book EntryID ([\[MS-OXCDATA\]](#) section 2.2.5.2).

2.2.1.43 PidTagSenderName Property

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

The **PidTagSenderName** property ([\[MS-OXPROPS\]](#) section 2.1078) contains the sending mailbox owner's display name, as specified by the **DisplayName** field of the **RecipientRow** structure ([\[MS-OXCDATA\]](#) section 2.8.3.2).

2.2.1.44 PidTagSenderSearchKey Property

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

The **PidTagSenderSearchKey** property ([\[MS-OXPROPS\]](#) section 2.1079) identifies an address book search key that contains a binary-comparable key computed by concatenating the value of the sending mailbox owner's **PidTagAddressType** property ([\[MS-OXOABK\]](#) section 2.2.3.13) and **PidTagEmailAddress** property ([\[MS-OXOABK\]](#) section 2.2.3.14) with a colon in between (for example, <TYPE>:<E_MAIL ADDRESS>), as specified by the **SearchKey** field of the **RecipientRow** structure ([\[MS-OXCDATA\]](#) section 2.8.3.2).

2.2.1.45 PidTagSenderSmtpAddress Property

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

The **PidTagSenderSmtpAddress** property ([\[MS-OXPROPS\]](#) section 2.1080) contains the **Simple Mail Transfer Protocol (SMTP)** e-mail address format of the e-mail address of the sending mailbox owner.

2.2.1.46 PidTagSentRepresentingAddressType Property

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

The **PidTagSentRepresentingAddressType** property ([\[MS-OXPROPS\]](#) section 2.1086) contains an e-mail address type (section [2.2.4.3](#)) for the end user represented by the sending mailbox owner. If the sending mailbox owner is sending on his or her own behalf, this property MUST be set to the value of the **PidTagSenderAddressType** property (section [2.2.1.40](#)).

2.2.1.47 PidTagSentRepresentingEmailAddress Property

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

The **PidTagSentRepresentingEmailAddress** property ([\[MS-OXPROPS\]](#) section 2.1087) contains an e-mail address, as specified by the **EmailAddress** field of the **RecipientRow** structure ([\[MS-OXCDATA\]](#) section 2.8.3.2), for the end user who is represented by the sending mailbox owner. If a sending mailbox owner is sending on his or her own behalf, this property is set to the value of the **PidTagSenderEmailAddress** property (section [2.2.1.41](#)).

2.2.1.48 PidTagSentRepresentingEntryId Property

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

The **PidTagSentRepresentingEntryId** property ([\[MS-OXPROPS\]](#) section 2.1088) identifies an address book EntryID, as specified by the address book EntryID ([\[MS-OXCDATA\]](#) section 2.2.5.2), that contains the identifier of the end user who is represented by the sending mailbox owner. If the sending mailbox owner is sending on his or her own behalf, this property is set to the value of the **PidTagSenderEntryId** property (section [2.2.1.42](#)).

2.2.1.49 PidTagSentRepresentingName Property

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

The **PidTagSentRepresentingName** property ([\[MS-OXPROPS\]](#) section 2.1090) contains the display name for the end user who is represented by the sending mailbox owner. If a sending mailbox owner is sending on his or her own behalf, this property MUST be set to the value of the **PidTagSenderName** property (section [2.2.1.43](#)).

2.2.1.50 PidTagSentRepresentingSearchKey Property

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

The **PidTagSentRepresentingSearchKey** property ([\[MS-OXPROPS\]](#) section 2.1091) identifies an address book search key, as specified by the **SearchKey** field of the **RecipientRow** structure ([\[MS-OXCDATA\]](#) section 2.8.3.2), that contains a binary-comparable key that represents the end user who is represented by the sending mailbox owner. If a sending mailbox owner sends on his or her own behalf, this property is set to the value of the **PidTagSenderSearchKey** property (section [2.2.1.44](#)).

2.2.1.51 PidTagSubjectPrefix Property

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

On an E-mail object, the **PidTagSubjectPrefix** property ([\[MS-OXCMSG\]](#) section 2.2.1.9) represents an action on the e-mail message, such as "RE: " for replying and "FW: " for forwarding. If this property is absent, there is no subject prefix for the e-mail message.

On report messages, the value of the **PidTagSubjectPrefix** property is set as follows for the specified types of reports and responses:

- Delivery receipts: "Delivered: "
- Read receipts: "Read: "
- Sender response on read receipt requests: "Approved: "
- Non-delivery reports: "Undeliverable: "
- Non-read receipts: "Not Read : " or "read: "

2.2.1.52 PidTagTransportMessageHeaders Property

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

The **PidTagTransportMessageHeaders** property ([\[MS-OXPROPS\]](#) section 2.1124) contains transport-specific message envelope information for e-mail, as specified in [\[RFC2821\]](#). For outgoing messages with recipients (1) who have an SMTP address type, and for incoming messages from a sender who has an SMTP address type, the client and server respectively MUST set this property to a copy of the beginning of the message stream as received from SMTP, up to the first blank line (double CRLF, as specified in [\[RFC5234\]](#)).

2.2.1.53 PidLidInternetAccountName Property

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

The **PidLidInternetAccountName** property ([\[MS-OXPROPS\]](#) section 2.152) specifies the user-visible e-mail account name through which the e-mail message is sent. The format of this string is implementation-dependent. This property can be used by the client to determine which server to direct the mail to but is optional and the value has no meaning to the server.

2.2.1.54 PidLidInternetAccountStamp Property

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

The **PidLidInternetAccountStamp** property ([\[MS-OXPROPS\]](#) section 2.153) specifies the e-mail account ID through which the e-mail message is sent. The format of this string is implementation-dependent. This property can be used by the client to determine which server to direct the mail to but is optional and the value has no meaning to the server.

2.2.1.55 PidTagPrimarySendAccount Property

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

The **PidTagPrimarySendAccount** property ([\[MS-OXPROPS\]](#) section 2.940) specifies the first server to be used by a client to send the mail with. The format of this property is implementation-dependent. This property can be used by the client to determine which server to direct the mail to but is optional and the value has no meaning to the server.

2.2.1.56 PidTagNextSendAcct Property

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

The **PidTagNextSendAcct** property ([\[MS-OXPROPS\]](#) section 2.882) specifies the server that a client is currently attempting to use to send mail. The format of this property is implementation-dependent. This property can be used by the client to determine which server to direct the mail to but is optional and the value has no meaning to the server.

2.2.1.57 PidLidUseTnef Property

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

The **PidLidUseTnef** property ([\[MS-OXPROPS\]](#) section 2.349) is set to **TRUE** if **Transport Neutral Encapsulation Format (TNEF)** is included on a message when the message is converted from TNEF to MIME or SMTP format. Otherwise, this property is set to **FALSE**. If this property is absent, implementers of this protocol MUST NOT include TNEF on the message.

2.2.1.58 Attachments

The client can use attachment properties as specified in [\[MS-OXCMSG\]](#) section 2.2.2.

2.2.1.59 Categories and Keywords

The client can set categories or keywords on an e-mail message as specified in [\[MS-OXCMMSG\]](#) section 2.2.1.17.

2.2.1.60 Contacts

The client can set the contacts on an e-mail message as specified in [\[MS-OXOCNTC\]](#) and [\[MS-OXCMMSG\]](#) section 2.2.1.49.2.

2.2.1.61 Flags

The client can set **flags** as specified in [\[MS-OXOFLAG\]](#).

2.2.1.62 Reminders

The client can set **reminders** as specified in [\[MS-OXORMDR\]](#).

2.2.1.63 Recipients

The client adds recipients (2) to an e-mail message by using the **RopModifyRecipients** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.5), as specified in [\[MS-OXCMMSG\]](#) section 2.2.3.5. The value of the **PidTagRecipientType** property (section 2.2.3.1) is set to 0x00000001 for the primary recipients, 0x00000002 for Cc recipients, or 0x00000003 for Bcc recipients. For more details about the **RecipientRow** structure, see [\[MS-OXCADATA\]](#) section 2.8.3.2.

2.2.1.64 PidLidAutoProcessState Property

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

The **PidLidAutoProcessState** property ([\[MS-OXPROPS\]](#) section 2.40) specifies the options used in automatic processing of e-mail messages. The property can be absent, in which case the default value of 0x00000000 is used. If set, this property is set to one of the values in the following table.

Value	Meaning
0x00000000	Do not automatically process the message.
0x00000001	Process the message automatically or when the message is opened.
0x00000002	Process the message only when the message is opened.

2.2.1.65 PidLidVerbStream Property

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

The **PidLidVerbStream** property ([\[MS-OXPROPS\]](#) section 2.352) specifies what voting responses the user can make in response to the message. The format of this property is shown in the following diagram.

0	1	2	3	4	5	6	7	8	9	1	0	1	2	3	4	5	6	7	8	9	2	0	1	2	3	4	5	6	7	8	9	3	0	1
Version										Count																								
...										VoteOption1 (variable)																								
...																																		
VoteOptionN (variable)																																		
...																																		
Version2										VoteOptionsExtras1 (variable)																								
...																																		
VoteOptionsExtras2 (variable)																																		
...																																		

Version (WORD): Set to 0x0102.

Count (DWORD): Specifies the number of **VoteOption** and **VoteOptionExtras** structures to follow.

VoteOption1 (variable length): The first **VoteOption** structure specified in section [2.2.1.65.1](#).

VoteOptionN (variable length): The last **VoteOption** structure specified in section [2.2.1.65.1](#).

Version2 (WORD): MUST be set to 0x0104.

VoteOptionExtras1 (variable length): The first **VoteOptionExtras** structure specified in section [2.2.1.65.2](#).

VoteOptionExtrasN (variable length): The last **VoteOptionExtras** structure specified in section [2.2.1.65.2](#).

2.2.1.65.1 VoteOption Structure

The verb stream contains two parallel arrays of **VoteOption** and **VoteOptionExtra** structures. Each element in these two arrays, when combined, describes a single voting option that can be taken by the user in response to the message. The format of the **VoteOption** structure is shown in the following diagram.

0	1	2	3	4	5	6	7	8	9	1	0	1	2	3	4	5	6	7	8	9	2	0	1	2	3	4	5	6	7	8	9	3	0	1
VerbType																																		
DisplayNameCount										DisplayName (variable)																								
...																																		
MsgClsNameCount										MsgClsName (variable)																								
...																																		
Internal1StringCount										DisplayNameCount Repeat										DisplayNameRepeat (variable)														
...																																		
Internal2																																		
Internal3										fUseUSHeaders																								
...										Internal4																								
...										SendBehavior																								
...										Internal5																								
...										ID																								
...										Internal6																								
...																																		

VerbType (DWORD): The verb used by this structure. Set to 4 (0x00000004).

DisplayNameCount (1 byte): The count of characters in the **DisplayName** field.

DisplayName (variable): The localized display name of the voting option (for example, "Yes") as an ANSI string, without the null terminating character.

MsgClsNameCount (1 byte): The count of characters in the **MsgClsName** field. Set to 8 (0x08).

MsgClsName (variable): Set to "IPM.Note", without the null terminating character.

Internal1StringCount (1 byte): The count of characters in the following string. Set to 0x00 for voting options.

Internal1String (variable): MUST NOT be present, as **Internal1StringCount** is always 0x00 for a voting option.

DisplayNameCountRepeat (1 byte): MUST have the same value as the **DisplayNameCount** field.

DisplayNameRepeat (variable): MUST have the same value as the **DisplayName** field.

Internal2 (DWORD): Set to 0x00000000.

Internal3 (1 byte): Set to 0x00.

fUseUSHeaders (DWORD): Indicates that a U.S. style reply header is to be used in the response message (as opposed to a localized response header). The value is set to either 0x00000001, using U.S. style reply header, or 0x00000000 otherwise.

Internal4 (DWORD): Set to 0x00000001.

SendBehavior (DWORD): Indicates the behavior on send. When a user chooses a voting option, **SendBehavior** specifies whether the user is to be prompted to edit the response mail or whether the client automatically sends it on behalf of the user. The value of this field is one of the values defined in the following table.

Value	Meaning
0x00000001	Automatically send the voting response message.
0x00000002	Prompt the user to specify whether he or she would like to automatically send or edit the voting response first.

Internal5 (DWORD): Set to 0x00000002.

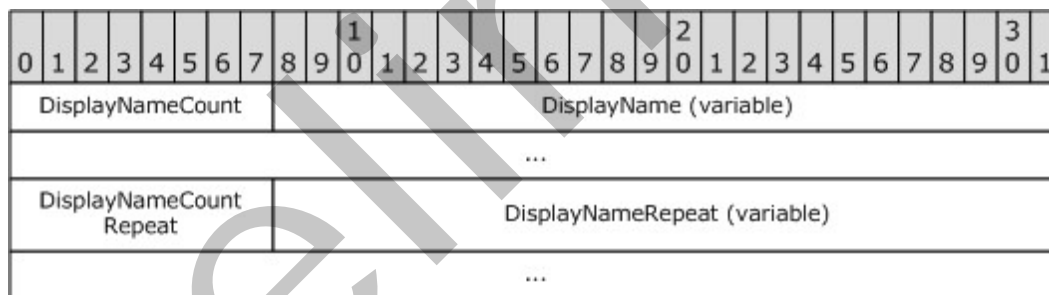
ID (DWORD): Specifies a numeric identifier for this voting option. The client SHOULD specify 1 for the first **VoteOption** structure and monotonically increase this value for each subsequent **VoteOption** structure.

Internal6 (DWORD): Set to "-1" (0xFFFFFFFF).

Note that because the **DisplayNameCount** field (and the **DisplayNameCountRepeat** field) is 1 byte long and contains the **COUNT** of characters in the **DisplayName** field (and the **DisplayNameRepeat** field), this implies a length limit of 255 characters in the **DisplayName** field of any voting option.

2.2.1.65.2 VoteOptionExtras Structure

Each element contains additional information about the corresponding **VoteOption** structure (section [2.2.1.65.1](#)). The format is shown in the following diagram.



DisplayNameCount (1 byte): The **COUNT** ([\[MS-OXCDATA\]](#) section 2.11.1) of **Unicode** characters (not bytes) in the **DisplayName** field.

DisplayName (: variable): The display name of this voting option, as a Unicode string without a null terminator.

DisplayNameCountRepeat (1 byte): The **COUNT** of characters in the **DisplayNameRepeat** field. MUST have the same value as the **DisplayNameCount** field.

DisplayNameRepeat [: Unicode **String (NOT null terminated)**]: A duplicate instance of the display name, as a Unicode string without a null terminator. MUST have the same value as the **DisplayName** field.

2.2.1.66 PidLidVerbResponse Property

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

The **PidLidVerbResponse** property ([\[MS-OXPROPS\]](#) section 2.351) specifies the voting option that a respondent has selected. Corresponds to one of the values of the **DisplayName** field in the **VoteOption** structure (section [2.2.1.65.1](#)). If present, this property MUST be set to the textual description of the user interface element selected by the user.

2.2.1.67 PidTagTargetEntryId Property

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

The **PidTagTargetEntryId** property ([\[MS-OXPROPS\]](#) section 2.1112) is used in conjunction with an optimizing send client. The semantics of an optimizing send are specified in section [3.2.4.4](#) and section [3.3.5.1.3](#).

2.2.1.68 PidTagAutoResponseSuppress Property

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

The **PidTagAutoResponseSuppress** property ([\[MS-OXPROPS\]](#) section 2.693) specifies whether a client or server application can forego sending automated replies in response to this message. Valid values are given in the following table.

Value	Meaning
-1	Suppress all automatic replies.
0	Do not suppress any automatic replies.
Values greater than 0	Suppress those replies indicated by the bits set on this value, as specified in the following table.

When the value of this property is greater than 0, it is interpreted as a bitwise OR of one or more of the following values.

Value	Meaning
0x00000001	Suppress delivery reports.
0x00000002	Suppress non-delivery reports.
0x00000004	Suppress read notifications from clients that receive the message.
0x00000008	Suppress non-read notifications from clients that receive the message.
0x00000010	Suppress Out of Office (OOF) messages.
0x00000020	Suppress all auto-reply messages other than OOF notifications.

2.2.1.69 PidTagMessageEditorFormat Property

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

The **PidTagMessageEditorFormat** property ([\[MS-OXPROPS\]](#) section 2.866) specifies the format that an e-mail editor can use for editing the message body (2). Valid values are listed in the following table.

Value name	Value	Meaning
EditorFormatDontKnow	0x00	The format for the editor to use is unknown.
EditorFormatPlainText	0x01	The optimal editing format is plain text.
EditorFormatHtml	0x02	The optimal editing format is HTML.
EditorFormatRtf	0x03	The optimal editing format is RTF.

2.2.1.70 PidTagMessageSubmissionId Property

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

The **PidTagMessageSubmissionId** property ([\[MS-OXPROPS\]](#) section 2.875) is a binary identifier assigned to a message by a **message transfer agent (MTA)**.

2.2.1.71 PidTagSenderIdStatus Property

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

The **PidTagSenderIdStatus** property ([\[MS-OXPROPS\]](#) section 2.1077) contains the results reported by the Sender ID agent, which compares the IP address of the message sender against the domain (called the purported responsible domain) of the sender's e-mail address. The value of this property indicates the likelihood that the current message is **spam**. Valid values are given in the following table.

Value	Meaning
0x00000001	Neutral. The verification check was inconclusive.
0x00000002	Pass. The IP address and the purported responsible domain match.
0x00000003	Fail. The IP address and the purported responsible domain do not match.
0x00000004	Soft fail. It is possible that the IP address does not belong to the purported responsible domain. A soft fail indicates less confidence in the message's authenticity than a value of Neutral (0x00000001).
0x00000005	None. No data could be obtained from the Domain Name System (DNS) .
0x00000006	Temporary error. There was a transient error (such as the unavailability of DNS) that prevented this value from being computed.
0x00000007	Permanent error. There was an unrecoverable error that prevented this value from being computed.

