Email Object Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation (“this documentation”) for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **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 can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft Open Specifications Promise or the Microsoft Community Promise. If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **License Programs.** To see all of the protocols in scope under a specific license program and the associated patents, visit the Patent Map.
- **Trademarks.** The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names.** The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are 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 as specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications documentation does 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 documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

Support. For questions and support, please contact dochelp@microsoft.com.
### Revision Summary

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision History</th>
<th>Revision Class</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>4/4/2008</td>
<td>0.1</td>
<td>New</td>
<td>Initial Availability.</td>
</tr>
<tr>
<td>4/25/2008</td>
<td>0.2</td>
<td>Minor</td>
<td>Revised and updated property names and other technical content.</td>
</tr>
<tr>
<td>6/27/2008</td>
<td>1.0</td>
<td>Major</td>
<td>Initial Release.</td>
</tr>
<tr>
<td>8/6/2008</td>
<td>1.01</td>
<td>Minor</td>
<td>Updated references to reflect date of initial release.</td>
</tr>
<tr>
<td>9/3/2008</td>
<td>1.02</td>
<td>Minor</td>
<td>Revised and edited technical content.</td>
</tr>
<tr>
<td>10/1/2008</td>
<td>1.03</td>
<td>Minor</td>
<td>Revised and edited technical content.</td>
</tr>
<tr>
<td>12/3/2008</td>
<td>1.04</td>
<td>Minor</td>
<td>Revised and edited technical content.</td>
</tr>
<tr>
<td>4/10/2009</td>
<td>2.0</td>
<td>Major</td>
<td>Updated technical content and applicable product releases.</td>
</tr>
<tr>
<td>7/15/2009</td>
<td>3.0</td>
<td>Major</td>
<td>Revised and edited for technical content.</td>
</tr>
<tr>
<td>11/4/2009</td>
<td>4.0.0</td>
<td>Major</td>
<td>Updated and revised the technical content.</td>
</tr>
<tr>
<td>2/10/2010</td>
<td>5.0.0</td>
<td>Major</td>
<td>Updated and revised the technical content.</td>
</tr>
<tr>
<td>5/5/2010</td>
<td>6.0.0</td>
<td>Major</td>
<td>Updated and revised the technical content.</td>
</tr>
<tr>
<td>8/4/2010</td>
<td>7.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>11/3/2010</td>
<td>7.1</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>3/18/2011</td>
<td>7.1</td>
<td>None</td>
<td>No changes to the meaning, language, and formatting of the technical content.</td>
</tr>
<tr>
<td>8/5/2011</td>
<td>8.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>10/7/2011</td>
<td>9.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>1/20/2012</td>
<td>10.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>4/27/2012</td>
<td>11.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>7/16/2012</td>
<td>11.1</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>10/8/2012</td>
<td>12.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>2/11/2013</td>
<td>13.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>7/26/2013</td>
<td>14.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>11/18/2013</td>
<td>15.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>2/10/2014</td>
<td>15.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>4/30/2014</td>
<td>16.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>7/31/2014</td>
<td>16.1</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>10/30/2014</td>
<td>16.2</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>3/16/2015</td>
<td>17.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>Date</td>
<td>Revision History</td>
<td>Revision Class</td>
<td>Comments</td>
</tr>
<tr>
<td>------------</td>
<td>------------------</td>
<td>----------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>5/26/2015</td>
<td>18.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>9/14/2015</td>
<td>18.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>6/13/2016</td>
<td>18.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>9/14/2016</td>
<td>18.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>10/17/2016</td>
<td>18.1</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>7/24/2018</td>
<td>19.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>10/1/2018</td>
<td>20.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>4/22/2021</td>
<td>21.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>8/17/2021</td>
<td>22.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
</tbody>
</table>
## Table of Contents

1 Introduction ........................................................................................................................................ 10
  1.1 Glossary ........................................................................................................................................ 10
  1.2 References .................................................................................................................................... 14
    1.2.1 Normative References ................................................................................................................ 15
    1.2.2 Informative References .............................................................................................................. 16
  1.3 Overview ........................................................................................................................................ 16
    1.3.1 E-Mail Objects ............................................................................................................................ 16
      1.3.1.1 Creating, Opening, and Saving E-Mail Objects ..................................................................... 16
      1.3.1.2 Sending Messages ............................................................................................................... 16
      1.3.1.3 Replying and Forwarding Messages .................................................................................... 16
    1.3.2 Report Messages ...................................................................................................................... 17
      1.3.2.1 Read Receipt ........................................................................................................................ 17
      1.3.2.2 Non-Read Receipt .............................................................................................................. 17
      1.3.2.3 Delivery Receipt .................................................................................................................. 17
      1.3.2.4 Non-Delivery Report ........................................................................................................... 17
    1.3.3 Voting and Tracking .................................................................................................................. 17
      1.3.4 Controlling Sending and Delivery of Mail .............................................................................. 17
  1.4 Relationship to Other Protocols ...................................................................................................... 18
  1.5 Prerequisites/Preconditions ............................................................................................................ 18
  1.6 Applicability Statement .................................................................................................................. 18
  1.7 Versioning and Capability Negotiation ............................................................................................. 18
  1.8 Vendor-Extensible Fields ............................................................................................................... 18
  1.9 Standards Assignments .................................................................................................................. 18

2 Messages ............................................................................................................................................ 19
  2.1 Transport ....................................................................................................................................... 19
  2.2 Message Syntax ............................................................................................................................. 19
    2.2.1 E-Mail Object Properties .......................................................................................................... 23
      2.2.1.1 PidTagBlockStatus Property ............................................................................................... 23
      2.2.1.2 PidTagConversationId Property .......................................................................................... 24
      2.2.1.3 PidTagConversationIndex Property ..................................................................................... 24
      2.2.1.4 PidTagConversationIndexTracking Property ...................................................................... 26
      2.2.1.5 PidTagConversationTopic Property ..................................................................................... 26
      2.2.1.6 PidTagDeferredDeliveryTime Property .............................................................................. 26
      2.2.1.7 PidTagDisplayBcc Property .................................................................................................. 26
      2.2.1.8 PidTagDisplayCc Property .................................................................................................. 26
      2.2.1.9 PidTagDisplayTo Property ................................................................................................... 27
      2.2.1.10 PidTagIconIndex Property .................................................................................................. 27
      2.2.1.11 PidTagInternetMailOverrideFormat Property ................................................................... 28
      2.2.1.12 PidTagInternetMessageId Property .................................................................................... 29
      2.2.1.13 PidTagInReplyToId Property .............................................................................................. 29
      2.2.1.14 PidTagLastVerbExecuted Property ..................................................................................... 29
      2.2.1.15 PidTagLastVerbExecutionTime Property .......................................................................... 31
      2.2.1.16 PidTagMessageClass Property .......................................................................................... 31
      2.2.1.17 PidTagMessageToMe Property ............................................................................................ 31
      2.2.1.18 PidTagMessageCcMe Property ........................................................................................... 32
      2.2.1.19 PidTagMessageRecipientMe Property ............................................................................... 32
      2.2.1.20 PidTagOriginatorDeliveryReportRequested Property ....................................................... 32
      2.2.1.21 PidTagOriginatorNonDeliveryReportRequested Property ............................................... 32
      2.2.1.22 PidTagOriginalSensitivity Property .................................................................................... 32
      2.2.1.23 PidTagReceivedRepresentingAddressType Property .......................................................... 33
      2.2.1.24 PidTagReceivedRepresentingEmailAddress Property ...................................................... 33
      2.2.1.25 PidTagReceivedRepresentingEntryId Property .................................................................. 33
      2.2.1.26 PidTagReceivedRepresentingName Property ...................................................................... 33
      2.2.1.27 PidTagReceivedRepresentingSearchKey Property ............................................................. 33
2.2.1.28 PidTagReceivedRepresentingSmtpAddress Property ........................................ 33
2.2.1.29 PidTagReadReceiptRequested Property ............................................................ 34
2.2.1.30 PidTagReadReceiptSmtpAddress Property .......................................................... 34
2.2.1.31 PidTagNonReceiptNotificationRequested Property ............................................... 34
2.2.1.32 PidTagOriginalAuthorEntryId Property ............................................................. 34
2.2.1.33 PidTagOriginalAuthorName Property ................................................................. 34
2.2.1.34 PidTagReportDisposition Property .................................................................... 34
2.2.1.35 PidTagReportDispositionMode Property ................................................................ 35
2.2.1.36 PidTagReceivedByAddressType Property ............................................................ 35
2.2.1.37 PidTagReceivedByEmailAddress Property ........................................................... 35
2.2.1.38 PidTagReceivedByEntryId Property .................................................................... 35
2.2.1.39 PidTagReceivedByName Property ....................................................................... 35
2.2.1.40 PidTagReceivedBySearchKey Property ................................................................ 36
2.2.1.41 PidTagReceivedBySmtpAddress Property ............................................................. 36
2.2.1.42 PidTagRecipientReassignmentProhibited Property .............................................. 36
2.2.1.43 PidTagReplyRecipientEntries Property ................................................................. 36
2.2.1.44 PidTagReplyRecipientNames Property ................................................................ 36
2.2.1.45 PidTagReplyRequested Property ........................................................................ 37
2.2.1.46 PidTagResponseRequested Property .................................................................... 37
2.2.1.47 PidTagSendRichInfo Property .............................................................................. 37
2.2.1.48 PidTagSenderAddressType Property .................................................................... 37
2.2.1.49 PidTagSenderEmailAddress Property ................................................................... 37
2.2.1.50 PidTagSenderEntryId Property ............................................................................ 37
2.2.1.51 PidTagSenderName Property ................................................................................. 38
2.2.1.52 PidTagSenderSearchKey Property ....................................................................... 38
2.2.1.53 PidTagSenderSmtpAddress Property ...................................................................... 38
2.2.1.54 PidTagSentRepresentingAddressType Property .................................................. 38
2.2.1.55 PidTagSentRepresentingEmailAddress Property ................................................ 38
2.2.1.56 PidTagSentRepresentingEntryId Property ........................................................... 38
2.2.1.57 PidTagSentRepresentingName Property ................................................................ 39
2.2.1.58 PidTagSentRepresentingSearchKey Property ....................................................... 39
2.2.1.59 PidTagSentRepresentingSmtpAddress Property .................................................... 39
2.2.1.60 PidTagSubjectPrefix Property ............................................................................ 39
2.2.1.61 PidTagTransportMessageHeaders Property ........................................................ 39
2.2.1.62 PidLidInternetAccountName Property .................................................................. 40
2.2.1.63 PidLidInternetAccountStamp Property ............................................................... 40
2.2.1.64 PidTagPrimarySendAccount Property .................................................................. 40
2.2.1.65 PidTagNextSendAcct Property ............................................................................ 40
2.2.1.66 PidLidUseTnef Property ....................................................................................... 40
2.2.1.67 Attachments ........................................................................................................ 40
2.2.1.68 Categories and Keywords .................................................................................... 40
2.2.1.69 Contacts .............................................................................................................. 41
2.2.1.70 Flags ..................................................................................................................... 41
2.2.1.71 Reminders ............................................................................................................ 41
2.2.1.72 Recipients ............................................................................................................. 41
2.2.1.73 PidLidAutoProcessState Property ....................................................................... 41
2.2.1.74 PidLidVerbStream Property .................................................................................... 41
2.2.1.74.1 VoteOption Structure ...................................................................................... 42
2.2.1.74.2 VoteOptionExtras Structure ............................................................................ 44
2.2.1.75 PidLidVerbResponse Property .............................................................................. 44
2.2.1.76 PidTagTargetEntryId Property ............................................................................. 45
2.2.1.77 PidTagAutoResponseSuppress Property ............................................................. 45
2.2.1.78 PidTagMessageEditorFormat Property ................................................................. 45
2.2.1.79 PidTagMessageSubmissionId Property ................................................................ 46
2.2.1.80 PidTagSenderIdStatus Property ......................................................................... 46
2.2.1.81 PidTagListHelp Property ...................................................................................... 46
2.2.1.82 PidTagListSubscribe Property ............................................................................. 47
2.2.1.83 PidTagListUnsubscribe Property ......................................................................... 47
2.2.1.84 PidTagDelegatedByRule Property ........................................ 47
2.2.1.85 PidTagOriginalMessageId Property ........................................ 47
2.2.1.86 PidTagOriginalMessageClass Property ..................................... 47
2.2.2 Message Status Reports Properties ............................................. 47
2.2.2.1 PidTagMessageClass Property .................................................. 47
2.2.2.2 PidTagOriginalDeliveryTime Property ...................................... 48
2.2.2.3 PidTagOriginalDisplayTo Property .......................................... 48
2.2.2.4 PidTagOriginalDisplayCc Property .......................................... 48
2.2.2.5 PidTagOriginalDisplayBcc Property ......................................... 48
2.2.2.6 PidTagOriginalSenderAddressType Property ............................... 49
2.2.2.7 PidTagOriginalSenderEmailAddress Property .............................. 49
2.2.2.8 PidTagOriginalSenderEntryId Property .................................... 49
2.2.2.9 PidTagOriginalSenderName Property ....................................... 49
2.2.2.10 PidTagOriginalSenderSearchKey Property ................................ 49
2.2.2.11 PidTagOriginalSentRepresentingAddressType Property ............... 49
2.2.2.12 PidTagOriginalSentRepresentingEmailAddress Property ............... 49
2.2.2.13 PidTagOriginalSentRepresentingEntryId Property ..................... 50
2.2.2.14 PidTagOriginalSentRepresentingName Property ........................ 50
2.2.2.15 PidTagOriginalSentRepresentingSearchKey Property .................. 50
2.2.2.16 PidTagOriginalSubject Property ........................................... 50
2.2.2.17 PidTagOriginalSubmitTime Property ....................................... 50
2.2.2.18 PidTagParentKey Property ..................................................... 50
2.2.2.19 PidTagReportEntryId Property .............................................. 51
2.2.2.20 PidTagReportName Property .................................................. 51
2.2.2.21 PidTagReportSearchKey Property ............................................ 51
2.2.2.22 PidTagReportTag Property .................................................... 51
2.2.2.23 PidTagReportText Property ................................................... 53
2.2.2.24 PidTagReadReceiptAddressType Property ................................ 53
2.2.2.25 PidTagReadReceiptEmailAddress Property ............................... 53
2.2.2.26 PidTagReadReceiptEntryId Property ...................................... 53
2.2.2.27 PidTagReadReceiptName Property ......................................... 54
2.2.2.28 PidTagReadReceiptSearchKey Property ................................... 54
2.2.2.29 PidTagDeliverTime Property .................................................. 54
2.2.2.30 PidTagNonDeliveryReportDiagCode Property ............................ 54
2.2.2.31 PidTagNonDeliveryReportReasonCode Property ......................... 54
2.2.2.32 PidTagNonDeliveryReportStatusCode Property ......................... 54
2.2.2.33 PidTagReceiptTime Property .................................................. 54
2.2.2.34 PidTagRemoteMessageTransferAgent Property .......................... 55
2.2.2.35 PidTagReportingMessageTransferAgent Property ....................... 55
2.2.2.36 PidTagSupplementaryInfo Property ........................................ 55
2.2.3 E-Mail Submission Properties .................................................... 55
2.2.3.1 PidTagRecipientType Property ................................................ 55
2.2.3.2 PidTagDeferredSendNumber Property ..................................... 56
2.2.3.3 PidTagDeferredSendUnits Property ......................................... 56
2.2.3.4 PidTagDeferredSendTime Property .......................................... 56
2.2.3.5 PidTagExpiryNumber Property ................................................ 57
2.2.3.6 PidTagExpiryUnits Property ................................................... 57
2.2.3.7 PidTagExpiryTime Property ..................................................... 57
2.2.3.8 PidTagDeleteAfterSubmit Property ......................................... 57
2.2.3.9 PidTagMessageDeliveryTime Property ..................................... 58
2.2.3.10 PidTagSentMailSrvEID Property .......................................... 58
2.2.3.11 PidTagClientSubmitTime Property ......................................... 58
2.2.4 Message Delivery ROPs ............................................................. 58
2.2.4.1 RopSubmitMessage ROP ......................................................... 58
2.2.4.1.1 RopSubmitMessage ROP Request Buffer ................................ 58
2.2.4.1.2 RopSubmitMessage ROP Response Buffer .............................. 59
2.2.4.2 RopAbortSubmit ROP ............................................................. 59
2.2.4.2.1 RopAbortSubmit ROP Request Buffer ................................... 59
2.2.4.2.2  RopAbortSubmit ROP Response Buffer ........................................ 59
2.2.4.3  RopGetAddressTypes ROP ................................................................. 59
2.2.4.3.1  RopGetAddressTypes ROP Request Buffer ........................................ 59
2.2.4.3.2  RopGetAddressTypes ROP Response Buffer ...................................... 60
2.2.4.4  RopOptionsData ROP ............................................................................. 60
2.2.4.4.1  RopOptionsData ROP Request Buffer .................................................. 60
2.2.4.4.2  RopOptionsData ROP Response Buffer ................................................ 60
2.2.5  Spooler and Transport ROPs ................................................................. 61
2.2.5.1  RopSetSpooler ROP ............................................................................... 61
2.2.5.1.1  RopSetSpooler ROP Request Buffer ..................................................... 61
2.2.5.1.2  RopSetSpooler ROP Response Buffer .................................................... 61
2.2.5.2  RopGetTransportFolder ROP ................................................................. 61
2.2.5.2.1  RopGetTransportFolder ROP Request Buffer ....................................... 61
2.2.5.2.2  RopGetTransportFolder ROP Response Buffer ..................................... 61
2.2.5.3  RopSpoolerLockMessage ROP .............................................................. 62
2.2.5.3.1  RopSpoolerLockMessage ROP Request Buffer ................................... 62
2.2.5.3.2  RopSpoolerLockMessage ROP Response Buffer ................................... 62
2.2.5.4  RopTransportSend ROP ......................................................................... 62
2.2.5.4.1  RopTransportSend ROP Request Buffer .............................................. 62
2.2.5.4.2  RopTransportSend ROP Response Buffer ............................................ 62
2.2.5.5  RopTransportNewMail ROP .................................................................. 63
2.2.5.5.1  RopTransportNewMail ROP Request Buffer ....................................... 63
2.2.5.5.2  RopTransportNewMail ROP Response Buffer ...................................... 63

3  Protocol Details ........................................................................................... 64
3.1  Common Details ......................................................................................... 64
3.1.1  Abstract Data Model ............................................................................... 64
3.1.1.1  Per Global ............................................................................................. 64
3.1.1.2  Per Mailbox .......................................................................................... 64
3.1.1.3  Per Message Object ............................................................................. 64
3.1.1.4  Per Send State ...................................................................................... 65
3.1.2  Timers ...................................................................................................... 65
3.1.3  Initialization ............................................................................................ 65
3.1.4  Higher-Layer Triggered Events .............................................................. 65
3.1.5  Message Processing Events and Sequencing Rules ......................... 65
3.1.6  Timer Events .......................................................................................... 65
3.1.7  Other Local Events ............................................................................... 65
3.2  Client Details ............................................................................................ 65
3.2.1  Abstract Data Model ............................................................................... 65
3.2.2  Timers ...................................................................................................... 66
3.2.3  Initialization ............................................................................................ 66
3.2.4  Higher-Layer Triggered Events .............................................................. 66
3.2.4.1  Sending a Message ................................................................................. 66
3.2.4.1.1  Sending the Message as a Represented Sender ..................................... 66
3.2.4.1.2  Sending the Message as the Actual Sender ......................................... 66
3.2.4.1.3  Sending the Message as the Sender Itself .......................................... 66
3.2.4.1.4  Sending the Message on Behalf of Another Person .............................. 66
3.2.4.2  Deferring Message Send ...................................................................... 66
3.2.4.3  Sending a Message with Expiry Time ................................................. 66
3.2.4.4  Optimizing Send .................................................................................. 67
3.2.4.5  Resending a Message .......................................................................... 67
3.2.4.6  Soliciting Votes from Voters ............................................................... 67
3.2.4.6.1  Associating Options with a Voting Message ...................................... 67
3.2.4.6.2  Interpreting a Voting Message ........................................................... 67
3.2.4.6.3  Crafting a Voting Response Message ............................................... 68
3.2.4.6.4  Aggregating Voting Responses .......................................................... 68
3.2.4.7  Sending Mail Through a Specific Server ............................................ 68
3.2.4.8  Processing E-mail Objects in the Spooler Queue................................. 68
3.2.4.9 Delivering Mail to the Server .................................................. 68
3.2.4.10 Sending Read Receipts and Non-Read Receipts ...................... 69
3.2.5 Message Processing Events and Sequencing Rules ....................... 70
  3.2.5.1 Sending a RopSubmitMessage ROP Request ........................... 70
  3.2.5.2 Sending a RopAbortSubmit ROP Request ............................... 70
  3.2.5.3 Sending a RopSpoolerLockMessage ROP Request ..................... 70
  3.2.5.4 Sending a RopTransportNewMail ROP Request ........................ 70
3.2.6 Timer Events ........................................................................ 70
3.2.7 Other Local Events .................................................................. 70
3.3 Server Details ........................................................................... 70
  3.3.1 Abstract Data Model ............................................................... 70
    3.3.1.1 Per Message Object .......................................................... 71
    3.3.1.2 Per E-mail Object ............................................................... 71
    3.3.1.3 Per User ......................................................................... 71
  3.3.2 Timers .................................................................................. 71
  3.3.3 Initialization ......................................................................... 72
  3.3.4 Higher-Layer Triggered Events ............................................... 72
  3.3.5 Message Processing Events and Sequencing Rules .................... 72
    3.3.5.1 Receiving a RopSubmitMessage ROP Request ....................... 72
      3.3.5.1.1 Permission Check ....................................................... 72
      3.3.5.1.2 Delivering Mail on a RopSubmitMessage or RopTransportSend Request ........................................................................... 73
      3.3.5.1.3 Properties Read and/or Set Upon Submission .................... 73
          3.3.5.1.3.1 Represented Sender Properties ................................. 74
          3.3.5.1.3.2 Actual Sender Properties ...................................... 74
          3.3.5.1.3.3 Deferred Properties .............................................. 74
          3.3.5.1.3.4 Expiry Properties .................................................. 74
      3.3.5.1.4 Rule Processing ........................................................... 75
    3.3.5.2 Receiving a RopAbortSubmit ROP Request ............................ 75
    3.3.5.3 Receiving a RopGetAddressTypes ROP Request ..................... 75
    3.3.5.4 Receiving a RopGetSpooler ROP Request .............................. 76
    3.3.5.5 Receiving a RopGetTransportFolder ROP Request .................. 76
    3.3.5.6 Receiving a RopSpoolerLockMessage ROP Request ................. 76
    3.3.5.7 Receiving a RopTransportSend ROP Request .......................... 77
    3.3.5.8 Receiving a RopTransportNewMail ROP Request .................... 77
    3.3.5.9 Receiving a RopOptionsData ROP Request ............................. 77
  3.3.6 Timer Events ........................................................................ 77
  3.3.7 Other Local Events .................................................................. 77
4 Protocol Examples ......................................................................... 78
  4.1 Submitting a Message ................................................................. 78
    4.1.1 ROP Request Buffer ............................................................ 78
    4.1.2 ROP Response Buffer .......................................................... 78
  4.2 Submitting a Deferred Message .................................................. 79
    4.2.1 ROP Request Buffer ............................................................ 79
    4.2.2 ROP Response Buffer .......................................................... 79
  4.3 Aborting a Message Submission .................................................. 80
    4.3.1 ROP Request Buffer ............................................................ 80
    4.3.2 ROP Response Buffer .......................................................... 80
  4.4 Sending an E-Mail Message from a Messaging User to Another Messaging User .......................... 81
  4.5 Sending a Message with Voting Options ....................................... 84
  4.6 Sending Mail to a Specific Server ................................................ 86
    4.6.1 Initialization ....................................................................... 86
      4.6.1.1 ROP Request Buffer ....................................................... 86
      4.6.1.2 ROP Response Buffer ...................................................... 86
    4.6.2 Submitting the Message to the Spooler Queue Folder ................. 86
    4.6.3 Locking the Message in the Spooler Queue Folder for Processing ... 87
      4.6.3.1 ROP Request Buffer ....................................................... 87
      4.6.3.2 ROP Response Buffer ...................................................... 87

[MS-OXOMSG] - v20210817
Email Object Protocol
Copyright © 2021 Microsoft Corporation
Release: August 17, 2021
4.6.4 Determining the Transport Folder ................................................................. 87
4.6.4.1 ROP Request Buffer ................................................................................. 87
4.6.4.2 ROP Response Buffer .............................................................................. 88
4.6.5 Sending the Message .................................................................................... 88
4.6.5.1 ROP Request Buffer ................................................................................. 88
4.6.5.2 ROP Response Buffer .............................................................................. 88
4.6.6 Marking the Message as Ready for Post-Send Server Processing .................. 91
4.6.6.1 ROP Request Buffer ................................................................................. 91
4.6.6.2 ROP Response Buffer .............................................................................. 91

5 Security .................................................................................................................. 92
5.1 Security Considerations for Implementers ........................................................ 92
5.2 Index of Security Parameters .............................................................................. 92

6 Appendix A: Product Behavior ............................................................................. 93

7 Change Tracking .................................................................................................... 95

8 Index ....................................................................................................................... 96
1 Introduction

The Email Object Protocol enables the creation, transmission and storage of e-mail messages by representing e-mails as Message objects. The Email 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-OXCMSG].

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

address book: A collection of Address Book objects, each of which are contained in any number of address lists.

address type: An identifier for the type of email address, such as SMTP and EX.

American National Standards Institute (ANSI) character set: A character set defined by a code page approved by the American National Standards Institute (ANSI). The term "ANSI" as used to signify Windows code pages is a historical reference and a misnomer that persists in the Windows community. The source of this misnomer stems from the fact that the Windows code page 1252 was originally based on an ANSI draft, which became International Organization for Standardization (ISO) Standard 8859-1 [ISO/IEC-8859-1]. In Windows, the ANSI character set can be any of the following code pages: 1252, 1250, 1251, 1253, 1254, 1255, 1256, 1257, 1258, 874, 932, 936, 949, or 950. For example, "ANSI application" is usually a reference to a non-Unicode or code-page-based application. Therefore, "ANSI character set" is often misused to refer to one of the character sets defined by a Windows code page that can be used as an active system code page; for example, character sets defined by code page 1252 or character sets defined by code page 950. Windows is now based on Unicode, so the use of ANSI character sets is strongly discouraged unless they are used to interoperate with legacy applications or legacy data.