2.2.1.72 PidTagListHelp Property

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

The **PidTagListHelp** property ([\[MS-OXPROPS\]](#) section 2.846) contains a **Uniform Resource Identifier (URI)** that provides detailed help information for the mailing list from which this e-mail message was sent. This property corresponds to the **List-Help header (2)** in MIME, which is specified in [\[RFC2369\]](#). Conversion between this property and the **List-Help header (2)** is specified in [\[MS-OXCMAIL\]](#) section 2.1.3.2.15.

2.2.1.73 PidTagListSubscribe Property

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

The **PidTagListSubscribe** property ([\[MS-OXPROPS\]](#) section 2.847) contains the URI that subscribes a recipient (2) to the message's associated mailing list. This property corresponds to the **List-Subscribe header (2)** in MIME, which is specified in [\[RFC2369\]](#). Conversion between this property and the **List-Subscribe header (2)** is specified in [\[MS-OXCMAIL\]](#) section 2.1.3.2.15.

2.2.1.74 PidTagListUnsubscribe Property

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

The **PidTagListUnsubscribe** property ([\[MS-OXPROPS\]](#) section 2.848) contains the URI that unsubscribes a recipient (2) from the message's associated mailing list. This property corresponds to the **List-Unsubscribe header (2)** in MIME, which is specified in [\[RFC2369\]](#). Conversion between this property and the **List-Unsubscribe header (2)** is specified in [\[MS-OXCMAIL\]](#) section 2.1.3.2.15.

2.2.1.75 PidTagDelegatedByRule Property

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

The **PidTagDelegatedByRule** property ([\[MS-OXPROPS\]](#) section 2.746) specifies whether the message was forwarded due to the triggering of a delegate forward rule. This property is set to **TRUE** if the message was so forwarded; otherwise, it is set to **FALSE**. Delegate rules are specified in [\[MS-OXODLGT\]](#) section 2.2.3.

2.2.1.76 PidTagCurrentDraftEntryId

Note All of the information in this section is subject to change because it applies to a preliminary implementation of the protocol or structure.

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

The **PidTagCurrentDraftEntryId** property ([\[MS-OXPROPS\]](#) section 2.736) MAY<4> be used to store an EntryID for the message that corresponds to the draft reply to another message. The value of this property is equal to the value of the **PidTagEntryId** property ([\[MS-OXCPerm\]](#) section 2.2.1.4) for the most recent version of the draft reply to the message. This property is stored on the message to which the user is replying.

2.2.2 Message Status Reports Properties

2.2.2.1 PidTagMessageClass Property

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

The **PidTagMessageClass** property ([\[MS-OXCMMSG\]](#) section 2.2.1.3) contains a Message object class name. For report messages, the property is set to the value in the form "REPORT.X.<receipt types>", where X is the original **message class** name, such as "IPM.NOTE" for an E-mail object, and <receipt-type> is one of the following receipt types:

- **IPNRN**: Read receipt
- **IPNNRN**: Non-read receipt
- **DR**: Delivery receipt
- **NDR**: Non-delivery report

Therefore, the report messages of the IPM.NOTE message class name are as listed in the following table.

Report type	Message class name (PtypString)
Read receipt	REPORT.IPM.NOTE.IPNRN
Non-read receipt	REPORT.IPM.NOTE.IPNNRN
Delivery receipt	REPORT.IPM.NOTE.DR
Non-delivery report	REPORT.IPM.NOTE.NDR

2.2.2.2 PidTagOriginalDeliveryTime Property

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

The **PidTagOriginalDeliveryTime** property ([\[MS-OXPROPS\]](#) section 2.896) is set on read receipt/non-read receipt objects or replying/forwarding Message objects by using the value of the **PidTagMessageDeliveryTime** property (section [2.2.3.9](#)) from the original message.

2.2.2.3 PidTagOriginalDisplayTo Property

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

The **PidTagOriginalDisplayTo** property ([\[MS-OXPROPS\]](#) section 2.899) is set on report messages by using the value of the **PidTagDisplayTo** property (section [2.2.1.9](#)) from the original message, if present.

2.2.2.4 PidTagOriginalDisplayCc Property

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

The **PidTagOriginalDisplayCc** property ([\[MS-OXPROPS\]](#) section 2.898) is set on report messages by using the value of the **PidTagDisplayCc** property (section [2.2.1.8](#)) from the original message, if present.

2.2.2.5 PidTagOriginalDisplayBcc Property

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

The **PidTagOriginalDisplayBcc** property ([\[MS-OXPROPS\]](#) section 2.897) is set on report messages by using the value of the **PidTagDisplayBcc** property (section [2.2.1.7](#)) from the original message, if present.

2.2.2.6 PidTagOriginalSenderAddressType Property

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

The **PidTagOriginalSenderAddressType** property ([\[MS-OXPROPS\]](#) section 2.903) is set on delivery report messages by using the value of the original message sender's **PidTagSenderAddressType** property (section [2.2.1.40](#)), as specified by **AddressType** field of the **RecipientRow** structure ([\[MS-OXCDATA\]](#) section 2.8.3.2).

2.2.2.7 PidTagOriginalSenderEmailAddress Property

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

The **PidTagOriginalSenderEmailAddress** property ([\[MS-OXPROPS\]](#) section 2.904) is set on delivery report messages to the value of the original message sender's **PidTagSenderEmailAddress** property (section [2.2.1.41](#)).

2.2.2.8 PidTagOriginalSenderEntryId Property

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

The **PidTagOriginalSenderEntryId** property ([\[MS-OXPROPS\]](#) section 2.905) contains an address book EntryID that is set on delivery report messages to the value of the **PidTagSenderEntryId** property (section [2.2.1.42](#)) from the original e-mail message.

2.2.2.9 PidTagOriginalSenderName Property

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

The **PidTagOriginalSenderName** property ([\[MS-OXPROPS\]](#) section 2.906) is set on delivery report messages to the value of the original message sender's **PidTagSenderName** property (section [2.2.1.43](#)).

2.2.2.10 PidTagOriginalSenderSearchKey Property

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

The **PidTagOriginalSenderSearchKey** property ([\[MS-OXPROPS\]](#) section 2.907) contains an address book search key that is set on delivery report messages to the value of the **PidTagSenderSearchKey** property (section [2.2.1.44](#)) of the original e-mail message.

2.2.2.11 PidTagOriginalSentRepresentingAddressType Property

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

The **PidTagOriginalSentRepresentingAddressType** property ([\[MS-OXPROPS\]](#) section 2.909) contains the address type of the end user who is represented by the original e-mail message sender.

It is set to the value of the **PidTagSentRepresentingAddressType** property (section [2.2.1.46](#)) of the original e-mail message.

2.2.2.12 PidTagOriginalSentRepresentingEmailAddress Property

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

The **PidTagOriginalSentRepresentingEmailAddress** property ([\[MS-OXPROPS\]](#) section 2.910) contains the e-mail address of the end user who is represented by the original e-mail message sender. It is set to the value of the **PidTagSentRepresentingEmailAddress** property (section [2.2.1.47](#)) of the original e-mail message.

2.2.2.13 PidTagOriginalSentRepresentingEntryId Property

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

The **PidTagOriginalSentRepresentingEntryId** property ([\[MS-OXPROPS\]](#) section 2.911) identifies an address book EntryID that contains the entry identifier of the end user who is represented by the original message sender. It is set to the value of the **PidTagSentRepresentingEntryId** property (section [2.2.1.48](#)) of the original message.

2.2.2.14 PidTagOriginalSentRepresentingName Property

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

The **PidTagOriginalSentRepresentingName** property ([\[MS-OXPROPS\]](#) section 2.912) contains the display name of the end user who is represented by the original e-mail message sender; set to the value of the **PidTagSentRepresentingName** property (section [2.2.1.49](#)) of the original e-mail message.

2.2.2.15 PidTagOriginalSentRepresentingSearchKey Property

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

The **PidTagOriginalSentRepresentingSearchKey** property ([\[MS-OXPROPS\]](#) section 2.913) identifies an address book search key that contains the value of the **SearchKey** field of the **RecipientRow** structure ([\[MS-OXCDATA\]](#) section 2.8.3.2) for the end user who is represented by the original message sender. It is set to the value of the **PidTagSentRepresentingSearchKey** property (section [2.2.1.50](#)) of the original message.

2.2.2.16 PidTagOriginalSubject Property

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

The **PidTagOriginalSubject** property ([\[MS-OXPROPS\]](#) section 2.914) specifies the subject of the original message and is set to the concatenated values of the **PidTagSubjectPrefix** property (section [2.2.1.51](#)) and the **PidTagNormalizedSubject** property ([\[MS-OXCMSG\]](#) section 2.2.1.10) of the original message.

2.2.2.17 PidTagOriginalSubmitTime Property

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

The **PidTagOriginalSubmitTime** property ([\[MS-OXPROPS\]](#) section 2.915) specifies the original e-mail message's submission date and time and is set to the value of the **PidTagClientSubmitTime**

property (section [2.2.3.11](#)). The property is used in reports only, and once set, it MUST NOT be changed.

2.2.2.18 PidTagParentKey Property

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

The **PidTagParentKey** property ([\[MS-OXPROPS\]](#) section 2.931) contains the search key that is used to correlate the original message and the reports about the original message. The server sets the property on the report message to the value of the **PidTagSearchKey** property ([\[MS-OXCPRPT\]](#) section 2.2.1.9) of the original e-mail message.

2.2.2.19 PidTagReportEntryId Property

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

The **PidTagReportEntryId** property ([\[MS-OXPROPS\]](#) section 2.993) is an optional property that can appear on a report message. This property contains an address book EntryID, as specified in [\[MS-OXCDATA\]](#) section 2.2.5.2, that represents the application that generated the report message.

2.2.2.20 PidTagReportName Property

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

The **PidTagReportName** property ([\[MS-OXPROPS\]](#) section 2.995) is an optional property that can appear on a report message. This property contains the display name for the application that generated the report message.

2.2.2.21 PidTagReportSearchKey Property

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

The **PidTagReportSearchKey** property ([\[MS-OXPROPS\]](#) section 2.996) is an optional property that can appear on a report message. This property contains an address book search key, as specified in [\[MS-OXCDATA\]](#) section 2.8.3.2, representing the application that generated the report message.

2.2.2.22 PidTagReportTag Property

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

The **PidTagReportTag** property ([\[MS-OXPROPS\]](#) section 2.997) contains the data that is used to correlate the report and the original message. The property can be absent if the sender does not request a reply or response to the original e-mail message. If the original E-mail object has either the **PidTagResponseRequested** property (section [2.2.1.38](#)) set to 0x01 or the **PidTagReplyRequested** property (section [2.2.1.37](#)) set to 0x01, then the property is set on the original E-mail object by using the following format.

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	
Cookie																																
...																																

...	Version
...	StoreEntryIdSize
...	StoreEntryId (variable)
...	
FolderEntryIdSize	
FolderEntryId (variable)	
...	
MessageEntryIdSize	
MessageEntryId (variable)	
...	
SearchFolderEntryIdSize	
SearchFolderEntryId (variable)	
...	
MessageSearchKeySize	
MessageSearchKey (variable)	
...	
ANSITextSize	
ANSIText (variable)	
...	

Cookie (9 bytes,): A null-terminated string of nine characters used for validation; set to "PCDFEB09".

Version (4 bytes): SHOULD be set to the value of the **CurrentVersion** field (0x00010002), but a client or server application SHOULD also recognize the value of the **NoSearchFolderVersion** field (0x00010001).

StoreEntryIdSize (4 bytes): Size of the **StoreEntryId** field.

StoreEntryId (variable length of bytes): If the value of the **StoreEntryIdSize** field is 0x00000000, this field is omitted. If the value is not zero, this field is filled with the number of bytes specified by the **StoreEntryIdSize** field.

FolderEntryIdSize (4 bytes): Size of the **FolderEntryId** field.

FolderEntryId (variable): If the value of the **FolderEntryIdSize** field is 0x00000000, this field is omitted. If the value is not zero, the field is filled with the number of bytes specified by the **FolderEntryIdSize** field.

MessageEntryIdSize (4 bytes): Size of the **MessageEntryId** field.

MessageEntryId (variable): If the value of the **MessageEntryIdSize** field is 0x00000000, this field is omitted. If the value is not zero, the field is filled with the number of bytes specified by the **MessageEntryIdSize** field.

SearchFolderEntryIdSize (4 bytes): If the value of the **Version** field equals the value of the **CurrentVersion** field, then the value of this field is the real size of the **SearchFolderEntryId** field. Otherwise, this field is set to 0x00000000.

SearchFolderEntryId (variable): If the value of the **SearchFolderEntryIdSize** field is 0x00000000, this field is omitted. If the value is not zero, the field is filled with the number of bytes specified by the **SearchFolderEntryIdSize** field.

MessageSearchKeySize (4 bytes): Size of the **MessageSearchKey** field.

MessageSearchKey (variable): If the value of the **MessageSearchKeySize** field is 0x00000000, this field is omitted. If the value is not zero, the **MessageSearchKey** field is filled with the number of bytes specified by the **MessageSearchKeySize** field.

ANSITextSize (4 bytes): Number of characters in the **ANSI Text** field.

ANSIText (variable): The subject of the original message. If the value of the **ANSITextSize** field is 0x00000000, this field is omitted. If the value is not zero, the field is filled with the number of bytes specified by the **ANSITextSize** field.

2.2.2.23 PidTagReportText Property

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

The **PidTagReportText** property ([\[MS-OXPROPS\]](#) section 2.998) contains the optional text for a report message. If this property is present, the server sets it to the user-readable text of the report message.

2.2.2.24 PidTagReadReceiptAddressType Property

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

The **PidTagReadReceiptAddressType** property ([\[MS-OXPROPS\]](#) section 2.953) contains the address type of the end user to whom a read receipt is directed.

2.2.2.25 PidTagReadReceiptEmailAddress Property

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

The **PidTagReadReceiptEmailAddress** property ([\[MS-OXPROPS\]](#) section 2.954) contains the e-mail address of the user to whom a read receipt is directed.

2.2.2.26 PidTagReadReceiptEntryId Property

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

The **PidTagReadReceiptEntryId** property ([\[MS-OXPROPS\]](#) section 2.955) contains an address book EntryID, as specified in [\[MS-OXCDATA\]](#) section 2.2.5.2, that represents the user to whom a read receipt is directed. This property is only used and validated if the **PidTagReadReceiptRequested** property (section [2.2.1.28](#)) is set to 0x01. This property can be absent, in which case, the value of the **PidTagReportEntryId** property (section [2.2.2.19](#)) is used as an alternative value. If neither property is present, the value of the **PidTagSenderEntryId** property (section [2.2.1.42](#)) is used to identify the user who receives the read receipt.

2.2.2.27 PidTagReadReceiptName Property

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

The **PidTagReadReceiptName** property ([\[MS-OXPROPS\]](#) section 2.956) contains the display name for the end user to whom a read receipt is directed.

2.2.2.28 PidTagReadReceiptSearchKey Property

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

The **PidTagReadReceiptSearchKey** property ([\[MS-OXPROPS\]](#) section 2.958) contains an address book search key, as specified in [\[MS-OXCDATA\]](#) section 2.8.3.2, that represents the user to whom a read receipt is directed. This property is only used and validated if the **PidTagReadReceiptRequested** property (section [2.2.1.28](#)) is set to 0x01. The property can be absent, in which case the **PidTagReportSearchKey** property (section [2.2.2.21](#)) is used as an alternative. If neither property is present, the **PidTagSenderSearchKey** property (section [2.2.1.44](#)) is used to identify the user who receives the read receipt.

2.2.3 E-Mail Submission Properties

The following are properties of the recipients (2) identified in the **recipient table**. These properties are used to control server behavior during message submission.

2.2.3.1 PidTagRecipientType Property

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

The **PidTagRecipientType** property ([\[MS-OXPROPS\]](#) section 2.982) represents the recipient type of a recipient (2) on the message. This property is set on each recipient (2). Valid values for this property are as follows.

Value	Meaning
0x00000000	The recipient (2) is the message originator.
0x00000001	The recipient (2) is a primary recipient.
0x00000002	The recipient (2) is a Cc recipient.
0x00000003	The recipient (2) is a Bcc recipient.

Additionally, the flags in the following table can be combined with the values listed in the previous table.

Flag	Meaning
0x10000000	If a message failed to be delivered to some recipients (1), the client can mark the message as a resend message by setting the mfResend bit (0x00000080) in the PidTagMessageFlags property ([MS-OXCMSG] section 2.2.1.6). Combining this flag with the value of the PidTagRecipientType property indicates that the server MUST resend the message to the recipient (1).
0x80000000	On a resend message, the recipient (1) received the message successfully and does not need to receive it again. The server MUST NOT send the resend message to the recipient (1).

2.2.3.2 PidTagDeferredSendNumber Property

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

When sending a message is deferred, the **PidTagDeferredSendNumber** property ([MS-OXPROPS] section 2.743) SHOULD be set along with the **PidTagDeferredSendUnits** property (section 2.2.3.3) if the **PidTagDeferredSendTime** property (section 2.2.3.4) is absent. The value is set between 0x00000000 and 0x000003E7 (0 and 999).

The **PidTagDeferredSendNumber** property is used to compute the value of the **PidTagDeferredSendTime** property when the **PidTagDeferredSendTime** property is not present.