ASCII: The American Standard Code for Information Interchange (ASCII) is an 8-bit character-encoding scheme based on the English alphabet. ASCII codes represent text in computers, communications equipment, and other devices that work with text. ASCII refers to a single 8-bit ASCII character or an array of 8-bit ASCII characters with the high bit of each character set to zero.

Attachment object: A set of properties that represents a file, Message object, or structured storage that is attached to a Message object and is visible through the attachments table for a Message object.

big-endian: Multiple-byte values that are byte-ordered with the most significant byte stored in the memory location with the lowest address.

blind carbon copy (Bcc) recipient: An addressee on a Message object that is not visible to recipients of the Message object.

body part: A part of an Internet message, as described in [RFC2045].

carbon copy (Cc) recipient: An address on a Message object that is visible to recipients of the Message object but is not necessarily expected to take any action.

conversation thread: A series of messages and responses to those messages, typically related by subject.
Coordinated Universal Time (UTC): A high-precision atomic time standard that approximately tracks Universal Time (UT). It is the basis for legal, civil time all over the Earth. Time zones around the world are expressed as positive and negative offsets from UTC. In this role, it is also referred to as Zulu time (Z) and Greenwich Mean Time (GMT). In these specifications, all references to UTC refer to the time at UTC-0 (or GMT).

delegate rule: A server-side rule that is used to send mail to delegates on behalf of a delegator.

delivery receipt: A report message that is generated and sent by a client or server to the sender of a message or another designated recipient when an email message is received by an intended recipient.

display name: A text string that is used to identify a principal or other object in the user interface. Also referred to as title.

distribution list: A collection of users, computers, contacts, or other groups that is used only for email distribution, and addressed as a single recipient.

Domain Name System (DNS): A hierarchical, distributed database that contains mappings of domain names to various types of data, such as IP addresses. DNS enables the location of computers and services by user-friendly names, and it also enables the discovery of other information stored in the database.

Email object: A Message object that represents an email message in a message store and adheres to the property descriptions that are described in [MS-OXOMSG].

Embedded Message object: A Message object that is stored as an Attachment object within another Message object.

entry ID: See EntryID.

EntryID: A sequence of bytes that is used to identify and access an object.

flags: A set of values used to configure or report options or settings.

folder associated information (FAI): A collection of Message objects that are stored in a Folder object and are typically hidden from view by email applications. An FAI Message object is used to store a variety of settings and auxiliary data, including forms, views, calendar options, favorites, and category lists.

Folder object: A messaging construct that is typically used to organize data into a hierarchy of objects containing Message objects and folder associated information (FAI) Message objects.

globally unique identifier (GUID): A term used interchangeably with universally unique identifier (UUID) in Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the value. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] must be used for generating the GUID. See also universally unique identifier (UUID).

handle: Any token that can be used to identify and access an object such as a device, file, or a window.

header: A name-value pair that supplies structured data in an Internet email message or MIME entity.

Help file: A file that contains the documentation for a specific product or technology.

Hypertext Markup Language (HTML): An application of the Standard Generalized Markup Language (SGML) that uses tags to mark elements in a document, as described in [HTML].
Hypertext Transfer Protocol (HTTP): An application-level protocol for distributed, collaborative, hypermedia information systems (text, graphic images, sound, video, and other multimedia files) on the World Wide Web.

Internet Message Access Protocol - Version 4 (IMAP4): A protocol that is used for accessing email and news items from mail servers, as described in [RFC3501].

Inter-Personal Mail (IPM): Typical user messaging items, such as email and calendar items.

Language code identifier (LCID): A 32-bit number that identifies the user interface human language dialect or variation that is supported by an application or a client computer.

Little-endian: Multiple-byte values that are byte-ordered with the least significant byte stored in the memory location with the lowest address.

Locale: A collection of rules and data that are specific to a language and a geographical area. A locale can include information about sorting rules, date and time formatting, numeric and monetary conventions, and character classification.

Logon object: A Server object that provides access to a private mailbox or a public folder. A client obtains a Logon object by issuing a RopLogon remote operation (ROP) to a server.

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.

Mailbox: A message store that contains email, calendar items, and other Message objects for a single recipient.

Message body: The main message text of an email message. A few properties of a Message object represent its message body, with one property containing the text itself and others defining its code page and its relationship to alternative body formats.

Message class: A property that loosely defines the type of a message, contact, or other Personal Information Manager (PIM) object in a mailbox.

Message object: A set of properties that represents an email message, appointment, contact, or other type of personal-information-management object. In addition to its own properties, a Message object contains recipient properties that represent the addressees to which it is addressed, and an attachments table that represents any files and other Message objects that are attached to it.

Message store: A unit of containment for a single hierarchy of Folder objects, such as a mailbox or public folders.

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 object: An object that exists in a mailbox. It can be only a Folder object or a Message object.

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

Multipurpose Internet Mail Extensions (MIME): A set of extensions that redefines and expands support for various types of content in email messages, as described in [RFC2045], [RFC2046], and [RFC2047].

Non-delivery report: A report message that is generated and sent by a server to the sender of a message if an email message could not be received by an intended recipient.
**non-read receipt**: A message that is generated when an email message is deleted at the expiration of a time limit or due to other client-specific criteria.

**Object Linking and Embedding (OLE)**: A technology for transferring and sharing information between applications by inserting a file or part of a file into a compound document. The inserted file can be either embedded or linked. See also embedded object and linked object.

**Out of Office (OOF)**: One of the possible values for the free/busy status on an appointment. It indicates that the user will not be in the office during the appointment.

**permission**: A rule that is associated with an object and that regulates which users can gain access to the object and in what manner. See also rights.

**plain text**: Text that does not have markup. See also plain text message body.

**Post Office Protocol - Version 3 (POP3)**: A protocol that is used for accessing email from mail servers, as described in [RFC1939].

**primary recipient**: A person for whom a message is directly intended.

**read receipt**: An email message that is sent to the sender of a message to indicate that a message recipient received the message.

**recipient**: (1) An entity that can receive email messages.

(2) An entity that is in an address list, can receive email messages, and contains a set of attributes. Each attribute has a set of associated values.

**recipient table**: The part of a **Message object** that represents users to whom a message is addressed. Each row of the table is a set of properties that represents one **recipient** (2).

**reminder**: A generally user-visible notification that a specified time has been reached. A reminder is most commonly related to the beginning of a meeting or the due time of a task but it can be applied to any object type.

**remote operation (ROP)**: An operation that is invoked against a server. Each ROP represents an action, such as delete, send, or query. A ROP is contained in a ROP buffer for transmission over the wire.

**remote procedure call (RPC)**: A communication protocol used primarily between client and server. The term has three definitions that are often used interchangeably: a runtime environment providing for communication facilities between computers (the RPC runtime); a set of request-and-response message exchanges between computers (the RPC exchange); and the single message from an RPC exchange (the RPC message). For more information, see [C706].

**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).

**Rich Text Format (RTF)**: Text with formatting as described in [MSFT-RTF].

**ROP request**: See **ROP request buffer**.

**ROP request buffer**: A ROP buffer that a client sends to a server to be processed.

**ROP response**: See **ROP response buffer**.

**ROP response buffer**: A ROP buffer that a server sends to a client to be processed.
search folder: A **Folder object** that provides a means of querying for items that match certain criteria. The search folder includes the search folder definition message and the search folder container.

search key: A binary-comparable key that identifies related objects for a search.

Sent Items folder: A special folder that is the default location for storing copies of **Message objects** after they are submitted or sent.

server object: A class of object in the configuration naming context (config NC). A server object can have an nTDSDSA object as a child.

Simple Mail Transfer Protocol (SMTP): A member of the TCP/IP suite of protocols that is used to transport Internet messages, as described in [RFC5321].

spam: An unsolicited email message.

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

subobject: For a folder, the messages and subfolders that are contained in that folder. For a message, the recipients (2) and attachments to that message. For an attachment, the **Embedded Message object** for that attachment.

To recipient: See **primary recipient**.

Transport Neutral Encapsulation Format (TNEF): A binary type-length-value encoding that is used to encode properties for transport, as described in [MS-OXTNEF].

Unicode: A character encoding standard developed by the Unicode Consortium that represents almost all of the written languages of the world. The Unicode standard [UNICODE5.0.0/2007] provides three forms (UTF-8, UTF-16, and UTF-32) and seven schemes (UTF-8, UTF-16, UTF-16 BE, UTF-16 LE, UTF-32, UTF-32 LE, and UTF-32 BE).

Uniform Resource Identifier (URI): A string that identifies a resource. The URI is an addressing mechanism defined in Internet Engineering Task Force (IETF) Uniform Resource Identifier (URI): Generic Syntax [RFC3986].

UTF-16: A standard for encoding Unicode characters, defined in the Unicode standard, in which the most commonly used characters are defined as double-byte characters. Unless specified otherwise, this term refers to the UTF-16 encoding form specified in [UNICODE5.0.0/2007] section 3.9.

UUEncoded attachment: A file that is attached to an email 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 defined in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

### 1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the [Errata](#).
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.

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

[MS-OXCBODY] Microsoft Corporation, "Best Body Retrieval Algorithm".

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

[MS-OXCFXICS] Microsoft Corporation, "Bulk Data Transfer Protocol".

[MS-OXCMEMAIL] Microsoft Corporation, "RFC 2822 and MIME to Email Object Conversion Algorithm".

[MS-OXCMAPIMHTTP] Microsoft Corporation, "Messaging Application Programming Interface (MAPI) Extensions for HTTP".


[MS-OXCNOTIF] Microsoft Corporation, "Core Notifications Protocol".


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


[MS-OXCSMP] Microsoft Corporation, "Spam Confidence Level Protocol".


[MS-OXOCAL] Microsoft Corporation, "Appointment and Meeting Object Protocol".

[MS-OXOCFG] Microsoft Corporation, "Configuration Information Protocol".

[MS-OXOCNTC] Microsoft Corporation, "Contact Object Protocol".

[MS-OXODLGT] Microsoft Corporation, "Delegate Access Configuration Protocol".

[MS-OXOFLAG] Microsoft Corporation, "Informational Flagging Protocol".

[MS-OXORMDR] Microsoft Corporation, "Reminder Settings Protocol".

[MS-OXORULE] Microsoft Corporation, "Email Rules Protocol".


1.2.2 Informative References


[MS-OXCFOLD] Microsoft Corporation, "Folder Object Protocol".

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


[MS-OXTNEF] Microsoft Corporation, "Transport Neutral Encapsulation Format (TNEF) Data Algorithm".

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

An E-mail object is created, opened, and saved in the same way that any Message object is created, opened, and saved, as described 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

Repying 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-OXCFOLD], 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 2.2.1.74. 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 2.2.1.75.

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 Email Object Protocol has the same dependencies as the Message and Attachment Object Protocol, as described in [MS-OXCMMSG].

For conceptual background information and overviews of the relationships and interactions between this and other protocols, see [MS-OXPROTO].

1.5 Prerequisites/Preconditions

The Email Object Protocol has the same prerequisites and preconditions as the Message and Attachment Object Protocol, as described [MS-OXCMMSG].

1.6 Applicability Statement

The Email 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 **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 Email 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 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 2.2.1.36)
- **PidTagReceivedByEmailAddress** (section 2.2.1.37)
- **PidTagReceivedByEntryId** (section 2.2.1.38)
- **PidTagReceivedByName** (section 2.2.1.39)
- **PidTagReceivedBySearchKey** (section 2.2.1.40)
- **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 2.2.1.54)
- **PidTagSentRepresentingEmailAddress** (section 2.2.1.55)
- **PidTagSentRepresentingEntryId** (section 2.2.1.56)
- **PidTagSentRepresentingName** (section 2.2.1.57)
- **PidTagSentRepresentingSearchKey** (section 2.2.1.58)
- **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 2.2.1.48)
- **PidTagSenderEmailAddress** (section 2.2.1.49)
- **PidTagSenderEntryId** (section 2.2.1.50)
- **PidTagSenderName** (section 2.2.1.51)
- **PidTagSenderSearchKey** (section 2.2.1.52)
- **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)
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 2.2.1.7)
- **PidTagOriginalDisplayTo** (section 2.2.3)

The following from properties are associated with Cc recipients:

- **PidTagDisplayCc** (section 2.2.1.8)
- **PidTagMessageCcMe** (section 2.2.1.8)
- **PidTagOriginalDisplayCc** (section 2.2.4)

The following from properties are associated with Bcc recipients:

- **PidTagDisplayBcc** (section 2.2.1.7)
- **PidTagOriginalDisplayBcc** (section 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 2.2.1.60)
- **PidTagOriginalSubject** (section 2.2.1.16)

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 2.2.1.56.1)
- **PidTagBodyHtml** ([MS-OXCMSG] section 2.2.1.56.3)
- **PidTagRtfCompressed** ([MS-OXCMSG] section 2.2.1.56.4)
- **PidTagRtfInSync** ([MS-OXCMSG] section 2.2.1.56.5)
- **PidTagMessageEditorFormat** (section 2.2.1.78)

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 2.2.8.8)
  - **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-OXCMSG] section 2.2.1.3)
  - PidTagReadReceiptRequested (section 2.2.1.29)
  - PidTagReadReceiptEntryId (section 2.2.2.26)
  - PidTagReadReceiptSearchKey (section 2.2.2.28)
  - PidTagOriginalSensitivity (section 2.2.1.22)
  - PidTagRecipientReassignmentProhibited (section 2.2.1.42)
  - PidTagReplyRequested (section 2.2.1.45)
  - PidTagResponseRequested (section 2.2.1.46)
  - PidTagReplyRecipientEntries (section 2.2.1.43)
  - PidTagReplyRecipientNames (section 2.2.1.44)
  - PidLidAutoProcessState (section 2.2.1.73)
  - PidLidVerbStream (section 2.2.1.74)
  - PidLidVerbResponse (section 2.2.1.75)

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-OXOABK] section 2.2.3.18)
- PidTagTransportMessageHeaders (section 2.2.1.61)
- 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-OXCMSG] 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 2.617) indicates whether the user's preference for viewing external content (such as links to images on a Hypertext Transfer Protocol (HTTP) server) is allowed or blocked in the message body.

A client can ignore this property and always allow or always block external content based on factors that are determined by the implementer. For example, a client can always allow or always block external content based on whether the sender is on a safe senders list. For information about the safe senders list, see the description of the PidTagJunkAddRecipientsToSafeSendersList property in [MS-OXCSPAM] section 2.2.2.1.

Valid values for this property are given in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000000</td>
<td>Default value. Block external content.</td>
</tr>
<tr>
<td>Variable nonzero</td>
<td>Allow or block external content, as described following this table.</td>
</tr>
</tbody>
</table>

If this property is used, the default (0x00000000) is to block the external content. However, if the value of this property falls within a specific range, as described in this section, 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.

\[
\text{result} = ((\text{floatdate} - \text{floor(floatdate)}) \times 100000000) + 3;
\]

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 (section [MS-OXPROPS] section 2.649) 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 0x01 (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, 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 (section [MS-OXPROPS] section 2.650) indicates the relative position of this message within a conversation thread. It is set according to the description in the following diagram.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 |
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

Conversation Index Header (22 bytes):

...
Reserved (8 bits): Set to 0x01.

Current FILETIME (40 bits): The time of delivery in UTC expressed as a PtypTime type ([MS-OXCDATA] section 2.11.1) is obtained, where the 32 bits of the high part and the 8 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.<2>

<table>
<thead>
<tr>
<th>40 bits</th>
<th>24 least significant bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Included</td>
<td>Excluded</td>
</tr>
</tbody>
</table>

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): A PtypGuid type ([MS-OXCDATA] section 2.11.1) that is 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.

DC (Delta code) (1 bit) and Time Delta (31 bits): Calculated based on TimeDiff, a 64-bit value representing the difference between the current time and the time stored in the conversation index header: <3>

- 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 TimeDiff value remaining after the 18 least significant bits are excluded. The following table depicts which portion of the TimeDiff value is included in the Time Delta field in this scenario.

<table>
<thead>
<tr>
<th>15 most significant bits</th>
<th>31 bits</th>
<th>18 least significant bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Excluded</td>
<td>Included</td>
<td>Excluded</td>
</tr>
</tbody>
</table>

- 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 TimeDiff value remaining after the 23 least significant bits are excluded. The following table depicts which portion of the TimeDiff value is included in the Time Delta field in this scenario.

<table>
<thead>
<tr>
<th>10 most significant bits</th>
<th>31 bits</th>
<th>23 least significant bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Excluded</td>
<td>Included</td>
<td>Excluded</td>
</tr>
</tbody>
</table>

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 2.651) is set to 0x01 (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 0x00 (FALSE).

### 2.2.1.5 PidTagConversationTopic Property

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

The PidTagConversationTopic property ([MS-OXPROPS] section 2.652) contains an unchanging copy of the original subject.<4> 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 2.662) contains the date and time, in UTC, at which the sender prefers the message to be delivered. This property MAY be included in the response. If the property 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.<5>

### 2.2.1.7 PidTagDisplayBcc Property

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

The PidTagDisplayBcc property ([MS-OXPROPS] section 2.674) 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 2.675) 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-OXCMMSG] 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 2.678) 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-OXCMMSG] 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 2.737) 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-OXCMMSG] section 2.2.1.3) or the `PidTagMessageFlags` property ([MS-OXCMMSG] section 2.2.1.6). Examples of values for the `PidTagIconIndex` property are shown in the following table.

<table>
<thead>
<tr>
<th>Mail item state</th>
<th>Mail item icon index</th>
</tr>
</thead>
<tbody>
<tr>
<td>New mail</td>
<td>0xFFFFFFFF</td>
</tr>
<tr>
<td>Read mail</td>
<td>0x00000100</td>
</tr>
<tr>
<td>Unread mail</td>
<td>0x00000101</td>
</tr>
<tr>
<td>Submitted mail</td>
<td>0x00000102</td>
</tr>
<tr>
<td>Unsent mail</td>
<td>0x00000103</td>
</tr>
<tr>
<td>Receipt mail</td>
<td>0x00000104</td>
</tr>
<tr>
<td>Replied mail</td>
<td>0x00000105</td>
</tr>
<tr>
<td>Forwarded mail</td>
<td>0x00000106</td>
</tr>
<tr>
<td>Remote mail</td>
<td>0x00000107</td>
</tr>
<tr>
<td>Delivery receipt</td>
<td>0x00000108</td>
</tr>
<tr>
<td>Read receipt</td>
<td>0x00000109</td>
</tr>
<tr>
<td><strong>Non-delivery report</strong></td>
<td>0x0000010A</td>
</tr>
<tr>
<td><strong>Non-read receipt</strong></td>
<td>0x0000010B</td>
</tr>
<tr>
<td>Recall_S mails</td>
<td>0x0000010C</td>
</tr>
<tr>
<td>Recall_F mail</td>
<td>0x0000010D</td>
</tr>
<tr>
<td>Tracking mail</td>
<td>0x0000010E</td>
</tr>
<tr>
<td>Out of Office mail</td>
<td>0x0000011B</td>
</tr>
</tbody>
</table>
### 2.2.1.11 PidTagInternetMailOverrideFormat Property

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

The `PidTagInternetMailOverrideFormat` property ([MS-OXPROPS] section 2.747) indicates the encoding method and [Hypertext Markup Language (HTML)](https://en.wikipedia.org/wiki/Hypertext_Markup_Language) 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.

#### Format 1 (3 bits)

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0</td>
<td>Default value. The mail system chooses the default encoding scheme, based on other fields in this property value.</td>
</tr>
<tr>
<td>0x1</td>
<td>The message is sent in MIME format with text/plain and text/HTML body parts.</td>
</tr>
<tr>
<td>0x2</td>
<td>The message is sent as plain text with UUEncoded attachments.</td>
</tr>
<tr>
<td>0x4</td>
<td>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.</td>
</tr>
</tbody>
</table>

#### E18 (2 bits)

Ignored if **Format 1** = 0 or **P2** = 0 or **M4** = 0. Otherwise, set to one of the following values to indicate the HTML inclusion.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0</td>
<td>Text/plain only.</td>
</tr>
<tr>
<td>0x1</td>
<td>Text/plain and text/HTML.</td>
</tr>
<tr>
<td>0x2</td>
<td>Text/plain and text/HTML. This value is treated the same as the 0x1 value.</td>
</tr>
</tbody>
</table>

#### M4 (1 bit)

Ignored if **Format 1** = 0 or **P2** = 0; otherwise, indicates the encoding, as shown in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Use the uuencode algorithm as described in [IEEE1003.1], and ignore the value of the E18 field.</td>
</tr>
<tr>
<td>1</td>
<td>Use MIME encoding, and use the value of the E18 field to determine body inclusions.</td>
</tr>
</tbody>
</table>
P2 (1 bit): Ignored if Format1 = 0; otherwise, indicates the preference, as shown in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Ignore the value of the M4 field.</td>
</tr>
<tr>
<td>1</td>
<td>Use the value of the M4 field to determine encoding.</td>
</tr>
</tbody>
</table>

2.2.1.12 PidTagInternetMessageId Property

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

The PidTagInternetMessageId property ([MS-OXPROPS] section 2.748) 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 2.742) 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 2.767) 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.

<table>
<thead>
<tr>
<th>Verb name</th>
<th>Alternate name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open</td>
<td>NOTEIVERB_OPEN</td>
<td>0</td>
</tr>
<tr>
<td>ReplyToSender</td>
<td>NOTEIVERB_REPLYTOSENDER</td>
<td>102</td>
</tr>
<tr>
<td>ReplyToAll</td>
<td>NOTEIVERB_REPLYTOALL</td>
<td>103</td>
</tr>
<tr>
<td>Forward</td>
<td>NOTEIVERB_FORWARD</td>
<td>104</td>
</tr>
<tr>
<td>Print</td>
<td>NOTEIVERB_PRINT</td>
<td>105</td>
</tr>
<tr>
<td>Save as</td>
<td>NOTEIVERB_SAVEAS</td>
<td>106</td>
</tr>
<tr>
<td>ReplyToFolder</td>
<td>NOTEIVERB_REPLYTOFOLDER</td>
<td>108</td>
</tr>
<tr>
<td>Save</td>
<td>NOTEIVERB_SAVE</td>
<td>500</td>
</tr>
<tr>
<td>Properties</td>
<td>NOTEIVERB_PROPERTIES</td>
<td>510</td>
</tr>
<tr>
<td>Followup</td>
<td>NOTEIVERB_FOLLOWUP</td>
<td>511</td>
</tr>
<tr>
<td>Verb name</td>
<td>Alternate name</td>
<td>Value</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>Accept</td>
<td>NOTEIVERB_ACCEPT</td>
<td>512</td>
</tr>
<tr>
<td>Tentative</td>
<td>NOTEIVERB_TENTATIVE</td>
<td>513</td>
</tr>
<tr>
<td>Reject</td>
<td>NOTEIVERB_REJECT</td>
<td>514</td>
</tr>
<tr>
<td>Decline</td>
<td>NOTEIVERB_DECLINE</td>
<td>515</td>
</tr>
<tr>
<td>Invite</td>
<td>NOTEIVERB_INVITE</td>
<td>516</td>
</tr>
<tr>
<td>Update</td>
<td>NOTEIVERB_UPDATE</td>
<td>517</td>
</tr>
<tr>
<td>Cancel</td>
<td>NOTEIVERB_CANCEL</td>
<td>518</td>
</tr>
<tr>
<td>SilentInvite</td>
<td>NOTEIVERB_SILENTINVITE</td>
<td>519</td>
</tr>
<tr>
<td>SilentCancel</td>
<td>NOTEIVERB_SILENTCANCEL</td>
<td>520</td>
</tr>
<tr>
<td>RecallMessage</td>
<td>NOTEIVERB_RECALL_MESSAGE</td>
<td>521</td>
</tr>
<tr>
<td>ForwardResponse</td>
<td>NOTEIVERB_FORWARD_RESPONSE</td>
<td>522</td>
</tr>
<tr>
<td>ForwardCancel</td>
<td>NOTEIVERB_FORWARD_CANCEL</td>
<td>523</td>
</tr>
<tr>
<td>FollowupClear</td>
<td>NOTEIVERB_FOLLOWUPCLEAR</td>
<td>524</td>
</tr>
<tr>
<td>ForwardAppointment</td>
<td>NOTEIVERB_FORWARD_APPT</td>
<td>525</td>
</tr>
<tr>
<td>OpenResend</td>
<td>NOTEIVERB_OPENRESEND</td>
<td>526</td>
</tr>
<tr>
<td>StatusReport</td>
<td>NOTEIVERB_STATUSREPORT</td>
<td>527</td>
</tr>
<tr>
<td>JournalOpen</td>
<td>NOTEIVERB_JOURNALOPEN</td>
<td>528</td>
</tr>
<tr>
<td>JournalOpenLink</td>
<td>NOTEIVERB_JOURNALOPENLINK</td>
<td>529</td>
</tr>
<tr>
<td>ComposeReplace</td>
<td>NOTEIVERB_COMPOSEREPLACE</td>
<td>530</td>
</tr>
<tr>
<td>Edit</td>
<td>NOTEIVERB_EDIT</td>
<td>531</td>
</tr>
<tr>
<td>DeleteProcess</td>
<td>NOTEIVERB_DELETEPROCESS</td>
<td>532</td>
</tr>
<tr>
<td>TentativeAppointmentTime</td>
<td>NOTEIVERB_TENTPNTIME</td>
<td>533</td>
</tr>
<tr>
<td>EditTemplate</td>
<td>NOTEIVERB_EDITTEMPLATE</td>
<td>534</td>
</tr>
<tr>
<td>FindInCalendar</td>
<td>NOTEIVERB_FINDINCALENDAR</td>
<td>535</td>
</tr>
<tr>
<td>ForwardAsFile</td>
<td>NOTEIVERB_FORWARDASFILE</td>
<td>536</td>
</tr>
<tr>
<td>ChangeAttendees</td>
<td>NOTEIVERB_CHANGE_ATTENDEES</td>
<td>537</td>
</tr>
<tr>
<td>RecalculateTitle</td>
<td>NOTEIVERB_RECALC_TITLE</td>
<td>538</td>
</tr>
<tr>
<td>PropertyChange</td>
<td>NOTEIVERB_PROP_CHANGE</td>
<td>539</td>
</tr>
<tr>
<td>ForwardAsVcal</td>
<td>NOTEIVERB_FORWARD_AS_VCAL</td>
<td>540</td>
</tr>
<tr>
<td>ForwardAsIcal</td>
<td>NOTEIVERB_FORWARD_AS_ICAL</td>
<td>541</td>
</tr>
<tr>
<td>ForwardAsBusinessCard</td>
<td>NOTEIVERB_FORWARD_AS_BCARD</td>
<td>542</td>
</tr>
<tr>
<td>DeclineAppointmentTime</td>
<td>NOTEIVERB_DECLPNTIME</td>
<td>543</td>
</tr>
<tr>
<td>Verb name</td>
<td>Alternate name</td>
<td>Value</td>
</tr>
<tr>
<td>-------------------------</td>
<td>----------------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>Process</td>
<td>NOTEIVERB_PROCESS</td>
<td>544</td>
</tr>
<tr>
<td>OpenWithWord</td>
<td>NOTEIVERB_OPENWITHWORD</td>
<td>545</td>
</tr>
<tr>
<td>OpenInstanceOfSeries</td>
<td>NOTEIVERB_OPEN_INSTANCE_OF_SERIES</td>
<td>546</td>
</tr>
<tr>
<td>FilloutThisForm</td>
<td>NOTEIVERB_FILLOUT_THIS_FORM</td>
<td>547</td>
</tr>
<tr>
<td>FollowupDefault</td>
<td>NOTEIVERB_FOLLOWUP_DEFAULT</td>
<td>548</td>
</tr>
<tr>
<td>ReplyWithMail</td>
<td>NOTEIVERB_REPLY_WITH_MAIL</td>
<td>549</td>
</tr>
<tr>
<td>ToDoToday</td>
<td>NOTEIVERB_TODO_TODAY</td>
<td>566</td>
</tr>
<tr>
<td>ToDoTomorrow</td>
<td>NOTEIVERB_TODO_TOMORROW</td>
<td>567</td>
</tr>
<tr>
<td>ToDoThisWeek</td>
<td>NOTEIVERB_TODO_THISWEEK</td>
<td>568</td>
</tr>
<tr>
<td>ToDoNextWeek</td>
<td>NOTEIVERB_TODO_NEXTWEEK</td>
<td>569</td>
</tr>
<tr>
<td>ToDoThisMonth</td>
<td>NOTEIVERB_TODO_THISMONTH</td>
<td>570</td>
</tr>
<tr>
<td>ToDoNextMonth</td>
<td>NOTEIVERB_TODO_NEXTMONTH</td>
<td>571</td>
</tr>
<tr>
<td>ToDoNoDate</td>
<td>NOTEIVERB_TODO_NODATE</td>
<td>572</td>
</tr>
<tr>
<td>FollowupComplete</td>
<td>NOTEIVERB_FOLLOWUPCOMPLETE</td>
<td>573</td>
</tr>
<tr>
<td>CopyToPostFolder</td>
<td>NOTEIVERB_COPYTOPOSTFOLDER</td>
<td>574</td>
</tr>
<tr>
<td>SeriesInvitationUpdateToPartialAttendeeList</td>
<td>NOTEIVERB_PARTIALRECIP_SILENTINVITE</td>
<td>579</td>
</tr>
<tr>
<td>SeriesCancellationUpdateToPartialAttendeeList</td>
<td>NOTEIVERB_PARTIALRECIP_SILENTCANCEL</td>
<td>580</td>
</tr>
</tbody>
</table>