2.2.3.3 PidTagDeferredSendUnits Property

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

The **PidTagDeferredSendUnits** property ([MS-OXPROPS] section 2.745) specifies the unit of time by which the value of the **PidTagDeferredSendNumber** property (section 2.2.3.2) is multiplied. If set, the **PidTagDeferredSendUnits** property has one of the values listed in the following table.

Value	Meaning
0x00000000	Minutes; for example, 60 seconds.
0x00000001	Hours; for example, 60x60 seconds.
0x00000002	Day; for example, 24x60x60 seconds.
0x00000003	Week; for example, 7x24x60x60 seconds.

2.2.3.4 PidTagDeferredSendTime Property

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

The **PidTagDeferredSendTime** property ([MS-OXPROPS] section 2.744) can be present if a client would like to defer sending the message after a certain amount of time.

If the **PidTagDeferredSendUnits** property (section 2.2.3.3) and the **PidTagDeferredSendNumber** property (section 2.2.3.2) are present, the value of this property is recomputed by using the following formula and the original value is ignored. In this formula, **TimeOf(PidTagDeferredSendUnits)** converts the property into the appropriate multiplier based on its value, as specified for the **PidTagDeferredSendUnits** property.

$$\text{PidTagDeferredSendTime} = \text{PidTagClientSubmitTime} +$$

```
PidTagDeferredSendNumber *
TimeOf(PidTagDeferredSendUnits)
```

If the value of the **PidTagDeferredSendTime** property is earlier than the current time (in UTC), the message is sent immediately.

2.2.3.5 PidTagExpiryNumber Property

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

The **PidTagExpiryNumber** property ([\[MS-OXPROPS\]](#) section 2.766) is used with the **PidTagExpiryUnits** property (section [2.2.3.6](#)) to define the expiry send time. If this property is present, the value is set between 0x00000000 and 0x000003E7 (0 and 999).

2.2.3.6 PidTagExpiryUnits Property

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

The **PidTagExpiryUnits** property ([\[MS-OXPROPS\]](#) section 2.768) is used to describe the unit of time that the value of the **PidTagExpiryNumber** property (section [2.2.3.5](#)) multiplies. If set, the following are the valid values of this property.

Value	Meaning
0x00000000	Minutes; for example, 60 seconds.
0x00000001	Hours; for example, 60 × 60 seconds.
0x00000002	Days; for example, 24 × 60 × 60 seconds.
0x00000003	Weeks; for example, 7 × 24 × 60 × 60 seconds.

2.2.3.7 PidTagExpiryTime Property

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

The **PidTagExpiryTime** property ([\[MS-OXPROPS\]](#) section 2.767) can be present when a client requests to receive an expiry event if the message arrives late.

If the **PidTagExpiryNumber** property (section [2.2.3.5](#)) and the **PidTagExpiryUnits** property (section [2.2.3.6](#)) are both present, the value of this property is recomputed by the following formula; the original value is ignored.

```
PidTagExpiryTime = PidTagClientSubmitTime +
PidTagExpiryNumber *
TimeOf(PidTagExpiryUnits)
```

2.2.3.8 PidTagDeleteAfterSubmit Property

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

The **PidTagDeleteAfterSubmit** property ([\[MS-OXPROPS\]](#) section 2.748) indicates that the original message **MUST** be deleted after the message is sent. If the property is not present, the server uses the value 0x00.

The valid values for this property are specified in the following table.

Value	Meaning
0x00	Do not delete the original message after it is sent.
0x01	Delete the original message after it is sent.

2.2.3.9 PidTagMessageDeliveryTime Property

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

The server sets the value of the **PidTagMessageDeliveryTime** property ([\[MS-OXPROPS\]](#) section 2.865) to the current time (in UTC) when it receives a message.

2.2.3.10 PidTagSentMailSvrEID Property

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

The **PidTagSentMailSvrEID** property ([\[MS-OXPROPS\]](#) section 2.1085) represents the **Sent Items folder** for the message. This folder MUST NOT be a **search folder (2)**. The server requires write permission on the folder so that the sent e-mail message can be copied to the Sent Items folder.

If this property is present, a copy of the message is created in the specified folder after the message is sent.

2.2.3.11 PidTagClientSubmitTime Property

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

The server sets the value of the **PidTagClientSubmitTime** property ([\[MS-OXPROPS\]](#) section 2.713) to the current time (in UTC) when the e-mail message is submitted.

2.2.4 Message Delivery ROPs

2.2.4.1 RopSubmitMessage ROP

The **RopSubmitMessage** ROP request ([\[MS-OXCROPS\]](#) section 2.2.7.1) sends an E-mail object to its designated recipients (2).

The message is identified by the **handle** index, which is maintained by both the server and client for the Message object. The handle index is acquired by a previous call to the **RopOpenMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.1) or the **RopCreateMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.2).

The complete syntax of the ROP request and response buffers for this ROP is specified in [\[MS-OXCROPS\]](#). This section specifies the syntax and semantics of various fields that are not fully specified in [\[MS-OXCROPS\]](#).

2.2.4.1.1 RopSubmitMessage ROP Request Buffer

The following description defines a valid field for the **RopSubmitMessage** ROP request buffer ([\[MS-OXCROPS\]](#) section 2.2.7.1.1).

SubmitFlags (1 byte): An integer flag that indicates how the message is to be delivered. Possible values are listed in the following table.

Value name	Value	Meaning
None	0x00	None.
PreProcess	0x01	The message needs to be preprocessed by the server.
NeedsSpooler	0x02	The message is to be processed by a client spooler.

2.2.4.1.2 RopSubmitMessage ROP Response Buffer

This protocol adds no additional information to the fields for the **RopSubmitMessage** ROP response buffer ([\[MS-OXCROPS\]](#) section 2.2.7.1).

2.2.4.2 RopAbortSubmit ROP

The **RopAbortSubmit** ROP ([\[MS-OXCROPS\]](#) section 2.2.7.2) is sent before an E-mail object is actually processed by the server or a client **mail spooler** in an attempt to abort the submission.

If the operation succeeds, the message currently queued on the server will be removed from the server. Unless the message is submitted for sending again, the message will not be delivered to its recipients (1).

The message to be aborted is identified by the **FolderId** and **MessageId** fields in the request buffer. The **RopSubmitMessage** ROP MUST have been invoked on this message previously.

The complete syntax of the ROP request and response buffers for this ROP is specified in [MS-OXCROPS]. This section specifies the syntax and semantics of various fields that are not fully specified in [MS-OXCROPS].

2.2.4.2.1 RopAbortSubmit ROP Request Buffer

This protocol adds no additional information to the fields for the **RopAbortSubmit** ROP request buffer ([\[MS-OXCROPS\]](#) section 2.2.7.2.1).

2.2.4.2.2 RopAbortSubmit ROP Response Buffer

This protocol adds no additional information to the fields for the **RopAbortSubmit** ROP response buffer ([\[MS-OXCROPS\]](#) section 2.2.7.2.2).

2.2.4.3 RopGetAddressTypes ROP

The **RopGetAddressTypes** ROP ([\[MS-OXCROPS\]](#) section 2.2.7.3) retrieves the address types of recipients (2) that are supported by the server.

In the request, the **Server object** that is associated with the **InputHandleIndex** field in the Server object table is the **Logon object**. However, in this **ROP request**, the Server object is ignored by the server.

The complete syntax of the ROP request and response buffers for this ROP is specified in [MS-OXCROPS]. This section specifies the syntax and semantics of various fields that are not fully specified in [MS-OXCROPS].

2.2.4.3.1 RopGetAddressTypes ROP Request Buffer

This protocol adds no additional information to the fields for the **RopGetAddressTypes** ROP request buffer ([\[MS-OXCROPS\]](#) section 2.2.7.3.1).

2.2.4.3.2 RopGetAddressTypes ROP Response Buffer

The following descriptions define valid fields for the **RopGetAddressTypes** ROP response buffer ([\[MS-OXCROPS\]](#) section 2.2.7.3.2).

AddressTypeCount (2 bytes): The number of address types that are returned.

AddressTypeSize (2 bytes): The total length of the **AddressTypes** field.

AddressTypes (variable): An array of null-terminated **ASCII** strings, each of which represents an address type. Examples of address types are "EX", "MAPIPD", "SMTP", "MHS", "PROFS", and "X400". The server processes address types it recognizes and leaves other address types to transports outside of the scope of this protocol.

2.2.4.4 RopOptionsData ROP

The **RopOptionsData** ROP ([\[MS-OXCROPS\]](#) section 2.2.7.9) retrieves the options data that is associated with an address type of recipients (2) supported by the server. [<5>](#)

The complete syntax of the ROP request and response buffers for this ROP is specified in [\[MS-OXCROPS\]](#). This section specifies the syntax and semantics of various fields that are not fully specified in [\[MS-OXCROPS\]](#).

2.2.4.4.1 RopOptionsData ROP Request Buffer

The following descriptions define valid fields for the **RopOptionsData** ROP request buffer ([\[MS-OXCROPS\]](#) section 2.2.7.9.1).

AddressType (variable): A null-terminated ASCII string. This value specifies the address type for which to return options. For details about address types, see section [2.2.4.3.2](#).

WantWin32 (1 byte): An 8-bit **Boolean**. This value specifies whether the help file data to be returned is in a format suited for 32-bit machines. Valid values are listed in the following table.

Value	Meaning
0x00	Help data file is not required to be in a format suited for 32-bit machines.
Any nonzero value	Help data file returned is required to be in a format suited for 32-bit machines.

2.2.4.4.2 RopOptionsData ROP Response Buffer

The following descriptions define valid fields for the **RopOptionsData** ROP response buffer ([\[MS-OXCROPS\]](#) section 2.2.7.9.2).

Reserved (1 byte): Reserved. This value is set to 0x01.

OptionsInfoSize (2 bytes): An unsigned 16-bit integer. This value specifies the size of the **OptionsInfo** field.

OptionsInfo (variable): An array of bytes. This field contains the same number of bytes as specified in the **OptionsInfoSize** field. This array contains opaque data from the server. Clients SHOULD ignore this field. Servers SHOULD [return](#) this field as an empty array.

HelpFileSize (2 bytes): An unsigned 16-bit integer. This value specifies the size of the **HelpFile** field.

HelpFile (variable, optional): An array of bytes. This field contains the same number of bytes as specified in the **HelpFileSize** field. This array specifies the help that is associated with an address type. This field MAY be omitted from the response.

HelpFileName (variable, optional): A null-terminated multibyte string. This string is present if **HelpFileSize** is nonzero and is not present otherwise. This string specifies the name that is associated with the help for this address type.

2.2.5 Spooler and Transport ROPs

The following ROPs can be used by a client to control the receipt of mail that is not delivered directly to the server, or the sending of mail from an e-mail account that is not supported on the server.

2.2.5.1 RopSetSpooler ROP

The **RopSetSpooler** ROP ([\[MS-OXCROPS\]](#) section 2.2.7.4) signals to the server that the client will act as a mail spooler. Multiple clients can act as spoolers.

The complete syntax of the ROP request and response buffers for this ROP is specified in [\[MS-OXCROPS\]](#). This section specifies the syntax and semantics of various fields that are not fully specified in [\[MS-OXCROPS\]](#).

2.2.5.1.1 RopSetSpooler ROP Request Buffer

For the **RopSetSpooler** ROP request buffer ([\[MS-OXCROPS\]](#) section 2.2.7.4.1), the **InputHandleIndex** field represents a Logon object handle.

This protocol adds no additional information to the fields for the **RopSetSpooler** ROP request buffer.

2.2.5.1.2 RopSetSpooler ROP Response Buffer

This protocol adds no additional information to the fields for the **RopAbortSubmit** ROP response buffer ([\[MS-OXCROPS\]](#) section 2.2.7.4.2).

2.2.5.2 RopGetTransportFolder ROP

The **RopGetTransportFolder** ROP ([\[MS-OXCROPS\]](#) section 2.2.7.8) retrieves the folder ID (FID) ([\[MS-OXCADATA\]](#) section 2.2.1.1) of the transport folder. Outgoing messages can be stored in this folder before a **RopTransportSend** ROP request ([\[MS-OXCROPS\]](#) section 2.2.7.6) is issued.

The complete syntax of the ROP request and response buffers for this ROP is specified in [\[MS-OXCROPS\]](#). This section specifies the syntax and semantics of various fields that are not fully specified in [\[MS-OXCROPS\]](#).

2.2.5.2.1 RopGetTransportFolder ROP Request Buffer

For the **RopGetTransportFolder** ROP request buffer ([MS-OXCROPS] section 2.2.7.8.1), the **InputHandleIndex** field is a Logon object handle.

This protocol adds no additional information to the fields for the **RopGetTransportFolder** ROP request buffer.

2.2.5.2.2 RopGetTransportFolder ROP Response Buffer

The following description defines a valid field for the **RopGetTransportFolder** ROP response buffer ([MS-OXCROPS] section 2.2.7.8.2).

FolderID: Contains the FID ([MS-OXCADATA] section 2.2.1.1) of the transport folder.

2.2.5.3 RopSpoolerLockMessage ROP

The **RopSpoolerLockMessage** ROP ([MS-OXCROPS] section 2.2.7.5) locks the specified message for spooling.

The complete syntax of the ROP request and response buffers for this ROP is specified in [MS-OXCROPS]. This section specifies the syntax and semantics of various fields that are not fully specified in [MS-OXCROPS].

2.2.5.3.1 RopSpoolerLockMessage ROP Request Buffer

For the **RopSpoolerLockMessage** ROP request buffer ([MS-OXCROPS] section 2.2.7.5.1), the **InputHandleIndex** field represents a Logon object handle.

The following descriptions define valid fields for the **RopSpoolerLockMessage** ROP request buffer.

MessageId (8 bytes): An integer that specifies the message to be locked.

LockState (1 byte): An integer flag that specifies a status to set on the message. Valid values are listed in the following table.

Value name	Value	Meaning
IstLock	0x00	Mark the message as locked.
IstUnlock	0x01	Mark the message as unlocked.
IstFinished	0x02	Mark the message as ready for processing by the server.

2.2.5.3.2 RopSpoolerLockMessage ROP Response Buffer

This protocol adds no additional information to the fields for the **RopSpoolerLockMessage** ROP response buffer ([MS-OXCROPS] section 2.2.7.5.2).

2.2.5.4 RopTransportSend ROP

The **RopTransportSend** ROP ([MS-OXCROPS] section 2.2.7.6) requests that the server send an e-mail message to recipients (1). The message to be sent is identified by the **InputHandleIndex** field, which is maintained by both the server and the client.

The complete syntax of the ROP request and response buffers for this ROP is specified in [MS-OXCROPS]. This section specifies the syntax and semantics of various fields that are not fully specified in [MS-OXCROPS].

2.2.5.4.1 RopTransportSend ROP Request Buffer

For the **RopTransportSend** ROP request buffer ([MS-OXCROPS] section 2.2.7.6.1), the **InputHandleIndex** field represents a Logon object handle.

This protocol adds no additional information to the fields for the **RopTransportSend** ROP request buffer.

2.2.5.4.2 RopTransportSend ROP Response Buffer

The following descriptions define valid fields for the **RopTransportSend** ROP response buffer ([MS-OXCROPS] section 2.2.7.6.2).

NoPropertiesReturned (1 byte): A Boolean integer that specifies whether any properties are included in the response. Set to 0x00 if properties are returned; otherwise, set to 0x01.

PropertyValueCount (2 bytes): The number of properties in the following **PropertyValues** array. Only exists if the value of the **NoPropertiesReturned** field is 0x00.

PropertyValues (variable): An array of **TaggedPropertyValue** structures, as specified in [MS-OXCDATA] section 2.11.4. This field contains the properties set on the message by the server in the process of sending the message. This field exists only if the value of the **NoPropertiesReturned** field is 0x00. This field contains the number of tags specified by the **PropertyValueCount** field.

2.2.5.5 RopTransportNewMail ROP

The **RopTransportNewMail** ROP ([MS-OXCROPS] section 2.2.7.7.1) notifies the server that new mail has been delivered to the Message store.

The complete syntax of the ROP request and response buffers for this ROP is specified in [MS-OXCROPS]. This section specifies the syntax and semantics of various fields that are not fully specified in [MS-OXCROPS].

2.2.5.5.1 RopTransportNewMail ROP Request Buffer

For the **RopTransportNewMail** ROP request buffer ([MS-OXCROPS] section 2.2.7.7.1), the **InputHandleIndex** field represents a Logon object handle.

The following descriptions define valid fields for the **RopTransportNewMail** ROP request buffer.

MessageId (8 bytes): An integer that specifies the message ID (MID) ([MS-OXCDATA] section 2.2.1.2) of the new message.

FolderId (8 bytes): An integer that specifies the location of the new message.

MessageClass (variable): A zero-terminated **ANSI character set** string that specifies the value of the **PidTagMessageClass** property ([MS-OXCMSG] section 2.2.1.3) of the message.

MessageFlags (4 bytes): A flag field that specifies the value of the **PidTagMessageFlags** property ([MS-OXCMSG] section 2.2.1.6) of the message.

2.2.5.5.2 RopTransportNewMail ROP Response Buffer

This protocol adds no additional information to the fields for the **RopTransportNewMail** ROP response buffer ([\[MS-OXCROPS\]](#) section 2.2.7.7.2).

Preliminary

3 Protocol Details

3.1 Common 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 data model (ADM) data types are defined in this section:

Global

Mailbox

Message

Send State

3.1.1.1 Global

The following ADM element is 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** ADM object type. The following ADM object is maintained for each **Mailbox** ADM object type.

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

3.1.1.3 Per Message Object

A Message object is represented by the **MessageObject** ADM type. The following ADM objects are maintained for each **MessageObject** ADM object type.

Mailbox.MessageObject.Recipients: The intended recipients of the message.