### 2.2.1.15 PidTagLastVerbExecutionTime Property

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

The **PidTagLastVerbExecutionTime** property ([MS-OXPROPS] section 2.768) 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-OXCMSG] 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 2.800) 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 2.786) 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 2.794) 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 2.845) indicates whether an e-mail sender requests an e-mail delivery report 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 2.846) 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 2.837) 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 2.893) 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 2.2.1.36).

2.2.1.24  PidTagReceivedRepresentingEmailAddress Property

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

The PidTagReceivedRepresentingEmailAddress property ([MS-OXPROPS] section 2.894) 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 2.2.1.37).

2.2.1.25  PidTagReceivedRepresentingEntryId Property

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

The PidTagReceivedRepresentingEntryId property ([MS-OXPROPS] section 2.895) 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 2.2.1.38).

2.2.1.26  PidTagReceivedRepresentingName Property

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

The PidTagReceivedRepresentingName property ([MS-OXPROPS] section 2.896) 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 message on his or her own behalf, this property is set to the value of the PidTagReceivedByName property (section 2.2.1.39).

2.2.1.27  PidTagReceivedRepresentingSearchKey Property

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

The PidTagReceivedRepresentingSearchKey property ([MS-OXPROPS] section 2.897) 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 2.2.1.40) 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  PidTagReceivedRepresentingSmtpAddress Property

Type: PtypString ([MS-OXCDATA] section 2.11.1)
The PidTagReceivedRepresentingSmtpAddress property ([MS-OXPROPS] section 2.898) contains the SMTP email address of the user represented by the receiving mailbox owner.

### 2.2.1.29 PidTagReadReceiptRequested Property

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

The PidTagReadReceiptRequested property ([MS-OXPROPS] section 2.883) 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).

### 2.2.1.30 PidTagReadReceiptSmtpAddress Property

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

The PidTagReadReceiptSmtpAddress property ([MS-OXPROPS] section 2.885) contains the SMTP email address of the user to whom a read receipt is directed.

### 2.2.1.31 PidTagNonReceiptNotificationRequested Property

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

The PidTagNonReceiptNotificationRequested property ([MS-OXPROPS] section 2.811) specifies whether the e-mail sender requests a non-read receipt from all recipients (1) if this message is deleted without being read. This property is set to 0x01 if the e-mail sender requests a non-read receipt from all recipients; otherwise, it is set to 0x00.

### 2.2.1.32 PidTagOriginalAuthorEntryId Property

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

The PidTagOriginalAuthorEntryId property ([MS-OXPROPS] section 2.823) contains an Address Book EntryID structure ([MS-OXCDATA] section 2.2.5.2) and is defined in report messages to identify the user who sent the original message.

### 2.2.1.33 PidTagOriginalAuthorName Property

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

The PidTagOriginalAuthorName property ([MS-OXPROPS] section 2.824) contains the display name of the sender of the original message referenced by a report message.

### 2.2.1.34 PidTagReportDisposition Property

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

The PidTagReportDisposition property ([MS-OXPROPS] section 2.920) contains a string indicating whether the original message was displayed to the user or deleted. It is only defined on report messages. Valid values for this property are as follows.
### 2.2.1.35 PidTagReportDispositionMode Property

**Type:** \texttt{PtypString} ([MS-OXCDATA] section 2.11.1)

The \texttt{PidTagReportDispositionMode} property ([MS-OXPROPS] section 2.921) contains a description of the action that a client has performed on behalf of a user (report messages only). Valid values for this property are as follows,

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>displayed</td>
<td>The message was viewed by the user.</td>
</tr>
<tr>
<td>deleted</td>
<td>The message was erased by the user without being read.</td>
</tr>
</tbody>
</table>

The client displayed a prompt to the user before sending the report message.

The client did not display a prompt to the user before sending the report message.

### 2.2.1.36 PidTagReceivedByAddressType Property

**Type:** \texttt{PtypString} ([MS-OXCDATA] section 2.11.1)

The \texttt{PidTagReceivedByAddressType} property ([MS-OXPROPS] section 2.887) contains the e-mail message receiver's e-mail \texttt{address type}, as specified by the \texttt{AddressType} field of the \texttt{RecipientRow} structure ([MS-OXCDATA] section 2.8.3.2).

### 2.2.1.37 PidTagReceivedByEmailAddress Property

**Type:** \texttt{PtypString} ([MS-OXCDATA] section 2.11.1)

The \texttt{PidTagReceivedByEmailAddress} property ([MS-OXPROPS] section 2.888) contains the e-mail message receiver's e-mail address, as specified by the \texttt{EmailAddress} field of the \texttt{RecipientRow} structure ([MS-OXCDATA] section 2.8.3.2).

### 2.2.1.38 PidTagReceivedByEntryId Property

**Type:** \texttt{PtypBinary} ([MS-OXCDATA] section 2.11.1)

The \texttt{PidTagReceivedByEntryId} property ([MS-OXPROPS] section 2.889) identifies an \texttt{address book EntryID} that contains the e-mail message receiver of the \texttt{E-mail object}. The address book EntryID data format is specified by the \texttt{EntryID} field of the \texttt{RecipientRow} structure ([MS-OXCDATA] section 2.8.3.2).

### 2.2.1.39 PidTagReceivedByName Property

**Type:** \texttt{PtypString} ([MS-OXCDATA] section 2.11.1)
The PidTagReceivedByName property ([MS-OXPROPS] section 2.890) 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).

2.2.1.40 PidTagReceivedBySearchKey Property

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

The PidTagReceivedBySearchKey property ([MS-OXPROPS] section 2.891) 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).

2.2.1.41 PidTagReceivedBySmtpAddress Property

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

The PidTagReceivedBySmtpAddress property ([MS-OXPROPS] section 2.892) contains the email message receiver's SMTP email address.

2.2.1.42 PidTagRecipientReassignmentProhibited Property

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

The PidTagRecipientReassignmentProhibited property ([MS-OXPROPS] section 2.906) 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.

2.2.1.43 PidTagReplyRecipientEntries Property

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

The PidTagReplyRecipientEntries property ([MS-OXPROPS] section 2.915) 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 2.2.1.44) 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 2.2.1.50). 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.

2.2.1.44 PidTagReplyRecipientNames Property

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

The PidTagReplyRecipientNames property ([MS-OXPROPS] section 2.916) 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 2.2.1.51). If present, the property is set
to one string containing the address book entry's recipient (2) display names separated by semicolons.

2.2.1.45   **PidTagReplyRequested Property**

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

The *PidTagReplyRequested* property ([MS-OXPROPS] section 2.917) 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.

2.2.1.46   **PidTagResponseRequested Property**

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

The *PidTagResponseRequested* property ([MS-OXPROPS] section 2.930) 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 2.2.1.75). 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.

2.2.1.47   **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.

2.2.1.48   **PidTagSenderAddressType Property**

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

The *PidTagSenderAddressType* property ([MS-OXPROPS] section 2.1000) contains the sending mailbox owner's e-mail address type, as specified by the *EmailAddress* field of the RecipientRow structure ([MS-OXCDATA] section 2.8.3.2 and section 2.2.4.3).

2.2.1.49   **PidTagSenderEmailAddress Property**

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

The *PidTagSenderEmailAddress* property ([MS-OXPROPS] section 2.1001) 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).

2.2.1.50   **PidTagSenderEntryId Property**

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

The *PidTagSenderEntryId* property ([MS-OXPROPS] section 2.1002) 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).
2.2.1.51 PidTagSenderName Property

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

The PidTagSenderName property ([MS-OXPROPS] section 2.1004) 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).

2.2.1.52 PidTagSenderSearchKey Property

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

The PidTagSenderSearchKey property ([MS-OXPROPS] section 2.1005) 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).

2.2.1.53 PidTagSenderSmtpAddress Property

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

The PidTagSenderSmtpAddress property ([MS-OXPROPS] section 2.1006) contains the Simple Mail Transfer Protocol (SMTP) e-mail address format of the e-mail address of the sending mailbox owner.

2.2.1.54 PidTagSentRepresentingAddressType Property

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

The PidTagSentRepresentingAddressType property ([MS-OXPROPS] section 2.1012) 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 2.2.1.48).

2.2.1.55 PidTagSentRepresentingEmailAddress Property

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

The PidTagSentRepresentingEmailAddress property ([MS-OXPROPS] section 2.1013) 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 PidTagSenderId property (section 2.2.1.49).

2.2.1.56 PidTagSentRepresentingEntryId Property

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

The PidTagSentRepresentingEntryId property ([MS-OXPROPS] section 2.1014) 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 PidTagSenderId property (section 2.2.1.50).
2.2.1.57  **PidTagSentRepresentingName Property**

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

The **PidTagSentRepresentingName** property ([MS-OXPROPS] section 2.1016) 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 2.2.1.51).

2.2.1.58  **PidTagSentRepresentingSearchKey Property**

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

The **PidTagSentRepresentingSearchKey** property ([MS-OXPROPS] section 2.1017) 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 2.2.1.52).

2.2.1.59  **PidTagSentRepresentingSmtpAddress Property**

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

The **PidTagSentRepresentingSmtpAddress** property ([MS-OXPROPS] section 2.1018) contains the SMTP e-mail address of the end user who is represented by the sending mailbox owner.

2.2.1.60  **PidTagSubjectPrefix Property**

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

On an E-mail object, the **PidTagSubjectPrefix** property ([MS-OXCMMSG] 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: "

2.2.1.61  **PidTagTransportMessageHeaders Property**

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

The **PidTagTransportMessageHeaders** property ([MS-OXPROPS] section 2.1050) 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 [RFC5253]).
2.2.1.62  **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.

2.2.1.63  **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.

2.2.1.64  **PidTagPrimarySendAccount Property**

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

The **PidTagPrimarySendAccount** property ([MS-OXPROPS] section 2.869) 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.

2.2.1.65  **PidTagNextSendAcct Property**

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

The **PidTagNextSendAcct** property ([MS-OXPROPS] section 2.806) 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.

2.2.1.66  **PidLidUseTnef Property**

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

The **PidLidUseTnef** property ([MS-OXPROPS] section 2.347) is set to 0x01 (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 0x00 (FALSE). If this property is absent, implementers of this protocol MUST NOT include TNEF on the message.

This property is set by either the client or the server, depending on which one is performing the conversion. For information about conversions involving TNEF and MIME, see [MS-OXTNEF] and [MS-OXCMAIL].

2.2.1.67  **Attachments**

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

2.2.1.68  **Categories and Keywords**

The client can set categories or keywords on an e-mail message as specified in [MS-OXCMSG] section 2.2.1.17.
2.2.1.69 Contacts

The client can set the contacts on an e-mail message as specified in [MS-OXOCNTC] and [MS-OXCMSG] section 2.2.1.57.2.

2.2.1.70 Flags

The client can set flags as specified in [MS-OXOFLAG].

2.2.1.71 Reminders

The client can set reminders as specified in [MS-OXORMDR].

2.2.1.72 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-OXCMSG] section 2.2.3.5. For each recipient (2), the client sets the PidTagRecipientType property (section 2.2.3.1) to 0x00000001 for a primary recipient, 0x00000002 for a Cc recipient, or 0x00000003 for a Bcc recipient. For details about the RecipientRow structure, which is used in the RopModifyRecipients ROP request buffer, see [MS-OXCDATA] section 2.8.3.2.

2.2.1.73 PidLidAutoProcessState Property

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

The PidLidAutoProcessState property ([MS-OXPROPS] section 2.40) specifies the options used in the processing of voting and tracking for 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.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000000</td>
<td>The client will not process the voting and tracking for the message.</td>
</tr>
<tr>
<td>0x00000001</td>
<td>The client will process the voting and tracking when the message is received or opened.</td>
</tr>
<tr>
<td>0x00000002</td>
<td>The client will process the voting and tracking only when the message is opened.</td>
</tr>
</tbody>
</table>