Mailbox.MessageObject.Sender: The user who is sending the message.

Mailbox.MessageObject.Subject: The topic to which the message pertains.

Mailbox.MessageObject.Body: The content of the message.

Mailbox.MessageObject.Attachments: A list of one or more files that are included with the e-mail message.

Mailbox.MessageObject.VotingOptions: A list of possible responses to a question asked by the sender of an e-mail message.

3.1.1.4 Per Send State

A Message object's send state is represented by the **SendState** ADM type. The following abstract elements are maintained for each **SendState**:

SendState.State: The delivery status of the message. The following **SendState.State** values identify the current send state:

- **Saved**. A send note stored within an Inter-Personal Mail (IPM) folder within a **store** object.
- **Submitted**. A send note that is marked to be sent by the server.
- **Sent**. A send note that has been claimed by the messaging transport for delivery to another messaging user.
- **Received**. A receive note that has been placed in the default Receive folder by the server.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

None.

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 Client 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 ADM types are defined in this section:

Global

Mailbox

Message

3.2.1.1 Global

The following ADM element is 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** ADM object type. The following ADM object is maintained for each **Mailbox** ADM object type.

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

3.2.1.3 Per Message Object

A Message object is represented by the **MessageObject** ADM type.

3.2.2 Timers

None.

3.2.3 Initialization

A client can control how e-mail messages are sent to the mail transport by implementing its own mail spooler. To do so, the client sends the **RopSetSpooler** ROP request ([\[MS-OXCROPS\]](#) section 2.2.7.4) after logging on to the server by using the **RopLogon** ROP ([\[MS-OXCROPS\]](#) section 2.2.3.1). The client also needs to save the FID ([\[MS-OXCADATA\]](#) section 2.2.1.1) of the spooler queue folder retrieved from the **RopLogon** ROP request for later use.

3.2.4 Higher-Layer Triggered Events

3.2.4.1 Sending a Message

A client sends an e-mail message by sending a **RopSubmitMessage** ROP request ([\[MS-OXCROPS\]](#) section 2.2.7.1) to the server. The client can specify the submit flags for sending the message, as specified in section [2.2.4.1](#). The client can also set the sender information of the message to instruct the server on how to properly process the message.

3.2.4.1.1 Sending the Message as a Represented Sender

The represented sender properties SHOULD be set by the client to represent the sender the message is intended to be sent from.

3.2.4.1.2 Sending the Message as the Actual Sender

Actual sender properties MUST be set to represent the sending mailbox owner.

3.2.4.1.3 Sending the Message as the Sender Itself

When a user intends to represent himself or herself as the actual sender of a message, and if the represented sender properties are present, they MUST be set to the values that represent the user.

3.2.4.1.4 Sending the Message on Behalf of Another Person

If a user sends the message on behalf of another user, the represented sender properties MUST be set to the user that the actual sender intends to represent.

3.2.4.2 Deferring Message Send

A client can set the **PidTagDeferredSendTime** property (section [2.2.3.4](#)) to send a message at a later time.

If both the **PidTagDeferredSendNumber** property (section [2.2.3.2](#)) and the **PidTagDeferredSendUnits** property (section [2.2.3.3](#)) are present, the **PidTagDeferredSendTime** property SHOULD be computed from the values of the **PidTagDeferredSendNumber** and **PidTagDeferredSendUnits** properties.

3.2.4.3 Sending a Message with Expiry Time

A client can set the **PidTagExpiryTime** property (section [2.2.3.7](#)) to set an expiry time on a message.

If both the **PidTagExpiryNumber** property (section [2.2.3.5](#)) and the **PidTagExpiryUnits** property (section [2.2.3.6](#)) are present, the **PidTagExpiryTime** property SHOULD be computed from the values of the **PidTagExpiryNumber** and **PidTagExpiryUnits** properties.

3.2.4.4 Optimizing Send

When a messaging client sends a message in a client implementation of an optimization, the client can set the value of the **PidTagTargetEntryId** property (section [2.2.1.67](#)) to the value of the **PidTagEntryId** property ([\[MS-OXCPERM\]](#) section 2.2.1.4) of the message being submitted. If this is done, the client moves the sent message to its local Sent Items folder after submission. Eventually, when the client imports its local Sent Mail folder changes to server, on the server side, the server can make use of the **PidTagTargetEntryId** property to optimize the operation by moving a copy of the submitted Message object to the Sent Items folder instead of requiring the client to upload the Message object content again. For more details about the server operation, see section [3.3.5.1.3](#).

3.2.4.5 Resending a Message

If a message fails to be delivered to all recipients (1), a client can mark this message as resend by setting **mfResend** in the **PidTagMessageFlags** property ([\[MS-OXCMSG\]](#) section 2.2.1.6).

The server will attempt to redeliver this message only to the recipients (1) who did not get the message in the previous delivery attempt.

3.2.4.6 Soliciting Votes from Voters

Voting is composed of a specific set of properties on a message that is used to communicate voting options and responses to one another. An overview of the sequence of events is as follows:

1. A client (sender) sends a voting message to a variety of recipients (1) (voters). This message contains a well-formed **PidLidVerbStream** property (section [2.2.1.65](#)) but is otherwise identical to a nonvoting message.

2. The voters, upon receiving the message and displaying it to the user, detect the existence of the **PidLidVerbStream** property and use the property information to display an additional voting user interface to the user.
3. If and when a voter selects a voting option, a specifically crafted response mail is generated and addressed to the sender.
4. The sender, upon receiving response messages, aggregates them for display to the user.

It is important to note that at each point in this process, the messages that are sent are identical to nonvoting messages except for the presence of both the **PidLidVerbStream** property and the **PidLidVerbResponse** property (section [2.2.1.66](#)).

3.2.4.6.1 Sending a Voting Message

A client associates a series of voting options with a message by setting the **PidLidVerbStream** property (section [2.2.1.65](#)).

3.2.4.6.2 Interpreting a Voting Message

When a client receives a message, it MUST check the **PidLidVerbStream** property (section [2.2.1.65](#)). If the client encounters a **VoteOption** structure that does not have 0x00000004 set for the **VerbType** field, the client ignores the existence of that **VoteOption** structure. [<7>](#)

3.2.4.6.3 Crafting a Voting Response Message

A voting response message MUST contain all of the following:

- The **PidTagSubjectPrefix** property (section [2.2.1.51](#)) set to the display name of the voting option chosen by the user.
- The **PidLidVerbResponse** property (section [2.2.1.66](#)) set to the voting option chosen by the user.

Otherwise, the message MUST be formatted as a regular reply e-mail message addressed to the initial voting sender, respecting all user preferences that are applicable to such.

The client MUST honor the **SendBehavior** field of the **VoteOption** structure. If the **SendBehavior** field specifies SendPrompt, and if the user selects "Edit", the appropriate user interface (as determined by the implementation) is displayed to allow the user to edit the automatically generated response.

3.2.4.6.4 Aggregating Voting Responses

The exact method for aggregating and displaying voting responses is a client implementation detail. [<8>](#)

3.2.4.7 Sending Mail Through a Specific Server

To control the specific server that sends a message, a client sends the message by using the **RopSubmitMessage** ROP request ([\[MS-OXCROPS\]](#) section 2.2.7.1) with the **NeedsSpooler** flag (0x02) set. The message is then put into the spooler queue folder of the Message store on the server.

3.2.4.8 Processing E-mail Objects in the Spooler Queue

When the client finds an E-mail object in the spooler queue folder that the client can handle, <9> it takes control of the message by sending the **RopSpoolerLockMessage** ROP request ([MS-OXCROPS] section 2.2.7.5) with the **LockState** field set to **IstLock**. The client then performs any implementation-dependent processing. If the client determines that the message can be handled by a particular server, it sends the **RopGetTransportFolder** ROP request ([MS-OXCROPS] section 2.2.7.8) to retrieve the FID ([MS-OXCDATA] section 2.2.1.1) of a folder where temporary transport objects can be stored (clients can cache the returned FID and avoid having to send the request multiple times), creates the message to be sent to the folder, and then sends the **RopTransportSend** ROP request ([MS-OXCROPS] section 2.2.7.6) to have that server deliver the message. If the client handles delivering the mail itself, it sets the **R** flag of the **RecipientFlags** field, as specified in [MS-OXCDATA] section 2.8.3.1, of each recipient (2) in the recipient table that it successfully delivers mail to.

After completing the previous steps, the client sends a **RopSpoolerLockMessage** ROP request ([MS-OXCROPS] section 2.2.7.5) with the **LockState** field set to **IstFinished** if the message has been sent to all recipients (2) or to **IstUnlock** if some recipients (2) have not yet been sent the message. If some recipients (2) have yet to be processed, the client determines whether another server can deliver the e-mail message. If another server is found, the client attempts to resubmit the message to the remaining recipients (2). If no remaining transports can deliver the mail, the client SHOULD generate a non-delivery report or notify the user of the error.

3.2.4.9 Delivering Mail to the Server

When a message is delivered to an account on the server by the client, such as a message received from a POP3 server that is set to deliver the message into a folder on the server, it SHOULD send a **RopTransportNewMail** ROP request ([MS-OXCROPS] section 2.2.7.7) for each mail delivered to inform the server of the new mail so that the server can process new mail.

3.2.5 Message Processing Events and Sequencing Rules

3.2.5.1 Sending a RopSubmitMessage ROP Request

If a client calling the **RopSubmitMessage** ROP ([MS-OXCROPS] section 2.2.7.1) has set the **PidTagTargetEntryId** property (section 2.2.1.67) on the E-mail object, it SHOULD set the following properties at the same time.

Property	Value
PidTagEntryId ([MS-OXCPerm] section 2.2.1.4)	SHOULD contain the same value as the PidTagTargetEntryId property, if present.
PidTagMessageFlags ([MS-OXCMSG] section 2.2.1.6)	The mfUnsent and mfRead flags MUST be cleared.
PidTagInternetMessageId (section 2.2.1.12)	The value SHOULD be copied from the original message.

3.2.5.2 Sending a RopAbortSubmit ROP Request

The **RopSubmitMessage** ROP ([MS-OXCROPS] section 2.2.7.1) MUST have been invoked on a message before calling the **RopAbortSubmit** ROP ([MS-OXCROPS] section 2.2.7.2).

3.2.5.3 Sending a RopLockSpoolerMessage ROP Request

The client sends a **RopLockSpoolerMessage** ROP request ([MS-OXCROPS] section 2.2.7.5) after determining, through an implementation-dependent mechanism, that it can handle the message. <10>

After a client makes a successful request to mark the message as locked, it MUST subsequently make a request to mark the message as unlocked or finished.

3.2.5.4 Sending a RopTransportNewMail ROP Request

The client MUST call the **RopSetSpooler** ROP ([MS-OXCROPS] section 2.2.7.4) before calling the **RopTransportNewMail** ROP ([MS-OXCROPS] section 2.2.7.7).

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

3.3 Server Details

3.3.1 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 ADM data types are defined in this section.

Global

Mailbox

MessageObject

EmailObject

User

3.3.1.1 Global

The following ADM element is common to both client and server.

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

3.3.1.2 Per Mailbox

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

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

3.3.1.3 Per Message Object

Messaging objects are represented by the **MessageObject** ADM data type. The following ADM elements are maintained for each **MessageObject** ADM data type:

MessageObject.Mid: An identifier for a **Mailbox.MessageObject** ADM data type that is a Message object or **Attachment object**.

MessageObject.FolderId: An identifier for a **Mailbox.MessageObject** ADM data type that is a Folder object.

MessageObject.EmailObject: A **Mailbox.MessageObject** ADM data type that represents an e-mail message.

3.3.1.4 Per E-mail Object

E-mail objects are represented by the **EmailObject** ADM data type. The following ADM elements are maintained for each **MessageObject.EmailObject** ADM data type:

LockState: A state that identifies whether the **MessageObject.EmailObject** ADM data type has been locked by a user acting as a mail spooler. This state has the following possible values:

- **Locked.** The **MessageObject.EmailObject** ADM data type is locked by the **User** ADM data type, or by another **User** ADM data type.
- **Unlocked.** The current **MessageObject.EmailObject** ADM data type is not locked. The current **User** ADM data type is permitted to change its state to **Locked**.

3.3.1.5 User

A logged-in user is represented by the **User** ADM data type. The following ADM elements are maintained for each **User** ADM data type:

IsSpooler: A Boolean state that indicates whether the **User** ADM data type is acting as a mail spooler. Valid values for this state are:

- **True.** The **User** ADM data type is acting as a mail spooler.
- **False.** The **User** ADM data type is not acting as a mail spooler.

The default is **False**.

3.3.2 Timers

None.

3.3.3 Initialization

None.

3.3.4 Higher-Layer Triggered Events

None.

3.3.5 Message Processing Events and Sequencing Rules

3.3.5.1 Receiving a RobSubmitMessage ROP Request

When an e-mail message is submitted using the **RopSubmitMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.7.1), any pending changes on the message are saved to the server.

The following specific error codes apply to the **RopSubmitMessage** ROP.

Error code name	Value	Meaning
ecShutoffQuotaExceeded	0x000004DD	Indicates that the maximum storage shut-off quota has been exceeded.
ecQuotaExceeded	0x000004D9	Indicates that the storage quota is exceeded for the mailbox, but the user can still receive mail.
ecNotSupported	0x80040102	Indicates that the Server object that is associated with the InputHandleIndex field in the Server object table is not of type message, or that the current logon session is a public logon.
ecTooManyRecips	0x00000505	Indicates that the number of recipients (2) on the message exceeds the allowed limit. If this error occurs, none of the recipients (2) will receive this message.
ecAccessDenied	0x80070005	Indicates that the message is a folder associated information (FAI) message.
ecRequiresRefResolve	0x0000047E	Indicates that the attachments contain references to paths that are inaccessible to the server and need to be resolved.

The server performs the operations specified in sections [3.3.5.1.1](#) through [3.3.5.1.4](#) on receipt of the **RopSubmitMessage** ROP request.

3.3.5.1.1 Permission Check

There are restrictions on the e-mail messages that can be submitted with a **RopSubmitMessage** ROP request ([\[MS-OXCROPS\]](#) section 2.2.7.1). The server checks the submitted messages against the restrictions and returns the corresponding error code if any of the conditions listed in the following table are met.

Condition	Error code name	Value
FAI message is submitted.	ecAccessDenied	0x80070005
Embedded message is submitted.	ecNotSupported	0x80040102
Upper limit of recipients (2) is exceeded.	ecTooManyRecips	0x00000505
Mailbox is running out of quota.	ecQuotaExceeded	0x000004D9
No write permission on the message.	ecAccessDenied	0x80070005

Further, the server **MUST** check that the sender has sufficient permissions to send this message on behalf of the actual sender that the current sender intends to represent.

If the message is sent by another user or user agent, the represented sender properties are set to the user that the actual sender intends to display on the message.

3.3.5.1.2 Delivering Mail on a RopSubmitMessage or RopTransportSend Request

When a client sends either the **RopSubmitMessage** ROP request ([\[MS-OXCROPS\]](#) section 2.2.7.1) with the **NeedsSpooler** flag (0x02) not set or the **RopTransportSend** ROP request ([\[MS-OXCROPS\]](#) section 2.2.7.6), the server is to attempt to send the e-mail message to the intended recipients (2). For each recipient (2) in the recipient table that it can send the e-mail message to, it sets the **R** flag of the **RecipientFlags** field ([\[MS-OXCDATA\]](#) section 2.8.3.1).

When the **NeedsSpooler** flag is set, the server MUST place the message into the spooler queue folder.

3.3.5.1.3 Properties Read and/or Set Upon Submission

The following properties are checked and modified by the server on a message submitted with the **RopSubmitMessage** ROP request ([\[MS-OXCROPS\]](#) section 2.2.7.1).

- **PidTagSentMailSvrEID** (section [2.2.3.10](#)): If this property is present, the message is copied to the folder identified by this property after the message is sent out.
- **PidTagDeleteAfterSubmit** (section [2.2.3.8](#)): If this property is set to 0x01, the message is deleted after the message is sent.
- **PidTagClientSubmitTime** (section [2.2.3.11](#)): The **PidTagClientSubmitTime** property is set to the current time in UTC.
- **PidTagContentFilterSpamConfidenceLevel** ([\[MS-OXCSPAM\]](#) section 2.2.1.3): The server SHOULD set this property to 0xFFFFFFFF (-1). A client can use this value as part of junk e-mail or spam filtering.
- **PidTagMessageLocaleId** ([\[MS-OXCMSG\]](#) section 2.2.1.5): The server SHOULD set this property to the current user logon's **language code identifier (LCID)**.
- **PidTagMessageFlags** ([\[MS-OXCMSG\]](#) section 2.2.1.6): If the **mfResend** flag in this property is set, the message is considered a resend message and the server will try to redeliver the message only to those recipients (1) who failed to receive it previously.
- **PidTagRecipientType** (section [2.2.3.1](#)): If a message is a resend message, and if this property of a recipient (2) has the 0x80000000 bit set, the server ignores this recipient (2); if the **PidTagRecipientType** property of a recipient (2) has the 0x10000000 bit set, the server tries to redeliver the message to this recipient (2).
- **PidTagTargetEntryId** (section [2.2.1.67](#)): When working in optimizing send mode and sending a message, a client creates a copy of the message in a server folder and can set the value of the new message's **PidTagTargetEntryId** property equal to the value of the **PidTagEntryId** property ([\[MS-OXCPerm\]](#) section 2.2.1.4) on the original message. Upon the invocation of the **RopSubmitMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.7.1), the server creates a copy of the submitted message and sets the value of the **PidTagEntryId** property to the value obtained from the **PidTagTargetEntryId** property. If the client sets the **PidTagTargetEntryId** property, the client keeps a copy of the submitted message in the Sent Items folder after submission. Eventually, the client will import the move in its local Sent Mail folder to the server. The server will find the matching item because the value of the **PidTagEntryId** property already exists on the server. Instead of requiring the client to upload the message content again, the server completes the operation by moving the copy of the submitted message already persisted on the

server to the Sent Items folder (server side). More details are specified in [\[MS-OXCFXICS\]](#) section 3.3.4.3.3.2.1.1.

- **PidTagEntryId**: If the client has set the **PidTagTargetEntryId** property and not set the **PidTagEntryId** property, the server SHOULD generate a new ID value and assign it to the **PidTagEntryId** property.

3.3.5.1.3.1 Represented Sender Properties

If the user or user agent who is sending the e-mail message is the mailbox owner and the represented sender properties are currently not present, the following represented sender properties MUST be set to the mailbox owner:

- **PidTagSentRepresentingAddressType** (section [2.2.1.46](#))
- **PidTagSentRepresentingEmailAddress** (section [2.2.1.47](#))
- **PidTagSentRepresentingEntryId** (section [2.2.1.48](#))
- **PidTagSentRepresentingName** (section [2.2.1.49](#))
- **PidTagSentRepresentingSearchKey** (section [2.2.1.50](#))

3.3.5.1.3.2 Actual Sender Properties

If the e-mail message is sent on behalf of another user and the represented sender properties represent a public folder or a distribution list, the actual sender properties MUST NOT be set. Otherwise, the following actual sender properties MUST be set by using the values of the mailbox owner:

- **PidTagSenderAddressType** (section [2.2.1.40](#))
- **PidTagSenderEmailAddress** (section [2.2.1.41](#))
- **PidTagSenderEntryId** (section [2.2.1.42](#))
- **PidTagSenderName** (section [2.2.1.43](#))
- **PidTagSenderSearchKey** (section [2.2.1.44](#))

3.3.5.1.3.3 Deferred Properties

When an e-mail message arrives with the deferred send properties set, the server MUST honor the deferred send time.

For a message with both the **PidTagDeferredSendNumber** property (section [2.2.3.2](#)) and the **PidTagDeferredSendUnits** property (section [2.2.3.3](#)) present, the server will recompute the value of the **PidTagDeferredSendTime** property (section [2.2.3.4](#)) from the **PidTagDeferredSendNumber** and **PidTagDeferredSendUnits** properties during message submission.

3.3.5.1.3.4 Expiry Properties

When an e-mail message arrives with the expiry properties set, the server MUST honor the expiry time.

For a message with both the **PidTagExpiryNumber** property (section [2.2.3.5](#)) and the **PidTagExpiryUnits** property (section [2.2.3.6](#)) present, the server will recompute the value of the **PidTagExpiryTime** property from the **PidTagExpiryNumber** and **PidTagExpiryUnits** properties during message submission.

3.3.5.1.4 Rule Processing

When an e-mail message is submitted or delivered, it is subject to further processing by rules, as specified in [\[MS-OXORULE\]](#).

3.3.5.2 Receiving a RopAbortSubmit ROP Request

When an e-mail message is submitted and is still queued on the server pending delivery, the submission can be terminated by sending a **RopAbortSubmit** ROP request ([\[MS-OXCROPS\]](#) section 2.2.7.2).

If the **mfSubmitted** bit of a submitted message's **PidTagMessageFlags** property ([\[MS-OXCMSG\]](#) section 2.2.1.6) has not been set yet, sending the **RopAbortSubmit** ROP request indicates to the server that it SHOULD stop delivering the message by removing the message from the **spooler queue**. The **mfUnsent** bit of the message's **PidTagMessageFlags** property is set and the **mfSubmitted** bit of the message's **PidTagMessageFlags** property is cleared. Even if the message's **PidTagDeferredSendTime** property (section [2.2.3.4](#)) has been set, the client will not be notified that the message has been deferred.

The **RopAbortSubmit** ROP can fail at the server's discretion. When the **RopAbortSubmit** ROP fails, the message can still be sent.

When a message is locked using the **RopSpoolerLockMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.7.5), the server MUST deny **RopAbortSubmit** ROP requests, as well as other requests to lock or access the message.

The following error codes can be returned by this ROP.

Error code name	Value	Meaning
ecUnableToAbort	0x80040114	The operation cannot be aborted.
ecNotInQueue	0x80040601	The message is no longer in the spooler queue of the Message store.
ecNotSupported	0x80040102	The Server object associated with the InputHandleIndex field in the Server object table is not of type logon or the current logon session is a public logon.
ecNotFound	0x8004010F	The parent FID ([MS-OXCDATA] section 2.2.1.1) or MID ([MS-OXCDATA] section 2.2.1.2) is invalid.

3.3.5.3 Receiving a RopGetAddressTypes ROP Request

The following error codes can be returned by the **RopGetAddressTypes** ROP ([\[MS-OXCROPS\]](#) section 2.2.7.3).

Error code name	Value	Meaning
ecBufferTooSmall	0x0000047D	The response buffer is not large enough to hold the results.

Error code name	Value	Meaning
ecNullObject	0x000004B9	An object handle reference in the RPC buffer could not be resolved. <11>
ecNotSupported	0x80040102	The server does not support returning address types.

3.3.5.4 Receiving a RopSetSpooler ROP Request

When the **RopSetSpooler** ROP request ([MS-OXCROPS] section 2.2.7.4) is received, the server marks the user logon to indicate that this is a spooler logon.

3.3.5.5 Receiving a RopGetTransportFolder ROP Request

In response to a **RopGetTransportFolder** ROP request ([MS-OXCROPS] section 2.2.7.8), the server MUST return a FID ([MS-OXCADATA] section 2.2.1.1) that identifies a folder that the client can use to temporarily store messages to be sent.

The following error code can be returned in the response.

Error code name	Value	Meaning
ecNullObject	0x000004B9	The InputHandleIndex field is not valid. <12>

3.3.5.6 Receiving a RopSpoolerLockMessage ROP Request

On receipt of a **RopSpoolerLockMessage** ROP request ([MS-OXCROPS] section 2.2.7.5), a server MUST take the actions listed in the following table based on the value of the **LockState** field.

Value Name	Action
IstLock	Locks the message for the client that is sending the request. The request fails if the message is locked by some other client.
IstUnlock	Unlock the message.
IstFinished	Unlock the message and complete post-processing of sent mail. The server moves or deletes the message based on the presence of the PidTagSentMailSvrEID property (section 2.2.3.10) and the PidTagDeleteAfterSubmit property (section 2.2.3.8) on the message.

The following error codes can be returned in the response.

Error code name	Value	Meaning
ecNoSupport	0x80040102	The server does not support sent message processing, or the client is not the spooler.
ecNotInQueue	0x80040601	An attempt was made to lock an already locked message, or the message is not in the spooler queue. <13>

3.3.5.7 Receiving a RopTransportSend ROP Request

If there was a failure to submit the message, the **RopTransportSend** ROP ([\[MS-OXCROPS\]](#) section 2.2.7.6) does not fail; in this case, the server generates a non-delivery report to the message instead.

The following error codes can be returned in a **RopTransportSend ROP response**.

Error code name	Value	Meaning
ecNotMe	0x80040502	The server could not handle the message and the message was not sent. The client SHOULD try another server if one is available.

3.3.5.8 Receiving a RopTransportNewMail ROP Request

When a server receives a **RopTransportNewMail** ROP request ([\[MS-OXCROPS\]](#) section 2.2.7.7), it MUST notify all clients that are connected to the mailbox of the receipt of new mail by using the **RopNotify** ROP ([\[MS-OXCROPS\]](#) section 2.2.14.2) and a **NewMail** event, as specified in [\[MS-OXCNOTIF\]](#) section 2.2.1.4.1.1.

The following error code can be returned in a **RopTransportNewMail ROP response**.

Error code name	Value	Meaning
ecNoSupport	0x80040102	The server did not receive a valid corresponding call for the RopSetSpooler ROP ([MS-OXCROPS] section 2.2.7.4).

3.3.5.9 Receiving a RopOptionsData ROP Request

The following error code can be returned in a **RopOptionsData ROP response** ([\[MS-OXCROPS\]](#) section 2.2.7.9).

Error code name	Value	Meaning
ecNullObject	0x000004B9	The InputHandleIndex field does not refer to a valid Server object. <14>

3.3.6 Timer Events

None.

3.3.7 Other Local Events

None.

4 Protocol Examples

This section includes examples of Message object operations that use sequences of ROP requests and ROP responses that a client and a server might exchange. Note that the examples listed here only show the relevant portions of the specified ROPs; this is not the final byte sequence that gets transmitted over the wire. Also note that the data for multibyte fields appears in little-endian format, with the bytes in the field presented from least significant to most significant. Generally speaking, these ROP requests are packed with other ROP requests and then compressed and packed in one or more RPC calls, as described in [\[MS-OXCROPS\]](#). These examples assume that the client has already successfully logged on to the server and has the appropriate permissions to the Message objects that the operations are being performed on.

4.1 Submitting a Message

In this example, the client has created a new Message object in the mailbox and is ready to submit the Message object. The client previously set a few message properties to values that are not of interest to this example and are not documented here.

4.1.1 ROP Request Buffer

The ROP request buffer in this example resembles the following.

```
0000: 32 00 02 00
```

The composition of the bytes is as follows:

RopId: 0x32 (**RopSubmitMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.7.1))

LogonId: 0x00

InputHandleIndex: 0x02

SubmitFlags: 0x00 (None)

The first three bytes refer to the **RopId**, **LogonId**, and **InputHandleIndex** fields, which are the same for all ROPs described in [\[MS-OXCROPS\]](#). The value of the **SubmitFlags** field is None. The message identified by an **InputHandleIndex** value of 0x02 was submitted.

4.1.2 ROP Response Buffer

The ROP response buffer in this example resembles the following.

```
0000: 32 02 00 00 00 00
```

The composition of the response buffer is as follows:

RopId: 0x32 (**RopSubmitMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.7.1))

InputHandleIndex: 0x02

ReturnValue: 0x00000000 (ecNone)

The value of the response's **InputHandleIndex** field is the same as the value of the **InputHandleIndex** field of the **RopSubmitMessage** ROP, and the return value of 0x00000000

indicates success. From the response, the client is aware that the message was submitted successfully.

4.2 Submitting a Deferred Message

In this example of submitting a deferred message, the client has created a new Message object in the mailbox and wants to submit the Message object. The client sets properties related to a deferred send. The client also sets other message properties that are not described in section 4.2.1, but the properties are not relevant to this example and are not included.

4.2.1 ROP Request Buffer

The ROP request buffer in this example resembles the following.

```
0000: 0A 01 01 0E 00 01 00 40 00 EF 3F 96 3F 7F F4 5E
0010: 6F C8 01
...
00xx: 32 01 01 00
```

The composition of the bytes is as follows:

ROPId: 0x0A (**RopSetProperties** ROP ([\[MS-OXCROPS\]](#) section 2.2.8.6))

LogonId: 0x01

InputHandleIndex: 0x01

PropertyValueSize: 0x000E

PropertyValueCount: 0x0001

PropertyValues[0].PropertyTag: 0x3FEF0040 (**PidTagDeferredSendTime** property (section 2.2.3.4))

PropertyValues[0].PropertyValue: 0x01C86F5EF47F3F96 (UTC FILETIME: 11:11:39PM 02/14/2008)

...

ROPId: 0x32 (**RopSubmitMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.7.1))

LogonId: 0x01

InputHandleIndex: 0x01

SubmitFlags: 0x00 (None)

The value of the **PidTagDeferredSendTime** property of the message (identified by the value 0x01 in the **InputHandleIndex** field) was set to 11:11:39 P.M. 02/14/2008 (UTC). The client intends to defer the submission until 11:11:39 P.M. on 02/14/2008.

4.2.2 ROP Response Buffer

The ROP response buffer in this example resembles the following.

```
0000: 0A 01 00 00 00 00 00 00
```

```
...
0000: 32 01 00 00 00 00
```

The composition of the response buffer is as follows:

RopId: 0x0A (**RopSetProperties** ROP ([\[MS-OXCROPS\]](#) section 2.2.8.6))

InputHandleIndex: 0x01

ReturnValue: 0x00000000 (ecNone)

PropertyProblemCount: 0x0000

RopId: 0x32 (**RopSubmitMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.7.1))

InputHandleIndex: 0x01

ReturnValue: 0x00000000 (ecNone)

The response messages to both the **RopSetProperties** ROP and the **RopSubmitMessage** ROP indicate that the two ROPs succeeded.

If the **RopSubmitMessage** ROP is issued after UTC time 11:11:39 P.M. 02/14/2008, the message is submitted immediately. If the **RopSubmitMessage** ROP is issued before this time, the message is deferred for submission until the current time is equal to or is later than the deferred send time.

4.3 Aborting a Message Submission

In this example of aborting a message submission, a client has submitted a Message object. While the message is still queued on the server, the client would like to terminate the submission.

4.3.1 ROP Request Buffer

The ROP request buffer in this example resembles the following.

```
0000: 34 00 00 01 00 00 03 b4-79 ca 47 01 00 00 03 b7 4
0010: e6 5f a7
```

The composition of the request buffer is as follows:

RopId: 0x34 (**RopAbortSubmit** ROP ([\[MS-OXCROPS\]](#) section 2.2.7.2))

LogonId: 0x00

InputHandleIndex: 0x00

FolderId: 0001-0003b479ca47 (the FID ([\[MS-OXCADATA\]](#) section 2.2.1.1) of the parent folder)

MessageId: 0001-0003b7e65fa7 (the MID ([\[MS-OXCADATA\]](#) section 2.2.1.2) of the message submitted)

The message identified by the value 0x00 in the **InputHandleIndex** field was submitted previously. While the message is still queued in the server, the client sends the **RopAbortSubmit** ROP request related to this message to terminate the submission.

4.3.2 ROP Response Buffer

The ROP response buffer in this example would look like the following.

```
0000: 34 00 00 00 00 00 00
```

The composition of the response buffer is as follows:

ROPId: 0x34 (**RopAbortSubmit** ROP ([\[MS-OXCROPS\]](#) section 2.2.7.2))

InputHandleIndex: 0x00

ReturnValue: 0x00000000 (ecNone)

The response message indicates that the **RopAbortSubmit** ROP succeeded. The message has been removed from the server. The **mfUnsent** bit is set (restored) and the **mfSubmitted** bit is cleared on the message's **PidTagMessageFlags** property ([\[MS-OXCMSG\]](#) section 2.2.1.6). Unless another **RopSubmitMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.7.1) is issued on this Message object, the message will not be sent.

4.4 Sending an E-Mail Message from a Messaging User to Another Messaging User

Consider the following scenario: Joe Healy needs to send a high importance e-mail message to inform his customer, Ed Banti, that the order request form that Ed sent needs to be signed. Joe also wants to get a read receipt when Ed reads this e-mail message. The following is a description of what a client might do to accomplish Joe's intentions and the responses a server might return.

To create an E-mail object, the client uses the **RopCreateMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.2). The server returns a success code and a handle to a Message object. Joe types the e-mail subject and message text (plain text format), sets the e-mail message to high importance, and requests a read receipt. The client then uses the **RopSetProperties** ROP ([\[MS-OXCROPS\]](#) section 2.2.8.6) to transmit Joe's e-mail message data to the server. The values of each of the properties set by the **RopSetProperties** ROP are listed in the following table. The types in the table are described in [\[MS-OXCDATA\]](#) section 2.11.1.

Property	Property ID	Type	Value
PidTagBody ([MS-OXCMSG] section 2.2.1.48.1)	0x1000	0x001f (PtypString)	"Please sign the order request.\LF\CR"
PidTagMessageClass ([MS-OXCMSG] section 2.2.1.3)	0x001A	0x001F (PtypString)	"IPM.Note"
PidTagMessageFlags ([MS-OXCMSG] section 2.2.1.6)	0x0E07	0x0003 (PtypInteger32)	mfUnsent
PidTagConversationTopic (section 2.2.1.5)	0x0070	0x001f (PtypString)	"Order Request"
PidTagConversationIndex (section 2.2.1.3)	0x0071	0x0102 (PtypBinary)	22 bytes 01 c8 74 0b 0f 9c 35 2c 02 17 93 af 43 a9 8b b4 c1 bb ef 97 7d 4f
PidTagImportance ([MS-	0x0017	0x0003	0x00000002

Property	Property ID	Type	Value
OXCMSG] section 2.2.1.11)		(PtypInteger32)	High Importance
PidTagMessageDeliveryTime (section 2.2.3.9)	0x0E06	0x0040 (PtypTime)	2008/02/20 21:53:00.000
PidTagReadReceiptRequested (section 2.2.1.28)	0x0029	0x000B (PtypBoolean)	0x01 (TRUE)
PidTagSentMailSvrEID (section 2.2.3.10)	0x6740	0x00FB (PtypServerId)	21 bytes 01 01 00 00 00 00 f0 e7 c1 00 00 00 00 00 00 00 00 00 00 00 00
PidTagIconIndex (section 2.2.1.10)	0x1080	0x0003 (PtypInteger32)	0xFFFFFFFF
PidTagMessageEditorFormat ([MS-OXPROPS] section 2.866)	0x5909	0x0003 (PtypInteger32)	0x00000001 plain text
PidTagPrimarySendAccount (section 2.2.1.55)	0x0E28	0x001F (PtypString)	000000023659R9-A11/o=First Organization/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/ CN=recipients/CN=JoeHealy Microsoft Exchange
PidTagNextSendAcct (section 2.2.1.56)	0x0E29	0x001F (PtypString)	000000023659R9-A11/o=First Organization/ou=Exchange Administrative Group (FYDIBOHF23SPDLT)/ CN=recipients/CN=JoeHealy Microsoft Exchange
PidTagMessageLocaleId ([MS-OXCMSG] section 2.2.1.5)	0x3FF1	0x0003 (PtypInteger32)	1033 (en-us)
PidTagReportTag (section 2.2.2.22)	0x0031	0x0102 (PtypBinary)	100 bytes (See the explanation that follows this table.)