2.2.1.74 PidLidVerbStream Property

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

The PidLidVerbStream property ([MS-OXPROPS] section 2.350) specifies what voting responses the user can make in response to the message. The format is 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)     |
 | ...               | ...             | ...                           |
```
Version (WORD): Set to 0x0102.

Count (DWORD): Specifies the count of the `VoteOption` structures plus the count of the `VoteOptionExtras` structures to follow.

`VoteOption1` (variable length): The first `VoteOption` structure specified in section 2.2.1.74.1, where the 'N' in the field name represents the count of all the `VoteOption` structures in the `PidLidVerbStream`.

`VoteOptionN` (variable length): The last `VoteOption` structure specified in section 2.2.1.74.1, where the 'N' in the field name represents the count of all the `VoteOption` structures in the `PidLidVerbStream`.

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

`VoteOptionExtras1` (variable length): The first `VoteOptionExtras` structure specified in section 2.2.1.74.2.

`VoteOptionExtrasN` (variable length): The last `VoteOptionExtras` structure specified in section 2.2.1.74.2, where the 'N' in the field name represents the count of all the `VoteOptionExtras` structures in the `PidLidVerbStream`.

2.2.1.74.1  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.
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VerbType (DWORD)</td>
<td>The verb used by this structure. Set to 4 (0x00000004).</td>
</tr>
<tr>
<td>DisplayNameCount (1 byte)</td>
<td>The count of characters in the DisplayName field.</td>
</tr>
<tr>
<td>DisplayName (variable)</td>
<td>The localized display name of the voting option (for example, &quot;Yes&quot;) as an ANSI string, without the null terminating character.</td>
</tr>
<tr>
<td>MsgClsNameCount (1 byte)</td>
<td>The count of characters in the MsgClsName field. Set to 8 (0x08).</td>
</tr>
<tr>
<td>MsgClsName (variable)</td>
<td>Set to &quot;IPM.Note&quot;, without the null terminating character.</td>
</tr>
<tr>
<td>Internal1StringCount (1 byte)</td>
<td>The count of characters in the following string. Set to 0x00 for voting options.</td>
</tr>
<tr>
<td>Internal1String (variable)</td>
<td>MUST NOT be present, as Internal1StringCount is 0x00 for a voting option.</td>
</tr>
<tr>
<td>DisplayNameCountRepeat (1 byte)</td>
<td>MUST have the same value as the DisplayNameCount field.</td>
</tr>
<tr>
<td>DisplayNameRepeat (variable)</td>
<td>MUST have the same value as the DisplayName field.</td>
</tr>
<tr>
<td>Internal2 (DWORD)</td>
<td>Set to 0x00000000.</td>
</tr>
<tr>
<td>Internal3 (1 byte)</td>
<td>Set to 0x00.</td>
</tr>
<tr>
<td>fUseUSHeaders (DWORD)</td>
<td>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.</td>
</tr>
<tr>
<td>Internal4 (DWORD)</td>
<td>Set to 0x00000001.</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000001</td>
<td>Automatically send the voting response message.</td>
</tr>
<tr>
<td>0x00000002</td>
<td>Prompt the user to specify whether he or she would like to automatically send or edit the voting response first.</td>
</tr>
</tbody>
</table>

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.

2.2.1.74.2 VoteOptionExtras Structure

Each element contains additional information about the corresponding VoteOption structure (section 2.2.1.74.1). The format is shown in the following diagram.

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>DisplayNameCount</td>
<td>DisplayName (variable)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DisplayNameCountRepeat</td>
<td>DisplayNameRepeat (variable)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 (variable): A duplicate instance of the display name, as a Unicode string without a null terminator. MUST have the same value as the DisplayName field.

2.2.1.75 PidLidVerbResponse Property

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

The PidLidVerbResponse property ([MS-OXPROPS] section 2.349) specifies the voting option that a respondent has selected. Corresponds to one of the values of the DisplayName field in the
VoteOption structure (section 2.2.1.74.1). If present, this property MUST be set to the textual description of the user interface element selected by the user.

2.2.1.76 PidTagTargetEntryId Property

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

The PidTagTargetEntryId property ([MS-OXPROPS] section 2.1039) 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.

2.2.1.77 PidTagAutoResponseSuppress Property

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

The PidTagAutoResponseSuppress property ([MS-OXPROPS] section 2.615) 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.

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

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000001</td>
<td>Suppress delivery reports.</td>
</tr>
<tr>
<td>0x00000002</td>
<td>Suppress non-delivery reports.</td>
</tr>
<tr>
<td>0x00000004</td>
<td>Suppress read notifications from clients that receive the message.</td>
</tr>
<tr>
<td>0x00000008</td>
<td>Suppress non-read notifications from clients that receive the message.</td>
</tr>
<tr>
<td>0x00000010</td>
<td>Suppress Out of Office (OOF) messages.</td>
</tr>
<tr>
<td>0x00000020</td>
<td>Suppress all auto-reply messages other than OOF notifications.</td>
</tr>
</tbody>
</table>

2.2.1.78 PidTagMessageEditorFormat Property

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

The PidTagMessageEditorFormat property ([MS-OXPROPS] section 2.790) specifies the format that an e-mail editor can use for editing the message body. Valid values are listed in the following table.

<table>
<thead>
<tr>
<th>Value name</th>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0x00000000</td>
<td>Suppress delivery reports.</td>
</tr>
<tr>
<td></td>
<td>0x00000001</td>
<td>Suppress non-delivery reports.</td>
</tr>
<tr>
<td></td>
<td>0x00000002</td>
<td>Suppress read notifications from clients that receive the message.</td>
</tr>
<tr>
<td></td>
<td>0x00000004</td>
<td>Suppress non-read notifications from clients that receive the message.</td>
</tr>
<tr>
<td></td>
<td>0x00000010</td>
<td>Suppress Out of Office (OOF) messages.</td>
</tr>
<tr>
<td></td>
<td>0x00000020</td>
<td>Suppress all auto-reply messages other than OOF notifications.</td>
</tr>
<tr>
<td>Value name</td>
<td>Value</td>
<td>Meaning</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------</td>
<td>--------------------------------------------------------------</td>
</tr>
<tr>
<td>EditorFormatDontKnow</td>
<td>0x00</td>
<td>The format for the editor to use is unknown.</td>
</tr>
<tr>
<td>EditorFormatPlainText</td>
<td>0x01</td>
<td>The optimal editing format is plain text.</td>
</tr>
<tr>
<td>EditorFormatHtml</td>
<td>0x02</td>
<td>The optimal editing format is HTML.</td>
</tr>
<tr>
<td>EditorFormatRtf</td>
<td>0x03</td>
<td>The optimal editing format is RTF.</td>
</tr>
</tbody>
</table>

### 2.2.1.79 PidTagMessageSubmissionId Property

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

The `PidTagMessageSubmissionId` property ([MS-OXPROPS] section 2.799) contains a unique identifier that indicates where the message originated. This property is optionally set, at the implementer's discretion, by a **message transfer agent (MTA)**.

### 2.2.1.80 PidTagSenderIdStatus Property

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

The `PidTagSenderIdStatus` property ([MS-OXPROPS] section 2.1003) 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.

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

### 2.2.1.81 PidTagListHelp Property

Type: **PtypString** ([MS-OXCDATA] section 2.11.1)
The **PidTagListHelp** property ([MS-OXPROPS] section 2.769) 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** in **MIME**, which is specified in [RFC2369]. Conversion between this property and the **List- Help header** is specified in [MS-OXCMAIL] section 2.1.3.2.15.

### 2.2.1.82 PidTagListSubscribe Property

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

The **PidTagListSubscribe** property ([MS-OXPROPS] section 2.770) contains the **URI** that subscribes a **recipient** (2) to the message's associated mailing list. This property corresponds to the **List-Subscribe header** in **MIME**, which is specified in [RFC2369]. Conversion between this property and the **List-Subscribe header** is specified in [MS-OXCMAIL] section 2.1.3.2.15.

### 2.2.1.83 PidTagListUnsubscribe Property

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

The **PidTagListUnsubscribe** property ([MS-OXPROPS] section 2.771) contains the **URI** that unsubscribes a **recipient** (2) from the message's associated mailing list. This property corresponds to the **List-Unsubscribe header** in **MIME**, which is specified in [RFC2369]. Conversion between this property and the **List-Unsubscribe header** is specified in [MS-OXCMAIL] section 2.1.3.2.15.

### 2.2.1.84 PidTagDelegatedByRule Property

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

The **PidTagDelegatedByRule** property ([MS-OXPROPS] section 2.666) specifies whether the message was resent due to the triggering of a **delegate rule**. This property is set to 0x01 (TRUE) if the message was resent; otherwise, it is set to 0x00 (FALSE). Delegate rules are specified in [MS-OXODLGT] section 2.2.3.

### 2.2.1.85 PidTagOriginalMessageId Property

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

The **PidTagOriginalMessageId** property ([MS-OXPROPS] section 2.831) is included in reply or resend messages to designate the message ID of the original message.

### 2.2.1.86 PidTagOriginalMessageClass Property

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

The **PidTagOriginalMessageClass** property ([MS-OXPROPS] section 2.830) is included in reply or resend messages to designate the value of the **PidTagMessageClass** property ([MS-OXCMSG] section 2.2.1.3) from the original message.

### 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-OXCMSG] 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-
type>", 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.

<table>
<thead>
<tr>
<th>Report type</th>
<th>Message class name (PtypString)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read receipt</td>
<td>REPORT.IPM.NOTE.IPNRN</td>
</tr>
<tr>
<td>Non-read receipt</td>
<td>REPORT.IPM.NOTE.IPNNRN</td>
</tr>
<tr>
<td>Delivery receipt</td>
<td>REPORT.IPM.NOTE.DR</td>
</tr>
<tr>
<td>Non-delivery report</td>
<td>REPORT.IPM.NOTE.NDR</td>
</tr>
</tbody>
</table>

### 2.2.2.2 PidTagOriginalDeliveryTime Property

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

The PidTagOriginalDeliveryTime property ([MS-OXPROPS] section 2.825) 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 2.828) is set on report messages by using the value of the PidTagDisplayTo property (section 2.2.9) 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 2.827) is set on report messages by using the value of the PidTagDisplayCc property (section 2.2.8) 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 2.826) is set on report messages by using the value of the PidTagDisplayBcc property (section 2.2.7) 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 2.832) is set on delivery report messages by using the value of the original message sender's PidTagSenderAddressType property (section 2.2.1.48), 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 2.833) is set on delivery report messages to the value of the original message sender's PidTagSenderEmailAddress property (section 2.2.1.49).

2.2.2.8 PidTagOriginalSenderEntryId Property

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

The PidTagOriginalSenderEntryId property ([MS-OXPROPS] section 2.834) contains an address book EntryID that is set on delivery report messages to the value of the PidTagSenderEntryId property (section 2.2.1.50) 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 2.835) is set on delivery report messages to the value of the original message sender's PidTagSenderName property (section 2.2.1.51).

2.2.2.10 PidTagOriginalSenderSearchKey Property

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

The PidTagOriginalSenderSearchKey property ([MS-OXPROPS] section 2.836) contains an address book search key that is set on delivery report messages to the value of the PidTagSenderSearchKey property (section 2.2.1.52) 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 2.838) 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 2.2.1.54) 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 2.839) 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 2.2.1.55) 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 2.840) 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 2.2.1.56) of the original message.

2.2.2.14 PidTagOriginalSentRepresentingName Property

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

The PidTagOriginalSentRepresentingName property ([MS-OXPROPS] section 2.841) 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 2.2.1.57) 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 2.842) 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 2.2.1.58) of the original message.

2.2.2.16 PidTagOriginalSubject Property

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

The PidTagOriginalSubject property ([MS-OXPROPS] section 2.843) specifies the subject of the original message and is set to the concatenated values of the PidTagSubjectPrefix property (section 2.2.1.60) 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 2.844) 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 2.860) 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 2.922) 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 2.924) 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 2.925) 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 2.926) 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 2.2.1.46) set to 0x01 or the PidTagReplyRequested property (section 2.2.1.45) 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 | 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 |
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| ... |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

...
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): This field specifies the version. If the SearchFolderEntryId field is present, this field MUST be set to 0x00020001; otherwise, this field MUST be set to 0x00010001.

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

StoreEntryId (variable length of bytes): This field specifies the entry ID of the mailbox that contains the original message. 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): This field specifies the entry ID of the folder that contains the original message. 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)**: This field specifies the entry ID of the original message. 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)**: Size of the **SearchFolderEntryId** field.

**SearchFolderEntryId (variable)**: This field specifies the entry ID of an alternate folder that contains the original message. 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)**: This field specifies the search key of the original message. 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 2.927) 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 2.879) 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 2.880) 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 2.881) 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 2.2.1.29) 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 2.2.1.50) 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 2.882) 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 2.884) 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 2.2.1.29) 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 2.2.1.52) is used to identify the user who receives the read receipt.

2.2.2.29 PidTagDeliverTime Property

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

The PidTagDeliverTime property ([MS-OXPROPS] section 2.671) contains the delivery time for a report message.

2.2.2.30 PidTagNonDeliveryReportDiagCode Property

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

The PidTagNonDeliveryReportDiagCode property ([MS-OXPROPS] section 2.808) contains a diagnostic code for a non-delivery report. For more details, see [MS-OXCMAIL] section 2.2.3.7.1.3.

2.2.2.31 PidTagNonDeliveryReportReasonCode Property

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

The PidTagNonDeliveryReportReasonCode property ([MS-OXPROPS] section 2.809) contains a value that provides information on the reason for a non-delivery report, as specified in [MS-OXCMAIL] section 2.2.3.7.1.3.

2.2.2.32 PidTagNonDeliveryReportStatusCode Property

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

The PidTagNonDeliveryReportStatusCode property ([MS-OXPROPS] section 2.810) contains a value derived from the Status header on a non-delivery report, as specified in [MS-OXCMAIL] section 2.2.3.7.1.3.

2.2.2.33 PidTagReceiptTime Property

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

The PidTagReceiptTime property ([MS-OXPROPS] section 2.886) contains the sent time for a read receipt.
2.2.2.34 PidTagRemoteMessageTransferAgent Property

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

The PidTagRemoteMessageTransferAgent property ([MS-OXPROPS] section 2.913) contains the name of the server that reported delivery status that resulted in a delivery receipt or non-delivery report. The value of this property corresponds to the value of the Remote-MTA header, as specified in [RFC3464].

2.2.2.35 PidTagReportingMessageTransferAgent Property

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

The PidTagReportingMessageTransferAgent property ([MS-OXPROPS] section 2.923) contains the name of the server that generated the delivery receipt or non-delivery report. The value of this property corresponds to the value of the Reporting-MTA header, as specified in [RFC3464].

2.2.2.36 PidTagSupplementaryInfo Property

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

The PidTagSupplementaryInfo property ([MS-OXPROPS] section 2.1035) contains supplementary information about a delivery receipt or a non-delivery report. For more details, see [MS-OXCMAIL] section 2.2.3.7.1.2.

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 2.909) represents the type of a recipient (2) on the message. This property is set on each recipient (2). Valid values for this property are as follows.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000000</td>
<td>The recipient (2) is the message originator.</td>
</tr>
<tr>
<td>0x00000001</td>
<td>The recipient (2) is a primary recipient.</td>
</tr>
<tr>
<td>0x00000002</td>
<td>The recipient (2) is a Cc recipient.</td>
</tr>
<tr>
<td>0x00000003</td>
<td>The recipient (2) is a Bcc recipient.</td>
</tr>
</tbody>
</table>

Additionally, the following flags apply to a resend message. These flags can be combined with the values listed in the previous table by using a bitwise OR. For details about resending a message, see section 3.2.4.5.

<table>
<thead>
<tr>
<th>Flag</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x10000000</td>
<td>Indicates that the resend message needs to be delivered to the recipient (1).</td>
</tr>
<tr>
<td>0x80000000</td>
<td>Indicates that the resend message does not need to be delivered to the recipient (1).</td>
</tr>
</tbody>
</table>
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 2.663) 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 2.665) 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.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000000</td>
<td>Minutes; for example, 60 seconds.</td>
</tr>
<tr>
<td>0x00000001</td>
<td>Hours; for example, 60x60 seconds.</td>
</tr>
<tr>
<td>0x00000002</td>
<td>Day; for example, 24x60x60 seconds.</td>
</tr>
<tr>
<td>0x00000003</td>
<td>Week; for example, 7x24x60x60 seconds.</td>
</tr>
</tbody>
</table>

2.2.3.4 PidTagDeferredSendTime Property

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

The PidTagDeferredSendTime property ([MS-OXPROPS] section 2.664) can be present if a client would like to defer sending the message after a specific amount of time, as determined by the implementation.

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.

\[
\text{PidTagDeferredSendTime} = \text{PidTagClientSubmitTime} + \text{PidTagDeferredSendNumber} \times \text{TimeOf(PidTagDeferredSendUnits)}
\]

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 2.688) 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 2.690) 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.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000000</td>
<td>Minutes; for example, 60 seconds.</td>
</tr>
<tr>
<td>0x00000001</td>
<td>Hours; for example, 60 × 60 seconds.</td>
</tr>
<tr>
<td>0x00000002</td>
<td>Days; for example, 24 × 60 × 60 seconds.</td>
</tr>
<tr>
<td>0x00000003</td>
<td>Weeks; for example, 7 × 24 × 60 × 60 seconds.</td>
</tr>
</tbody>
</table>

2.2.3.7 PidTagExpiryTime Property

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

The PidTagExpiryTime property ([MS-OXPROPS] section 2.689) 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.

\[
\text{PidTagExpiryTime} = \text{PidTagClientSubmitTime} + \text{PidTagExpiryNumber} \times \text{TimeOf(PidTagExpiryUnits)}
\]

2.2.3.8 PidTagDeleteAfterSubmit Property

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

The PidTagDeleteAfterSubmit property ([MS-OXPROPS] section 2.668) indicates whether the original message is 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.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>Do not delete the original message after it is sent.</td>
</tr>
</tbody>
</table>
### 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 2.789) 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 2.1011) represents the Sent Items folder for the message. This folder MUST NOT be a search folder. 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 2.635) 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.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x01</td>
<td>Delete the original message after it is sent.</td>
</tr>
<tr>
<td>Value name</td>
<td>Value</td>
</tr>
<tr>
<td>---------------</td>
<td>-------</td>
</tr>
<tr>
<td>None</td>
<td>0x00</td>
</tr>
<tr>
<td>PreProcess</td>
<td>0x01</td>
</tr>
<tr>
<td>NeedsSpooler</td>
<td>0x02</td>
</tr>
</tbody>
</table>

### 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", "MAPIPDL", "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.<6>

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.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>Help file data is not required to be in a format suited for 32-bit machines.</td>
</tr>
<tr>
<td>Any nonzero value</td>
<td>Help file data returned is required to be in a format suited for 32-bit machines.</td>
</tr>
</tbody>
</table>

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 MAY 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 is optional and MAY be included in a 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 RopSetSpooler 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-OXCDATA] 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-OXCDATA] 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.

<table>
<thead>
<tr>
<th>Value name</th>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>lstLock</td>
<td>0x00</td>
<td>Mark the message as locked.</td>
</tr>
<tr>
<td>lstUnlock</td>
<td>0x01</td>
<td>Mark the message as unlocked.</td>
</tr>
<tr>
<td>lstFinished</td>
<td>0x02</td>
<td>Mark the message as ready for processing by the server.</td>
</tr>
</tbody>
</table>

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 PropertyValue array. Only exists if the value of the NoPropertiesReturned field is 0x00.

PropertyValue (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).
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 Per Global

The following ADM elements are common to both client and server:

Global.Handle, as specified in [MS-OXCRPC] section 3.1.1.1.

Session context cookie, as specified in [MS-OXCMAPIHTTP] section 3.1.1.<7>

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.


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.Sender: The user who is sending the message.

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


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 element is 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 message store.
  - **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.

No ADM types other than those specified in section 3.1.1 are defined for the client.
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-OXCDATA] 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 by using the RopSetProperties ROP ([MS-OXCROPS] section 2.2.8.6) 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

To send a message at a later time, a client sets the PidTagDeferredSendTime property (section 2.2.3.4).

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

To set an expiration time on a message, a client sets the PidTagExpiryTime property (section 2.2.3.7).
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 2.2.1.76) to the value of the `PidTagEntryId` property ([MS-OXCPERM] section 2.2.4) 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 a resend message by setting `mfrResend` in the `PidTagMessageFlags` property ([MS-OXCMSG] section 2.2.1.86). A client MUST also set the value of the `PidTagOriginalMessageClass` property (section 2.2.1.86) to resend the message.

The client sets the flags in the `PidTagRecipientType` property (section 2.2.3.1) of a recipient (1) to indicate whether that recipient is to receive the resend message. The server will attempt to redeliver the message only to the recipients (1) indicated.

### 3.2.4.6 Soliciting Votes from Voters

To enable voting on a message and solicit votes from voters, a client sets a specific set of properties on a message. 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 2.2.1.74) 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 2.2.1.75).

#### 3.2.4.6.1 Associating Options with a Voting Message

To associate a series of voting options with a message, a client sets the `PidLidVerbStream` property (section 2.2.1.74).

#### 3.2.4.6.2 Interpreting a Voting Message

When a client receives a message, it MUST check the `PidLidVerbStream` property (section 2.2.1.74). 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. <8>
3.2.4.6.3 Crafting a Voting Response Message

To craft a voting response message, a client MUST set the following properties on a Message object:

- The PidTagSubjectPrefix property (section 2.2.1.60) set to the display name of the voting option chosen by the user.
- The PidLidVerbResponse property (section 2.2.1.75) 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 0x00000002 (prompt before sending), and if the user selects "Edit", the 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.<9>

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. The messages that are placed in the spooler queue are processed as specified in section 3.2.4.8.

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,<10> 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.

### 3.2.4.10 Sending Read Receipts and Non-Read Receipts

To send a read receipt, a client first checks to see whether the PidTagReadReceiptRequested property (section 2.2.1.29) is set. To send a non-read receipt, a client checks to see whether the PidTagNonReceiptNotificationRequested property (section 2.2.1.31) is set.

If the PidTagNonReceiptNotificationRequested property is set to 0x00 and the PidTagReadReceiptRequested property (section 2.2.1.29) is set to 0x01, the client SHOULD send a non-read receipt if the message is deleted without being read.

The client SHOULD also set the following properties on the Message object representing the receipt for both read receipt and non-read receipt messages:

- PidTagInReplyToId (section 2.2.1.13)
- PidTagReportTag (section 2.2.2.22)
- PidTagReportText (section 2.2.2.23)
- PidTagMessageClass ([MS-OXCMSG] section 2.2.1.3)
- PidTagReportTime ([MS-OXCSPAM] section 2.2.2.6)
- PidTagOriginalAuthorEntryId (section 2.2.1.32)
- PidTagOriginalAuthorName (section 2.2.1.33)
- PidTagReportDisposition (section 2.2.1.34)
- PidTagReportDispositionMode (section 2.2.1.35)
- PidTagOriginalDeliveryTime (section 2.2.2.17)
- PidTagParentKey (section 2.2.1.18)
- PidTagDeleteAfterSubmit (section 2.2.3.8)
- PidTagOriginalSubject (section 2.2.2.16)
- PidTagReceivedRepresentingName (section 2.2.1.26)
- PidTagSentRepresentingName (section 2.2.1.57)
- PidTagSentRepresentingEntryId (section 2.2.1.56)
- PidTagOriginalSubmitTime (section 2.2.2.17)
- PidTagOriginalDisplayTo (section 2.2.2.3)
- PidTagOriginalDisplayCc (section 2.2.2.4)
- PidTagOriginalDisplayBcc (section 2.2.2.5)
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 2.2.1.76) on the E-mail object, it SHOULD set the following properties at the same time:

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PidTagEntryId ([MS-OXCPERM] section 2.2.4)</td>
<td>SHOULD contain the same value as the PidTagTargetEntryId property, if present.</td>
</tr>
<tr>
<td>PidTagMessageFlags ([MS-OXCMSG] section 2.2.1.6)</td>
<td>The mfUnsent and mfRead flags MUST be cleared.</td>
</tr>
<tr>
<td>PidTagInternetMessageId (section 2.2.1.12)</td>
<td>The value SHOULD be copied from the original message.</td>
</tr>
</tbody>
</table>

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 RopSpoolerLockMessage ROP Request

The client sends a RopSpoolerLockMessage ROP request ([MS-OXCROPS] section 2.2.7.5) after determining, through an implementation-dependent mechanism, that it can handle the message.<11>

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.

**MessageObject**

**EmailObject**

**User**

3.3.1.1 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.

3.3.1.2 Per E-mail Object

**E-mail objects** are represented by the **EmailObject** ADM data type. The following ADM element is 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**.

3.3.1.3 Per User

A logged-in user is represented by the **User** ADM data type. The following ADM element is 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 RobSubmitMessage 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 RobSubmitMessage ROP.

<table>
<thead>
<tr>
<th>Error code name</th>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ecShutoffQuotaExceeded</td>
<td>0x000004DD</td>
<td>Indicates that the maximum storage shut-off quota, determined by the server implementer or server administrator, has been exceeded.</td>
</tr>
<tr>
<td>ecQuotaExceeded</td>
<td>0x000004D9</td>
<td>Indicates that the storage quota, determined by the server implementer or server administrator, is exceeded for the mailbox, but the user can still receive mail.</td>
</tr>
<tr>
<td>ecNotSupported</td>
<td>0x80040102</td>
<td>Indicates that the Server object that is associated with the InputHandleIndex field in the Server object table is not a Message object, or that the current logon session is a public logon.</td>
</tr>
<tr>
<td>ecTooManyRecips</td>
<td>0x00000505</td>
<td>Indicates that the number of recipients (2) on the message exceeds the allowed limit, which is determined by the server implementer or server administrator. If this error occurs, none of the recipients (2) will receive this message.</td>
</tr>
<tr>
<td>ecAccessDenied</td>
<td>0x80070005</td>
<td>Indicates that the message is a folder associated information (FAI) message.</td>
</tr>
<tr>
<td>ecRequiresRefResolve</td>
<td>0x0000047E</td>
<td>Indicates that the path specified in the PidTagAttachLongFilename property ([MS-OXCMSG] section 2.2.2.10) to the attachment is not valid.</td>
</tr>
</tbody>
</table>

The server performs the operations specified in sections 3.3.5.1.1 through 3.3.5.1.4 on receipt of the RobSubmitMessage ROP request.

3.3.5.1.1 Permission Check

There are restrictions on the e-mail messages that can be submitted with a RobSubmitMessage 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.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Error code name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAI message is submitted.</td>
<td>ecAccessDenied</td>
<td>0x80070005</td>
</tr>
<tr>
<td>Embedded Message object is submitted.</td>
<td>ecNotSupported</td>
<td>0x80040102</td>
</tr>
<tr>
<td>Upper limit of recipients (2) is exceeded.</td>
<td>ecTooManyRecips</td>
<td>0x00000505</td>
</tr>
</tbody>
</table>
### Condition

<table>
<thead>
<tr>
<th>Condition</th>
<th>Error code name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mailbox is running out of quota.</td>
<td>ecQuotaExceeded</td>
<td>0x000004D9</td>
</tr>
<tr>
<td>No write permission on the message.</td>
<td>ecAccessDenied</td>
<td>0x80070005</td>
</tr>
</tbody>
</table>

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-OXCCDATA] 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 2.2.1.76): 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 2.2.4) 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 about message moves 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 2.2.1.54)
- **PidTagSentRepresentingEmailAddress** (section 2.2.1.55)
- **PidTagSentRepresentingEntryId** (section 2.2.1.56)
- **PidTagSentRepresentingName** (section 2.2.1.57)
- **PidTagSentRepresentingSearchKey** (section 2.2.1.58)

### 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 2.2.1.48)
- **PidTagSenderEmailAddress** (section 2.2.1.49)
- **PidTagSenderEntryId** (section 2.2.1.50)
- **PidTagSenderName** (section 2.2.1.51)
- **PidTagSenderSearchKey** (section 2.2.1.52)

### 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.

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

#### 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).

<table>
<thead>
<tr>
<th>Error code name</th>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ecBufferTooSmall</td>
<td>0x0000047D</td>
<td>The response buffer is not large enough to hold the results.</td>
</tr>
<tr>
<td>ecNullObject</td>
<td>0x000004B9</td>
<td>An object handle reference in the RPC buffer could not be resolved.</td>
</tr>
</tbody>
</table>
### Error code name | Value | Meaning
--- | --- | ---
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-OXCDATA] 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.

<table>
<thead>
<tr>
<th>Error code name</th>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
</table>
ecNullObject | 0x000004B9 | The **InputHandleIndex** field is not valid. [13]

#### 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.

<table>
<thead>
<tr>
<th>Value name</th>
<th>Action</th>
</tr>
</thead>
</table>
estLock | Locks the message for the client that is sending the request. The request fails if the message is locked by some other client. |
estUnlock | Unlocks the message. |
estFinished | Unlocks the message and completes post-processing of sent mail according to 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.

<table>
<thead>
<tr>
<th>Error code name</th>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
</table>
ecNotSupported | 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. [14] |
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 code can be returned in a RopTransportSend ROP response.

<table>
<thead>
<tr>
<th>Error code name</th>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ecNotSupported</td>
<td>0x80040102</td>
<td>The server could not handle the message and the message was not sent. The client SHOULD try another server if one is available.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Error code name</th>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ecNotSupported</td>
<td>0x80040102</td>
<td>The server did not receive a valid corresponding call for the RopSetSpooler ROP ([MS-OXCROPS] section 2.2.7.4).</td>
</tr>
</tbody>
</table>

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).

<table>
<thead>
<tr>
<th>Error code name</th>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ecNullObject</td>
<td>0x000004B9</td>
<td>The InputHandleIndex field does not refer to a valid Server object.</td>
</tr>
</tbody>
</table>

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
```