The value of the **PidTagReportTag** property is as follows:

```

0000: 50 43 44 46 45 42 30 39-00 01 00 02 00 00 00 00
0010: 00 00 00 00 00 00 00 00-00 00 00 2e 00 00 00 00
0020: 00 1a f8 62 55 f6 35 01-4f b0 20 ce 17 75 e8 64
0030: 0b 01 00 61 2a 7b ab 49-f6 4e 4b 9c 52 db fb 5a
0040: 53 aa 1c 00 00 00 f0 e7-c1 00 00 10 00 00 00 fd
0050: 02 6f a5 55 15 2a 41 ab-1f 64 5d 1b da 0c 38 01
0060: 00 00 00 00

```

Joe then addresses this e-mail message to Ed Banti as the primary recipient. The client locates Ed Banti's address data from the client's address book and adds Ed Banti's address data to the recipient table of this E-mail object by using the **RopModifyRecipients** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.5). The values of the **RecipientRow** elements are listed in the following table.

RecipientRow element	Value	Description
RowID	0x00000001	Row ID number
RecipientType	0x00000001	Primary recipient
DataSize	399	
RecipientFlag	0x0651	AddressType.EXCH DisplayName XmitSameAsDisplay StandardPropsUnicode SimpleDisplayName
DNPrefixLen	0x5A (90)	
EX-Address.Type	0x00000000	DT_MAILUSER
EX-Address.EmailAddress	edbanti@example.com	
DisplayName	Ed Banti	
SimpleDisplayName	Ed Banti	

The client adds the following additional properties to the **RecipientRow** structure.

Property	PropertyID	Type	Value
PidTagObjectType ([MS-OXCPRPT] section 2.2.1.7)	0x0FFE	0x0003 (PtypInteger32)	0x00000006 (MAILUSER)
PidTagDisplayType ([MS-EXOABK] section 2.2.3.11)	0x3900	0x0003 (PtypInteger32)	0x00000000 DT_MAILUSER
PidTagAddressBookDisplayNamePrintable ([MS-OXCMSG] section 2.2.1.30)	0x39FF	0x001F (PtypString)	Ed Banti
PidTagSmtptAddress ([MS-EXOABK] section 2.2.3.21)	0x39FE	0x001F (PtypString)	edbanti@example.com
PidTagSendInternetEncoding ([MS-EXOABK] section 2.2.3.19)	0x3A71	0x0003 (PtypInteger32)	0x00000000
PidTagAccount ([MS-OXOCNTC] section 2.2.1.10.11)	0x3A00	0x001F (PtypString)	edbanti
PidTagDisplayTypeEx ([MS-EXOABK] section 2.2.3.12)	0x3905	0x0003 (PtypInteger32)	0x40000000
PidTagRecipientTrackStatus ([MS-OXOCAL] section 2.2.4.10.2)	0x5FFF	0x0003 (PtypInteger32)	0x00000000
PidTagRecipientResourceState ([MS-	0x5FDE	0x0003	0x00000000

Property	PropertyID	Type	Value
[MS-OXOCAL] section 2.979)		(PtypInteger32)	
PidTagRecipientFlags ([MS-OXOCAL] section 2.2.4.10.1)	0x5FFD	0x0003 (PtypInteger32)	0x00000001
PidTagRecipientDisplayName ([MS-OXOCAL] section 2.971)	0x5FF6	0x001F (PtypString)	Ed Banti
PidTagRecipientEntryId ([MS-OXOCAL] section 2.972)	0x5FF7	0x0102 (PtypBinary)	126 bytes (see the sample value for the PidTagRecipientEntryId property following this table)
PidTagRecipientOrder ([MS-OXOCAL] section 2.2.1.40)	0x5FDF	0x0003 (PtypInteger32)	0x00000000

The value of the **PidTagRecipientEntryId** property is as follows:

```

0000: 00 00 00 00 dc a7 40 c8-c0 42 10 1a b4 b9 08 00
0010: 2b 2f e1 82 01 00 00 00-00 00 00 00 2f 6f 3d 46
0020: 69 72 73 74 20 4f 72 67-61 6e 69 7a 61 74 69 6f
0030: 6e 2f 6f 75 3d 45 78 63-68 61 6e 67 65 20 41 64
0040: 6d 69 6e 69 73 74 72 61-74 69 76 65 20 47 72 6f
0050: 75 70 20 28 46 59 44 49-42 4f 48 46 32 33 53 50
0060: 44 4c 54 29 2f 63 6e 3d-52 65 63 69 70 69 65 6e
0070: 74 73 2f 63 6e 3d 65 64-62 61 6e 74 69 00

```

Last, Joe sends the e-mail message. The client sets the following calculated subject properties on the E-mail object based on the subject text on Joe's submitted message by using the **RopSetProperties** ROP.

Property	PropertyID	Type	Value
PidTagSubjectPrefix (section 2.2.1.51)	0x0003	0x001F (PtypString)	Empty string
PidTagNormalizedSubject ([MS-OXOCAL] section 2.2.1.10)	0x0E1D	0x001F (PtypString)	"Order Form Issue"

The client then sends a **RopSubmitMessage** ROP request ([\[MS-OXOCAL\]](#) section 2.2.7.1) to ask the server to deliver this e-mail message to Ed Banti and sends a **RopRelease** ROP request ([\[MS-OXOCAL\]](#) section 2.2.15.3) to release the E-mail object.

For more details about the ROPs used in this example, see [\[MS-OXOCAL\]](#), [\[MS-OXOCAL\]](#), and section [2.2.4](#) of this document. For more details about a client's offline e-mail address book and recipient (2) address data entry, see [\[MS-OXOAB\]](#) and [\[MS-OXOABK\]](#).

4.5 Sending a Message with Voting Options

In this example, a user wants to send a message that has "Yes", "No", and "Maybe" voting options. To do so, the client constructs the message to contain a **PidLidVerbStream** property (section [2.2.1.65](#)).

The complete contents of the **PidLidVerbStream** property in this example are shown in the following stream. The other properties of the message are not specific to voting and are omitted.

```
0000: 02 01 03 00 00 00 04 00-00 00 03 59 65 73 08 49
0010: 50 4D 2E 4E 6F 74 65 00-03 59 65 73 00 00 00 00
0020: 00 00 00 00 00 00 01 00 00-00 02 00 00 00 02 00 00
0030: 00 01 00 00 00 00 FF FF FF-FF 04 00 00 00 02 4E 6F
0040: 08 49 50 4D 2E 4E 6F 74-65 00 02 4E 6F 00 00 00
0050: 00 00 00 00 00 00 01 00-00 00 02 00 00 00 02 00
0060: 00 00 02 00 00 00 FF FF-FF FF 04 00 00 00 05 4D
0070: 61 79 62 65 08 49 50 4D-2E 4E 6F 74 65 00 05 4D
0080: 61 79 62 65 00 00 00 00-00 00 00 00 00 01 00 00
0090: 00 02 00 00 00 02 00 00-00 03 00 00 00 FF FF FF
00A0: FF 04 01 03 59 00 65 00-73 00 03 59 00 65 00 73
00B0: 00 02 4E 00 6F 00 02 4E-00 6F 00 05 4D 00 61 00
00C0: 79 00 62 00 65 00 05 4D-00 61 00 79 00 62 00 65
00D0: 00
```

The first six bytes contain the **Version** and **Count** fields specified in section [2.2.1.65](#).

```
0000: 02 01 03 00 00 00
```

Version: 0x0102

Count: 0x00000003

This indicates that the structure contains three **VoteOption** structures. The first **VoteOption** structure begins at byte 0x0006.

```
0006: 04 00 00 00 03 59 65 73-08 49 50 4D 2E 4E 6F 74
0016: 65 00 03 59 65 73 00 00-00 00 00 00 00 00 00 01
0026: 00 00 00 02 00 00 00 02-00 00 00 01 00 00 00 FF
0036: FF FF FF
```

VerbType: 0x00000004

DisplayNameCount: 0x03

DisplayName: ANSI string (not null-terminated): "Yes"

MsgClsNameCount: 0x08

MsgClsName: ANSI string (not null-terminated): "IPM.Note"

Internal1StringCount: 0x00

DisplayNameCountRepeat: 0x03

DisplayNameRepeat: ANSI string (not null-terminated): "Yes"

Internal2: 0x00000000

Internal3: 0x00

fUseUSHeaders: False (0x00000000)

Internal4: 0x00000001

SendBehavior: 0x00000002 (SendPrompt)

Internal5: 0x00000002

ID: 0x00000001

Internal6: 0xFFFFFFFF

The second and third **VoteOption** structures (for "No" and "Maybe") begin at bytes 0x0039 and 0x006A respectively. The third **VoteOption** structure concludes at byte 0x00A0, and byte 0x00A1 begins the **Version2** field.

```
00A1: 04 01
```

Version2: 0x0104

This is followed by three **VoteOptionExtras** structures — a parallel array that contains additional information about the three **VoteOption** structures seen earlier. The first begins at byte 0x00A3.

```
00A3: 03 59 00 65 00 73 00 03-59 00 65 00 73 00
```

DisplayNameCount: 0x03

DisplayName: Unicode string (not null-terminated): "Yes"

DisplayNameCountRepeat: 0x03

DisplayNameRepeat: Unicode String (not null-terminated): "Yes"

The second and third **VoteOptionExtras** structures (for "No" and "Maybe") begin at bytes 0x00B1 and 0x00BB, respectively, and constitute the remainder of the buffer.

4.6 Sending Mail to a Specific Server

Ellen Adams is using a mail client that is connected to both her work and personal e-mail accounts. Her personal e-mail account is accessed through a mail protocol such as **Internet Message Access Protocol - Version 4 (IMAP4)** or POP3 and not using the protocol described in this and related documents. Her personal e-mail is set to deliver e-mail messages to a folder in her work account.

4.6.1 Initialization

When the mail client first initializes, it sends a **RopSetSpooler** ROP request ([\[MS-OXCROPS\]](#) section 2.2.7.4) to inform the server that the client wants to be responsible for routing e-mail messages to the **messaging transport**.

4.6.1.1 ROP Request Buffer

The ROP request buffer in this example resembles the following.

```
0000: 47 06 00
```

The composition of the bytes is as follows:

RopId: 0x47 (**RopSetSpooler** ROP ([\[MS-OXCROPS\]](#) section 2.2.7.4))

LogonID: 0x06

InputHandleIndex: 0x00 (handle to the Logon object)

4.6.1.2 ROP Response Buffer

The server then returns a ROP response buffer that resembles the following.

```
0000: 47 00 00 00 00 00
```

The composition of the response buffer is as follows:

RopId: 0x47 (**RopSetSpooler** ROP ([\[MS-OXCROPS\]](#) section 2.2.7.4))

InputHandleIndex: 0x00

ReturnValue: ecNone (Success)

4.6.2 Submitting the Message to the Spooler Queue Folder

Ellen then sends an e-mail message from her work account. The client follows the example in section [4.1](#), except that it sets the **NeedsSpooler** (0x2) bit in the **SubmitFlags** field, as well as setting a property or somehow informing the spooler which mail transport to use.

The server places the message in the spooler queue folder (the FID ([\[MS-OXCDATA\]](#) section 2.2.1.1) of this folder is returned in the response buffer of a **RopLogon** ROP request ([\[MS-OXCROPS\]](#) section 2.2.3.1)).

4.6.3 Locking the Message in the Spooler Queue Folder for Processing

Next, the client finds that an e-mail message has been placed in the spooler queue folder. After determining that it can handle the message, the client sends the **RopSpoolerLockMessage** ROP request ([\[MS-OXCROPS\]](#) section 2.2.7.5) to lock the message.

4.6.3.1 ROP Request Buffer

The ROP request buffer in this example resembles the following.

0000: 48 06 00 01 00 00 03 BB-97 31 A7 00

The composition of the bytes is as follows:

RopId: 0x48 (**RopSpoolerLockMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.7.5))

LogonID: 0x06

InputHandleIndex: 0x00 (handle to the Logon object)

MessageId: 0001-0003BB9731A7

LockState: 0x00 (lock)

4.6.3.2 ROP Response Buffer

The server then returns a ROP response buffer that resembles the following.

0000: 48 00 00 00 00 00 00

The composition of the response buffer is as follows:

RopId: 0x48 (**RopSpoolerLockMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.7.5))

InputHandleIndex: 0x00

ReturnValue: ecNone (success) (0x00000000)

4.6.4 Determining the Transport Folder

The client determines which server to route the e-mail message to (Ellen's work server). The server can be the same as or different from the server that is holding the spooler queue. The client sends a **RopGetTransportFolder** ROP request ([\[MS-OXCROPS\]](#) section 2.2.7.8) to request the location of a temporary folder for transport.

4.6.4.1 ROP Request Buffer

The ROP request buffer in this example resembles the following.

0000: 6D 07 01

The composition of the bytes is as follows:

RopId: 0x6D (**RopGetTransportFolder** ROP ([\[MS-OXCROPS\]](#) section 2.2.7.8))

LogonID: 0x07

InputHandleIndex: 0x01 (handle to the Logon object)

4.6.4.2 ROP Response Buffer

The server then returns a ROP response buffer with the FID ([\[MS-OXCDATA\]](#) section 2.2.1.1) of a folder that can be used to store temporary transport objects.

```
0000: 6D 01 00 00 00 00 01 00-00 00 00 00 00 25
```

The composition of the response buffer is as follows:

RopId: 0x6D (**RopGetTransportFolder** ROP ([\[MS-OXCROPS\]](#) section 2.2.7.8))

InputHandleIndex: 0x01

ReturnValue: ecNone (success) (0x00000000)

FolderId: 0001-0000000000025

4.6.5 Sending the Message

The client examines the locked e-mail message, performs any required processing (for example, it determines whether there are any recipients (1) that the server cannot deliver to), and creates a copy of the message to be delivered in the folder just retrieved by using the **RopCreateMessage** ROP request ([\[MS-OXCROPS\]](#) section 2.2.6.2).

The client then sends a **RopTransportSend** ROP request ([\[MS-OXCROPS\]](#) section 2.2.7.6) to have the server send the message.

4.6.5.1 ROP Request Buffer

The ROP request buffer in this example resembles the following.

```
0000: 4A 07 00
```

The composition of the bytes is as follows:

RopId: 0x4A (**RopTransportSend** ROP ([\[MS-OXCROPS\]](#) section 2.2.7.6))

LogonID: 0x07

InputHandleIndex: 0x00 (handle to the message from the **RopCreateMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.6.2))

4.6.5.2 ROP Response Buffer

The server then returns the following ROP response buffer.

```
0000: 4A 00 00 00 00 00 08-00 40 00 48 00 B0 5D 07
0010: 11 A1 AF C8 01 0A 00 47-00 0F 01 04 80 1E 00 1A
0020: 0C 75 73 65 72 31 00 02-01 19 0C 7C 00 00 00 00
0030: 00 DC A7 40 C8 C0 42 10-1A B4 B9 08 00 2B 2F E1
0040: 82 01 00 00 00 00 00 00-00 2F 4F 3D 46 49 52 53
0050: 54 20 4F 52 47 41 4E 49-5A 41 54 49 4F 4E 2F 4F
0060: 55 3D 45 58 43 48 41 4E-47 45 20 41 44 4D 49 4E
0070: 49 53 54 52 41 54 49 56-45 20 47 52 4F 55 50 20
0080: 28 46 59 44 49 42 4F 48-46 32 33 53 50 44 4C 54
0090: 29 2F 43 4E 3D 52 45 43-49 50 49 45 4E 54 53 2F
00a0: 43 4E 3D 55 53 45 52 31-00 02 01 1D 0C 63 00 45
00b0: 58 3A 2F 4F 3D 46 49 52-53 54 20 4F 52 47 41 4E
00c0: 49 5A 41 54 49 4F 4E 2F-4F 55 3D 45 58 43 48 41
00d0: 4E 47 45 20 41 44 4D 49-4E 49 53 54 52 41 54 49
```

```

00e0: 56 45 20 47 52 4F 55 50-20 28 46 59 44 49 42 4F
00f0: 48 46 32 33 53 50 44 4C-54 29 2F 43 4E 3D 52 45
0100: 43 49 50 49 45 4E 54 53-2F 43 4E 3D 55 53 45 52
0110: 31 00 1E 00 42 00 75 73-65 72 31 00 02 01 41 00
0120: 7C 00 00 00 00 00 DC A7-40 C8 C0 42 10 1A B4 B9
0130: 08 00 2B 2F E1 82 01 00-00 00 00 00 00 2F 4F
0140: 3D 46 49 52 53 54 20 4F-52 47 41 4E 49 5A 41 54
0150: 49 4F 4E 2F 4F 55 3D 45-58 43 48 41 4E 47 45 20
0160: 41 44 4D 49 4E 49 53 54-52 41 54 49 56 45 20 47
0170: 52 4F 55 50 20 28 46 59-44 49 42 4F 48 46 32 33
0180: 53 50 44 4C 54 29 2F 43-4E 3D 52 45 43 49 50 49
0190: 45 4E 54 53 2F 43 4E 3D-55 53 45 52 31 00 02 01
01a0: 3B 00 63 00 45 58 3A 2F-4F 3D 46 49 52 53 54 20
01b0: 4F 52 47 41 4E 49 5A 41-54 49 4F 4E 2F 4F 55 3D
01c0: 45 58 43 48 41 4E 47 45-20 41 44 4D 49 4E 49 53
01d0: 54 52 41 54 49 56 45 20-47 52 4F 55 50 20 28 46
01e0: 59 44 49 42 4F 48 46 32-33 53 50 44 4C 54 29 2F
01f0: 43 4E 3D 52 45 43 49 50-49 45 4E 54 53 2F 43 4E
0200: 3D 55 53 45 52 31 00

```

The composition of the response buffer is as follows:

RopId: 0x4A (**RopTransportSend** ROP ([\[MS-OXCROPS\]](#) section 2.2.7.6))

InputHandleIndex: 0x00

ReturnValue: ecNone (success) (0x00000000)

NoPropertiesReturned: 0x00 (FALSE)

PropertyValueCount: 0x08

PropertyValues: The properties that are included in the response buffer are described in the following table. Types are described in [\[MS-OXCDATA\]](#) section 2.11.1.

Property ID	Property name	Type	Data
0x00480040	PidTagProviderSubmitTime ([MS-OXPROPS] section 2.947)	PtypTime	2008/05/06 17:46:09.035
0x00470102	PidTagMessageSubmissionId (section 2.2.1.70)	PtypBinary	Error: 0x8004010f (MAPI_E_NOT_FOUND)
0x0C1A001E	PidTagSenderName (section 2.2.1.43)	PtypString 8	"user1"
0x0C190102	PidTagSenderEntryId (section 2.2.1.42)	PtypBinary	See the data for the PidTagSenderEntryId property following the table (1).
0x0C1D0102	PidTagSenderSearchKey (section 2.2.1.44)	PtypBinary	See the data for the PidTagSenderSearchKey property following the table (2).
0x0042001E	PidTagSentRepresentingName (section 2.2.1.49)	PtypString 8	"user1"

Property ID	Property name	Type	Data
0x00410102	PidTagSentRepresentingEntryId (section 2.2.1.48)	PtypBinary	See the data for the PidTagSentRepresentingEntryId property following the table (3).
0x003B0102	PidTagSentRepresentingSearchKey (section 2.2.1.50)	PtypBinary	See the data for the PidTagSentRepresentingSearchKey property following the table (4).

Data for the **PidTagSenderEntryId** property (1)

Size: 124

```

0000: 00 00 00 00 DC A7 40 C8-C0 42 10 1A B4 B9 08 00 .....@..B.....
0010: 2B 2F E1 82 01 00 00 00-00 00 00 00 2F 4F 3D 46 +/...../O=F
0020: 49 52 53 54 20 4F 52 47-41 4E 49 5A 41 54 49 4F IRST ORGANIZATIO
0030: 4E 2F 4F 55 3D 45 58 43-48 41 4E 47 45 20 41 44 N/OU=EXCHANGE AD
0040: 4D 49 4E 49 53 54 52 41-54 49 56 45 20 47 52 4F MINISTRATIVE GRO
0050: 55 50 20 28 46 59 44 49-42 4F 48 46 32 33 53 50 UP (FYDIBOHF23SP
0060: 44 4C 54 29 2F 43 4E 3D-52 45 43 49 50 49 45 4E DLT)/CN=RECIPIEN
0070: 54 53 2F 43 4E 3D 55 53-45 52 31 00 TS/CN=USER1.

```

Data for the **PidTagSenderSearchKey** property (2)

Size: 99

```

0000: 45 58 3A 2F 4F 3D 46 49-52 53 54 20 4F 52 47 41 EX:/O=FIRST ORGA
0010: 4E 49 5A 41 54 49 4F 4E-2F 4F 55 3D 45 58 43 48 NIZATION/OU=EXCH
0020: 41 4E 47 45 20 41 44 4D-49 4E 49 53 54 52 41 54 ANGE ADMINISTRAT
0030: 49 56 45 20 47 52 4F 55-50 20 28 46 59 44 49 42 IVE GROUP (FYDIB
0040: 4F 48 46 32 33 53 50 44-4C 54 29 2F 43 4E 3D 52 OHF23SPDLT)/CN=R
0050: 45 43 49 50 49 45 4E 54-53 2F 43 4E 3D 55 53 45 ECIPIENTS/CN=USE
0060: 52 31 00 R1.

```

Data for the **PidTagSentRepresentingEntryId** property (3)

Size: 124

```

0000: 00 00 00 00 DC A7 40 C8-C0 42 10 1A B4 B9 08 00 .....@..B.....
0010: 2B 2F E1 82 01 00 00 00-00 00 00 00 2F 4F 3D 46 +/...../O=F
0020: 49 52 53 54 20 4F 52 47-41 4E 49 5A 41 54 49 4F IRST ORGANIZATIO
0030: 4E 2F 4F 55 3D 45 58 43-48 41 4E 47 45 20 41 44 N/OU=EXCHANGE AD
0040: 4D 49 4E 49 53 54 52 41-54 49 56 45 20 47 52 4F MINISTRATIVE GRO
0050: 55 50 20 28 46 59 44 49-42 4F 48 46 32 33 53 50 UP (FYDIBOHF23SP
0060: 44 4C 54 29 2F 43 4E 3D-52 45 43 49 50 49 45 4E DLT)/CN=RECIPIEN
0070: 54 53 2F 43 4E 3D 55 53-45 52 31 00 TS/CN=USER1.

```

Data for the **PidTagSentRepresentingSearchKey** property (4)

Size: 99

```
0000: 45 58 3A 2F 4F 3D 46 49-52 53 54 20 4F 52 47 41 EX:/O=FIRST ORGA
0010: 4E 49 5A 41 54 49 4F 4E-2F 4F 55 3D 45 58 43 48 NIZATION/OU=EXCH
0020: 41 4E 47 45 20 41 44 4D-49 4E 49 53 54 52 41 54 ANGE ADMINISTRAT
0030: 49 56 45 20 47 52 4F 55-50 20 28 46 59 44 49 42 IVE GROUP (FYDIB
0040: 4F 48 46 32 33 53 50 44-4C 54 29 2F 43 4E 3D 52 OHF23SPDLT)/CN=R
0050: 45 43 49 50 49 45 4E 54-53 2F 43 4E 3D 55 53 45 ECIPIENTS/CN=USE
0060: 52 31 00 R1.
```

4.6.6 Marking the Message as Ready for Post-Send Server Processing

Finally, the client sends the **RopSpoolerLockMessage** ROP request ([\[MS-OXCROPS\]](#) section 2.2.7.5) with the finish flag to the server to have it perform any postprocessing on the sent message.

4.6.6.1 ROP Request Buffer

The ROP request buffer in this example resembles the following.

```
0000: 48 06 00 01 00 00 03 BB-97 31 A7 02
```

The composition of the bytes is as follows:

RopId: 0x48 (**RopSpoolerLockMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.7.5))

LogonID: 0x06

InputHandleIndex: 0x00 (handle to the Logon object)

MessageId: 0001-0003BB9731A7

LockState: 0x02 (finish)

4.6.6.2 ROP Response Buffer

The server then returns a ROP response buffer that resembles the following.

```
0000: 48 00 00 00 00 00
```

The composition of the response buffer is as follows:

RopId: 0x48 (**RopSpoolerLockMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.7.5))

InputHandleIndex: 0x00

ReturnValue: ecNone (success) (0x00000000)

5 Security

5.1 Security Considerations for Implementers

There are no security considerations specific to this protocol. General security considerations pertaining to the underlying RPC-based transport apply as described in [\[MS-OXCROPS\]](#).

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

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

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

[<1> Section 2.2.1.2:](#) The computation of the value of the **PidTagConversationId** property is not supported by Exchange 2003, Exchange 2007, Office Outlook 2003, and Office Outlook 2007.

[<2> Section 2.2.1.5:](#) Exchange 2003 and Exchange 2007 do not support the **PidTagConversationTopic** property.

[<3> Section 2.2.1.6:](#) The **PidTagDeferredDeliveryTime** property is used by Exchange 2003. In Exchange 2007, only the **PidTagDeferredSendTime** property (section [2.2.3.4](#)) is used.

[<4> Section 2.2.1.76:](#) Outlook 15 Technical Preview stores this property.

[<5> Section 2.2.4.4:](#) Outlook 2010 does not send a **RopOptionsData** ROP request ([\[MS-OXCROPS\]](#) section 2.2.7.9).

[<6> Section 2.2.4.4.2:](#) Exchange 2003, Exchange 2007, and Exchange 2010 do not return an empty array.

[<7> Section 3.2.4.6.2:](#) Office Outlook 2007 also uses the **PidLidVerbStream** property (section [2.2.1.65](#)) for actions unrelated to voting that are not covered by this protocol. Each of these actions has a specific **VerbType** associated with it. The format of the **VoteOption** structure is identical for these actions that are unrelated to voting; however, the internal values that are specific in the structure will vary. Future versions of Outlook might further define additional **VerbTypes**; it is therefore advised that clients ignore **VoteOption** structures that do not specify **VerbTypes** that they understand. Likewise, Office Outlook 2007 SP1 ignores **VoteOption** structures with unknown **VerbTypes**.

<8> [Section 3.2.4.6.4](#): Office Outlook 2007 uses a system similar to meeting responses in order to track voting options. When it receives a voting response, it finds the initial voting message in the Sent Items folder. It then updates the recipient table for the recipient (2) who sent the response to store the index of their response. If the user opens a voting message from the Sent Items folder, it then sums the total of each response received thus far from the recipient table and displays it to the user.

<9> [Section 3.2.4.8](#): Office Outlook 2003 and Office Outlook 2007 set the **PidTagNextSendAcct** property (section [2.2.1.56](#)) to a user-specified value before submitting the message by using the **RopSubmitMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.7.1) to inform the spooler of the desired mail transport to use.

<10> [Section 3.2.5.3](#): Office Outlook 2003 and Office Outlook 2007 examine the **PidTagNextSendAcct** property (section [2.2.1.56](#)).

<11> [Section 3.3.5.3](#): Exchange 2007 SP2 returns ecNone (0x00000000) instead of ecNullObject when an invalid object handle reference is passed to the **RopGetAddressTypes** ROP ([\[MS-OXCROPS\]](#) section 2.2.7.3).

<12> [Section 3.3.5.5](#): On Exchange 2003 and Exchange 2007, the **RopGetTransportFolder** ROP ([\[MS-OXCROPS\]](#) section 2.2.7.8) returns ecNone instead of ecNullObject when an invalid input handle is provided.

<13> [Section 3.3.5.6](#): Exchange 2010 will return Success (0x00000000) for the **RopSpoolerLockMessage** ROP ([\[MS-OXCROPS\]](#) section 2.2.7.5) even if the message is not in the spooler queue.

<14> [Section 3.3.5.9](#): Exchange 2007 returns Success (0x00000000) for the **RopOptionsData** ROP ([\[MS-OXCROPS\]](#) section 2.2.7.9) regardless of whether the call succeeds or fails.

7 Change Tracking

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
2.2.1.76 PidTagCurrentDraftEntryId	Added a section that specifies the PidTagCurrentDraftEntryId property.	Y	New content added.
3.2.4.1.1 Sending the Message as a Represented Sender	Changed the name of the section from "Represented Sender Properties" to "Sending the Message as a Represented Sender".	N	Content updated.
3.2.4.1.2 Sending the Message as the Actual Sender	Changed the name of the section from "Actual Sender Properties" to "Sending the Message as the Actual Sender".	N	Content updated.
3.2.4.2 Deferring Message Send	Changed the name of the section from "Deferring Sending a Message" to "Deferring Message Send".	N	Content updated.
3.2.4.6 Soliciting Votes from Voters	Changed the name of the section from "Client to Client Interop: Voting" to "Soliciting Votes from Voters".	N	Content updated.
3.2.4.7 Sending Mail Through a Specific Server	Changed the name of the section from "Controlling the Sending of Mail" to "Sending Mail Through a Specific Server".	N	Content updated.
6 Appendix A: Product Behavior	Added Exchange 15 Technical Preview and Outlook 15 Technical Preview to the list of applicable product versions.	Y	Content updated.
	Removed the "PidTagSenderTelephoneNumber Property" section.	N	Content removed.

8 Index

A

Aborting a message submission example
[overview](#) 75
[ROP request buffer](#) 75
[ROP response buffer](#) 76

Abstract data model
[client](#) 60
[server](#) 65

Abstract data model - server
[Global ADM data type](#) 65
[per E-mail object](#) 66
[per mailbox](#) 65
[User ADM data type](#) 66

Abstract data model - server
[per Message object](#) 66

[Applicability](#) 14
[Attachments E-mail object properties](#) 35

C

[Capability negotiation](#) 14
[Categories and keywords E-mail object properties](#) 36
[Change tracking](#) 91

Client
[abstract data model](#) 60
[initialization](#) 61
[other local events](#) 65
[timer events](#) 65
[timers](#) 61

Client - higher layer triggered events
[client-to-client interop: voting](#) 62
[controlling the sending of mail](#) 63
[deferring sending a message](#) 62
[delivering mail to the server](#) 64
[optimizing send](#) 62
[processing a mail in the spooler queue](#) 64
[resending a message](#) 62
[sending a message](#) 61
[sending a message with expiry time](#) 62

Client - sequencing rules
[sending a RopAbortSubmit ROP request](#) 64
[sending a RopLockSpoolerMessage ROP request](#) 65
[sending a RopSubmitMessage ROP request](#) 64
[sending a RopTransportNewMail ROP request](#) 65

[Contacts E-mail object properties](#) 36

Controlling sending and delivery of mail
[overview](#) 14

Creating
opening
[and saving e-mail objects overview](#) 12

D

Data model - abstract
[client](#) 60

[server](#) 65
[Delivery receipt overview](#) 13

E

E-mail object properties
[attachments](#) 35
[categories and keywords](#) 36
[contacts](#) 36
[flags](#) 36
[PidLidAutoProcessState property](#) 36
[PidLidInternetAccountName property](#) 35
[PidLidInternetAccountStamp property](#) 35
[PidLidUseTnef property](#) 35
[PidLidVerbResponse property](#) 40
[PidLidVerbStream property](#) 36
[PidTagAutoResponseSuppress property](#) 40
[PidTagBlockStatus property](#) 19
[PidTagConversationId property](#) 20
[PidTagConversationIndex property](#) 20
[PidTagConversationIndexTracking property](#) 22
[PidTagConversationTopic property](#) 22
[PidTagDeferredDeliveryTime property](#) 22
[PidTagDelegatedByRule property](#) 42
[PidTagDisplayBcc property](#) 23
[PidTagDisplayCc property](#) 23
[PidTagDisplayTo property](#) 23
[PidTagIconIndex property](#) 23
[PidTagInReplyToId property](#) 25
[PidTagInternetMailOverrideFormat property](#) 24
[PidTagInternetMessageId property](#) 25
[PidTagLastVerbExecuted property](#) 25
[PidTagLastVerbExecutionTime property](#) 28
[PidTagListHelp property](#) 42
[PidTagListSubscribe property](#) 42
[PidTagListUnsubscribe property](#) 42
[PidTagMessageCcMe property](#) 28
[PidTagMessageClass property](#) 28
[PidTagMessageEditorFormat property](#) 41
[PidTagMessageRecipientMe property](#) 28
[PidTagMessageSubmissionId property](#) 41
[PidTagMessageToMe property](#) 28
[PidTagNextSendAcct property](#) 35
[PidTagOriginalSensitivity property](#) 29
[PidTagOriginatorDeliveryReportRequested property](#) 28
[PidTagOriginatorNonDeliveryReportRequested property](#) 29
[PidTagPrimarySendAccount property](#) 35
[PidTagReadReceiptRequested property](#) 30
[PidTagReceivedByAddressType property](#) 30
[PidTagReceivedByEmailAddress property](#) 30
[PidTagReceivedByEntryId property](#) 31
[PidTagReceivedByName property](#) 31
[PidTagReceivedBySearchKey property](#) 31
[PidTagReceivedRepresentingAddressType property](#) 29

[PidTagReceivedRepresentingEmailAddress property](#) 29
[PidTagReceivedRepresentingEntryId property](#) 29
[PidTagReceivedRepresentingName property](#) 30
[PidTagReceivedRepresentingSearchKey property](#) 30
[PidTagRecipientReassignmentProhibited property](#) 31
[PidTagReplyRecipientEntries property](#) 31
[PidTagReplyRecipientNames property](#) 32
[PidTagReplyRequested property](#) 32
[PidTagResponseRequested property](#) 32
[PidTagSenderAddressType property](#) 32
[PidTagSenderEmailAddress property](#) 32
[PidTagSenderEntryId property](#) 33
[PidTagSenderIdStatus property](#) 41
[PidTagSenderName property](#) 33
[PidTagSenderSearchKey property](#) 33
[PidTagSenderSmtpAddress property](#) 33
[PidTagSendRichInfo property](#) 32
[PidTagSentRepresentingAddressType property](#) 33
[PidTagSentRepresentingEmailAddress property](#) 33
[PidTagSentRepresentingEntryId property](#) 34
[PidTagSentRepresentingName property](#) 34
[PidTagSentRepresentingSearchKey property](#) 34
[PidTagSubjectPrefix property](#) 34
[PidTagTargetEntryId property](#) 40
[PidTagTransportMessageHeaders property](#) 34
[Recipients](#) 36
[Reminders](#) 36
[E-Mail Object Properties message](#) 19
 E-mail objects overview
 Creating
 opening
 [and saving e-mail objects](#) 12
 [replying and forwarding messages](#) 13
 [sending messages](#) 13
 E-mail submission properties
 [PidTagClientSubmitTime property](#) 52
 [PidTagDeferredSendNumber property](#) 50
 [PidTagDeferredSendTime property](#) 50
 [PidTagDeferredSendUnits property](#) 50
 [PidTagDeleteAfterSubmit property](#) 51
 [PidTagExpiryNumber property](#) 51
 [PidTagExpiryTime property](#) 51
 [PidTagExpiryUnits property](#) 51
 [PidTagMessageDeliveryTime property](#) 52
 [PidTagRecipientType property](#) 49
 [PidTagSentMailSvrEID property](#) 52
[E-Mail Submission Properties message](#) 49
 Examples - aborting a message submission
 [overview](#) 75
 [ROP request buffer](#) 75
 [ROP response buffer](#) 76
[Examples - sending a message with voting options](#) 80
[Examples - sending an e-mail message from a messaging user to another messaging user](#) 76
 Examples - sending mail to a specific server
 [determining the transport folder overview](#) 83

[determining the transport folder ROP request buffer](#) 83
[determining the transport folder ROP response buffer](#) 83
[initialization overview](#) 82
[initialization ROP request buffer](#) 82
[initialization ROP response buffer](#) 82
[locking the message in the spooler queue folder for processing overview](#) 82
[locking the message in the spooler queue folder for processing ROP request buffer](#) 82
[locking the message in the spooler queue folder for processing ROP response buffer](#) 83
[marking the message as ready for post-send server processing overview](#) 87
[marking the message as ready for post-send server processing ROP request buffer](#) 87
[marking the message as ready for post-send server processing ROP response buffer](#) 87
[overview](#) 81
[sending the message overview](#) 84
[sending the message ROP request buffer](#) 84
[sending the message ROP response buffer](#) 84
[submitting the message to the spooler queue folder](#) 82
 Examples - submitting a deferred message
 [overview](#) 74
 [ROP request buffer](#) 74
 [ROP response buffer](#) 74
 Examples - submitting a message
 [overview](#) 73
 [ROP request buffer](#) 73
 [ROP response buffer](#) 73

F

[Fields - vendor-extensible](#) 14
[Flags E-mail object properties](#) 36

G

[Glossary](#) 9

H

Higher layer triggered events - client
 [client-to-client interop: voting](#) 62
 [controlling the sending of mail](#) 63
 [deferring sending a message](#) 62
 [delivering mail to the server](#) 64
 [optimizing send](#) 62
 [processing a mail in the spooler queue](#) 64
 [resending a message](#) 62
 [sending a message](#) 61
 [sending a message with expiry time](#) 62
 Higher-layer triggered events
 [server](#) 66

I

[Implementer - security considerations](#) 88
[Index of security parameters](#) 88

[Informative references](#) 12

Initialization

[client](#) 61

[server](#) 66

[Introduction](#) 9

M

Message delivery ROPs

[RopAbortSubmit ROP](#) 53

[RopGetAddressTypes ROP](#) 53

[RopOptionsData ROP](#) 54

[RopSubmitMessage ROP](#) 52

Message processing - client

[sending a RopAbortSubmit ROP request](#) 64

[sending a RopLockSpoolerMessage ROP request](#) 65

[sending a RopSubmitMessage ROP request](#) 64

[sending a RopTransportNewMail ROP request](#) 65

Message processing - server

[receiving a RopAbortSubmit ROP request](#) 70

[receiving a RopGetAddressTypes ROP request](#) 70

[receiving a RopGetTransportFolder ROP request](#) 71

[receiving a RopOptionsData ROP request](#) 72

[receiving a RopSetSpooler ROP request](#) 71

[receiving a RopSpoolerLockMessage ROP request](#) 71

[receiving a RopSubmitMessage ROP request](#) 67

[receiving a RopTransportNewMail ROP request](#) 72

[receiving a RopTransportSend ROP request](#) 72

Message status reports properties

[PidTagMessageClass property \(section 2.2.1.76 42, section 2.2.2.1 43\)](#)

[PidTagOriginalDeliveryTime property](#) 43

[PidTagOriginalDisplayBcc property](#) 44

[PidTagOriginalDisplayCc property](#) 43

[PidTagOriginalDisplayTo property](#) 43

[PidTagOriginalSenderAddressType property](#) 44

[PidTagOriginalSenderEmailAddress property](#) 44

[PidTagOriginalSenderEntryId property](#) 44

[PidTagOriginalSenderName property](#) 44

[PidTagOriginalSenderSearchKey property](#) 44

[PidTagOriginalSentRepresentingAddressType property](#) 44

[PidTagOriginalSentRepresentingEmailAddress property](#) 45

[PidTagOriginalSentRepresentingEntryId property](#) 45

[PidTagOriginalSentRepresentingName property](#) 45

[PidTagOriginalSentRepresentingSearchKey property](#) 45

[PidTagOriginalSubject property](#) 45

[PidTagOriginalSubmitTime property](#) 45

[PidTagParentKey property](#) 46

[PidTagReadReceiptAddressType property](#) 48

[PidTagReadReceiptEmailAddress property](#) 48

[PidTagReadReceiptEntryId property](#) 49

[PidTagReadReceiptName property](#) 49

[PidTagReadReceiptSearchKey property](#) 49

[PidTagReportEntryId property](#) 46

[PidTagReportName property](#) 46

[PidTagReportSearchKey property](#) 46

[PidTagReportTag property](#) 46

[PidTagReportText property](#) 48

[Message syntax](#) 15

Messages

[E-Mail Object Properties](#) 19

[E-Mail Submission Properties](#) 49

[Spooler and Transport ROPs](#) 55

[syntax](#) 15

[transport](#) 15

N

[Non-delivery receipt overview](#) 13

[Non-read receipt overview](#) 13

[Normative references](#) 11

O

Other local events

[client](#) 65

[server](#) 72

Overview

[controlling sending and delivery of mail](#) 14

[report messages](#) 13

[voting and tracking](#) 13

[Overview \(synopsis\)](#) 12

P

[Parameters - security index](#) 88

[PidLidAutoProcessState E-mail object property](#) 36

[PidLidInternetAccountName E-mail object property](#) 35

[PidLidInternetAccountStamp E-mail object property](#) 35

[PidLidUseTnef E-mail object property](#) 35

[PidLidVerbResponse E-mail object property](#) 40

[PidLidVerbStream E-mail object property](#) 36

[PidTagAutoResponseSuppress E-mail object property](#) 40

[PidTagBlockStatus E-mail object property](#) 19

[PidTagClientSubmitTime e-mail submission property](#) 52

[PidTagConversationId E-mail object property](#) 20

[PidTagConversationIndex E-mail object property](#) 20

[PidTagConversationIndexTracking E-mail object property](#) 22

[PidTagConversationTopic E-mail object property](#) 22

[PidTagDeferredDeliveryTime E-mail object property](#) 22

[PidTagDeferredSendNumber e-mail submission property](#) 50

[PidTagDeferredSendTime e-mail submission property](#) 50

[PidTagDeferredSendUnits e-mail submission property](#) 50

[PidTagDelegatedByRule E-mail object property](#) 42

[PidTagDeleteAfterSubmit e-mail submission property](#) 51

[PidTagDisplayBcc E-mail object property](#) 23

[PidTagDisplayCc E-mail object property](#) 23
[PidTagDisplayTo E-mail object property](#) 23
[PidTagExpiryNumber e-mail submission property](#) 51
[PidTagExpiryTime e-mail submission property](#) 51
[PidTagExpiryUnits e-mail submission property](#) 51
[PidTagIconIndex E-mail object property](#) 23
[PidTagInReplyToId E-mail object property](#) 25
[PidTagInternetMailOverrideFormat E-mail object property](#) 24
[PidTagInternetMessageId E-mail object property](#) 25
[PidTagLastVerbExecuted E-mail object property](#) 25
[PidTagLastVerbExecutionTime E-mail object property](#) 28
[PidTagListHelp E-mail object property](#) 42
[PidTagListSubscribe E-mail object property](#) 42
[PidTagListUnsubscribe E-mail object property](#) 42
[PidTagMessageCcMe E-mail object property](#) 28
[PidTagMessageClass E-mail object property](#) 28
[PidTagMessageClass message status report property](#) ([section 2.2.1.76](#) 42, [section 2.2.2.1](#) 43)
[PidTagMessageDeliveryTime e-mail submission property](#) 52
[PidTagMessageEditorFormat E-mail object property](#) 41
[PidTagMessageRecipientMe E-mail object property](#) 28
[PidTagMessageSubmissionId E-mail object property](#) 41
[PidTagMessageToMe E-mail object property](#) 28
[PidTagNextSendAcct E-mail object property](#) 35
[PidTagOriginalDeliveryTime message status report property](#) 43
[PidTagOriginalDisplayBcc message status report property](#) 44
[PidTagOriginalDisplayCc message status report property](#) 43
[PidTagOriginalDisplayTo message status report property](#) 43
[PidTagOriginalSenderAddressType message status report property](#) 44
[PidTagOriginalSenderEmailAddress message status report property](#) 44
[PidTagOriginalSenderEntryId message status report property](#) 44
[PidTagOriginalSenderName message status report property](#) 44
[PidTagOriginalSenderSearchKey message status report property](#) 44
[PidTagOriginalSensitivity E-mail object property](#) 29
[PidTagOriginalSentRepresentingAddressType message status report property](#) 44
[PidTagOriginalSentRepresentingEmailAddress message status report property](#) 45
[PidTagOriginalSentRepresentingEntryId message status report property](#) 45
[PidTagOriginalSentRepresentingName message status report property](#) 45
[PidTagOriginalSentRepresentingSearchKey message status report property](#) 45
[PidTagOriginalSubject message status report property](#) 45
[PidTagOriginalSubmitTime message status report property](#) 45
[PidTagOriginatorDeliveryReportRequested E-mail object property](#) 28
[PidTagOriginatorNonDeliveryReportRequested E-mail object property](#) 29
[PidTagParentKey message status report property](#) 46
[PidTagPrimarySendAccount E-mail object property](#) 35
[PidTagReadReceiptAddressType message status report property](#) 48
[PidTagReadReceiptEmailAddress message status report property](#) 48
[PidTagReadReceiptEntryId message status report property](#) 49
[PidTagReadReceiptName message status report property](#) 49
[PidTagReadReceiptRequested E-mail object property](#) 30
[PidTagReadReceiptSearchKey message status report property](#) 49
[PidTagReceivedByAddressType E-mail object property](#) 30
[PidTagReceivedByEmailAddress E-mail object property](#) 30
[PidTagReceivedByEntryId E-mail object property](#) 31
[PidTagReceivedByName E-mail object property](#) 31
[PidTagReceivedBySearchKey E-mail object property](#) 31
[PidTagReceivedRepresentingAddressType E-mail object property](#) 29
[PidTagReceivedRepresentingEmailAddress E-mail object property](#) 29
[PidTagReceivedRepresentingEntryId E-mail object property](#) 29
[PidTagReceivedRepresentingName E-mail object property](#) 30
[PidTagReceivedRepresentingSearchKey E-mail object property](#) 30
[PidTagRecipientReassignmentProhibited E-mail object property](#) 31
[PidTagRecipientType e-mail submission property](#) 49
[PidTagReplyRecipientEntries E-mail object property](#) 31
[PidTagReplyRecipientNames E-mail object property](#) 32
[PidTagReplyRequested E-mail object property](#) 32
[PidTagReportEntryId message status report property](#) 46
[PidTagReportName message status report property](#) 46
[PidTagReportSearchKey message status report property](#) 46
[PidTagReportTag message status report property](#) 46
[PidTagReportText message status report property](#) 48
[PidTagResponseRequested E-mail object property](#) 32
[PidTagSenderAddressType E-mail object property](#) 32

[PidTagSenderEmailAddress E-mail object property](#) 32
[PidTagSenderEntryId E-mail object property](#) 33
[PidTagSenderIdStatus E-mail object property](#) 41
[PidTagSenderName E-mail object property](#) 33
[PidTagSenderSearchKey E-mail object property](#) 33
[PidTagSenderSmtAddress E-mail object property](#) 33
[PidTagSendRichInfo E-mail object property](#) 32
[PidTagSentMailSvrEID e-mail submission property](#) 52
[PidTagSentRepresentingAddressType E-mail object property](#) 33
[PidTagSentRepresentingEmailAddress E-mail object property](#) 33
[PidTagSentRepresentingEntryId E-mail object property](#) 34
[PidTagSentRepresentingName E-mail object property](#) 34
[PidTagSentRepresentingSearchKey E-mail object property](#) 34
[PidTagSubjectPrefix E-mail object property](#) 34
[PidTagTargetEntryId E-mail object property](#) 40
[PidTagTransportMessageHeaders E-mail object property](#) 34
[Preconditions](#) 14
[Prerequisites](#) 14
[Product behavior](#) 89

R

[Read receipt overview](#) 13
[Recipients E-mail object properties](#) 36
References
 [informative](#) 12
 [normative](#) 11
[Relationship to other protocols](#) 14
[Reminders E-mail object properties](#) 36
[Replying and forwarding messages overview](#) 13
Report messages
 [delivery receipt](#) 13
 [non-delivery receipt](#) 13
 [non-read receipt](#) 13
 [overview](#) 13
 [read receipt](#) 13
[RopAbortSubmit message delivery ROP](#) 53
[RopGetAddressTypes message delivery ROP](#) 53
[RopGetTransportFolder spooler and transfer ROP](#) 55
[RopOptionsData message delivery ROP](#) 54
[RopSetSpooler spooler and transfer ROP](#) 55
[RopSpoolerLockMessage spooler and transfer ROP](#) 56
[RopSubmitMessage message delivery ROP](#) 52
[RopTransportNewMail spooler and transfer ROP](#) 57
[RopTransportSend spooler and transfer ROP](#) 56

S

Security
 [implementer considerations](#) 88
 [parameter index](#) 88
[Sending a message with voting options example](#) 80

[Sending an e-mail message from a messaging user to another messaging user example](#) 76
Sending mail to a specific server example – determining the transport folder
 [overview](#) 83
 [ROP request buffer](#) 83
 [ROP response buffer](#) 83
Sending mail to a specific server example – initialization
 [overview](#) 82
 [ROP request buffer](#) 82
 [ROP response buffer](#) 82
Sending mail to a specific server example – locking the message in the spooler queue folder for processing
 [overview](#) 82
 [ROP request buffer](#) 82
 [ROP response buffer](#) 83
Sending mail to a specific server example – marking the message as ready for post-send server processing
 [overview](#) 87
 [ROP request buffer](#) 87
 [ROP response buffer](#) 87
[Sending mail to a specific server example - overview](#) 81
Sending mail to a specific server example – sending the message
 [overview](#) 84
 [ROP request buffer](#) 84
 [ROP response buffer](#) 84
[Sending mail to a specific server example - submitting the message to the spooler queue folder](#) 82
[Sending messages overview](#) 13
Sequencing rules - client
 [sending a RopAbortSubmit ROP request](#) 64
 [sending a RopLockSpoolerMessage ROP request](#) 65
 [sending a RopSubmitMessage ROP request](#) 64
 [sending a RopTransportNewMail ROP request](#) 65
Sequencing rules - server
 [receiving a RopAbortSubmit ROP request](#) 70
 [receiving a RopGetAddressTypes ROP request](#) 70
 [receiving a RopGetTransportFolder ROP request](#) 71
 [receiving a RopOptionsData ROP request](#) 72
 [receiving a RopSetSpooler ROP request](#) 71
 [receiving a RopSpoolerLockMessage ROP request](#) 71
 [receiving a RopSubmitMessage ROP request](#) 67
 [receiving a RopTransportNewMail ROP request](#) 72
 [receiving a RopTransportSend ROP request](#) 72
Server
 [abstract data model](#) 65
 [higher-layer triggered events](#) 66
 [initialization](#) 66
 [other local events](#) 72
 [timer events](#) 72
 [timers](#) 66
Server - abstract data model

[Global ADM data type](#) 65
[per E-mail object](#) 66
[per mailbox](#) 65
[per Message object](#) 66
[User ADM data type](#) 66

Server - sequencing rules
[receiving a RopAbortSubmit ROP request](#) 70
[receiving a RopGetAddressTypes ROP request](#) 70
[receiving a RopGetTransportFolder ROP request](#) 71
[receiving a RopOptionsData ROP request](#) 72
[receiving a RopSetSpooler ROP request](#) 71
[receiving a RopSpoolerLockMessage ROP request](#) 71
[receiving a RopSubmitMessage ROP request](#) 67
[receiving a RopTransportNewMail ROP request](#) 72
[receiving a RopTransportSend ROP request](#) 72

Spooler and transport ROPs
[RopGetTransportFolder ROP](#) 55
[RopSetSpooler ROP](#) 55
[RopSpoolerLockMessage ROP](#) 56
[RopTransportNewMail ROP](#) 57
[RopTransportSend ROP](#) 56

[Spooler and Transport ROPs message](#) 55
[Standards assignments](#) 14

Submitting a deferred message example
[overview](#) 74
[ROP request buffer](#) 74
[ROP response buffer](#) 74

Submitting a message example
[overview](#) 73
[ROP request buffer](#) 73
[ROP response buffer](#) 73

T

Timer events
[client](#) 65
[server](#) 72

Timers
[client](#) 61
[server](#) 66

[Tracking changes](#) 91
[Transport](#) 15

Triggered events - client
[client-to-client interop: voting](#) 62
[controlling the sending of mail](#) 63
[deferring sending a message](#) 62
[delivering mail to the server](#) 64
[optimizing send](#) 62
[processing a mail in the spooler queue](#) 64
[resending a message](#) 62
[sending a message](#) 61
[sending a message with expiry time](#) 62

Triggered events - higher-layer
[server](#) 66

V

[Vendor-extensible fields](#) 14
[Versioning](#) 14
[Voting and tracking overview](#) 13