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

[MS-OXOMSG] - v20210817
Email Object Protocol
Copyright © 2021 Microsoft Corporation
Release: August 17, 2021
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-OXCDATA] section 2.2.1.1) of the parent folder)

**MessageId**: 0001-0003b7e65fa7 (the MID ([MS-OXCDATA] 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
```

[MS-OXOMSG] - v20210817

Email Object Protocol

Copyright © 2021 Microsoft Corporation

Release: August 17, 2021
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.

<table>
<thead>
<tr>
<th>Property</th>
<th>Property ID</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PidTagBody</strong> ([MS-OXCMSG] section 2.2.1.56.1)</td>
<td>0x1000</td>
<td>0x001f (PtypString)</td>
<td>&quot;Please sign the order request.\LF\CR&quot;</td>
</tr>
<tr>
<td><strong>PidTagMessageClass</strong> ([MS-OXCMSG] section 2.2.1.3)</td>
<td>0x001A</td>
<td>0x001F (PtypString)</td>
<td>&quot;IPM.Note&quot;</td>
</tr>
<tr>
<td><strong>PidTagMessageFlags</strong> ([MS-OXCMSG] section 2.2.1.6)</td>
<td>0x0E07</td>
<td>0x0003 (PtypInteger32)</td>
<td><strong>mfUnsent</strong></td>
</tr>
<tr>
<td><strong>PidTagConversationTopic</strong> (section 2.2.1.5)</td>
<td>0x0070</td>
<td>0x001f (PtypString)</td>
<td>&quot;Order Request&quot;</td>
</tr>
<tr>
<td><strong>PidTagConversationIndex</strong> (section 2.2.1.3)</td>
<td>0x0071</td>
<td>0x0102 (PtypBinary)</td>
<td>22 bytes 01 c8 74 0b 0f 9c 35 2c 02 17 93 af 43 a9 8b b4 c1 bb ef 97 7d 4f</td>
</tr>
<tr>
<td><strong>PidTagImportance</strong> ([MS-OXCMSG] section 2.2.1.11)</td>
<td>0x0017</td>
<td>0x0003 (PtypInteger32)</td>
<td>0x00000002 High Importance</td>
</tr>
<tr>
<td><strong>PidTagMessageDeliveryTime</strong> (section 2.2.3.9)</td>
<td>0x0E06</td>
<td>0x0040 (PtypTime)</td>
<td>2008/02/20 21:53:00.000</td>
</tr>
<tr>
<td><strong>PidTagReadReceiptRequested</strong> (section 2.2.1.29)</td>
<td>0x0029</td>
<td>0x000B (PtypBoolean)</td>
<td>0x01 (TRUE)</td>
</tr>
<tr>
<td><strong>PidTagSentMailSvrEID</strong> ([MS-OXCROPS] section 2.2.3.10)</td>
<td>0x6740</td>
<td>0x00FB (PtypServerId)</td>
<td>21 bytes 01 01 00 00 00 00 00 f0 e7</td>
</tr>
</tbody>
</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 2e 00 00 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 00 e7-cl 00 00 10 00 00 00 fd 
0050: 02 6f a5 55 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

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNPrefixLen</td>
<td>0x5A (90)</td>
<td></td>
</tr>
<tr>
<td>EX-Address.Type</td>
<td>0x00000000</td>
<td>DT_MAILUSER</td>
</tr>
<tr>
<td>EX-Address.EmailAddress</td>
<td><a href="mailto:edbanti@example.com">edbanti@example.com</a></td>
<td></td>
</tr>
<tr>
<td>DisplayName</td>
<td>Ed Banti</td>
<td></td>
</tr>
<tr>
<td>SimpleDisplayName</td>
<td>Ed Banti</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Property</th>
<th>PropertyID</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PidTagObjectType ([MS-OXPRPT] section 2.2.1.7)</td>
<td>0x0FFE</td>
<td>0x0003 (PtypInteger32)</td>
<td>0x00000006 (MAILUSER)</td>
</tr>
<tr>
<td>PidTagDisplayType ([MS-OXOABK] section 2.2.3.11)</td>
<td>0x3900</td>
<td>0x0003 (PtypInteger32)</td>
<td>0x0000000000 DT_MAILUSER</td>
</tr>
<tr>
<td>PidTagAddressBookDisplayNamePrintable ([MS-OXCMGS] section 2.2.1.30)</td>
<td>0x39FF</td>
<td>0x001F (PtypString)</td>
<td>Ed Banti</td>
</tr>
<tr>
<td>PidTagSmtpAddress ([MS-OXOABK] section 2.2.3.21)</td>
<td>0x39FE</td>
<td>0x001F (PtypString)</td>
<td><a href="mailto:edbanti@example.com">edbanti@example.com</a></td>
</tr>
<tr>
<td>PidTagSendInternetEncoding ([MS-OXOABK] section 2.2.3.19)</td>
<td>0x3A71</td>
<td>0x0003 (PtypInteger32)</td>
<td>0x0000000000</td>
</tr>
<tr>
<td>PidTagAccount ([MS-OXOCNTC] section 2.2.1.10.11)</td>
<td>0x3A00</td>
<td>0x001F (PtypString)</td>
<td>edbanti</td>
</tr>
<tr>
<td>PidTagDisplayTypeEx ([MS-OXOABK] section 2.2.3.12)</td>
<td>0x3905</td>
<td>0x0003 (PtypInteger32)</td>
<td>0x400000000</td>
</tr>
<tr>
<td>PidTagRecipientTrackStatus ([MS-OXOCAL] section 2.2.4.10.2)</td>
<td>0x5FFF</td>
<td>0x0003 (PtypInteger32)</td>
<td>0x000000000</td>
</tr>
<tr>
<td>Unspecified property</td>
<td>0x5FDE</td>
<td>0x0003 (PtypInteger32)</td>
<td>0x00000000</td>
</tr>
<tr>
<td>PidTagRecipientFlags ([MS-OXOCAL] section 2.2.4.10.1)</td>
<td>0x5FFD</td>
<td>0x0003 (PtypInteger32)</td>
<td>0x000000001</td>
</tr>
<tr>
<td>PidTagRecipientDisplayName ([MS-OXOPROPS] section 2.899)</td>
<td>0x5FF6</td>
<td>0x001F (PtypString)</td>
<td>Ed Banti</td>
</tr>
<tr>
<td>PidTagRecipientEntryId ([MS-OXPROPS] section 2.900)</td>
<td>0x5FF7</td>
<td>0x0102 (PtypBinary)</td>
<td>126 bytes (see the sample value for the PidTagRecipientEntryId property following this table)</td>
</tr>
<tr>
<td>PidTagRecipientOrder ([MS-OXCMGS] section 2.2.1.40)</td>
<td>0x5FDF</td>
<td>0x0003 (PtypInteger32)</td>
<td>0x000000000</td>
</tr>
</tbody>
</table>

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

---

[MS-OXOMSG] - v20210817
Email Object Protocol
Copyright © 2021 Microsoft Corporation
Release: August 17, 2021
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.

<table>
<thead>
<tr>
<th>Property</th>
<th>PropertyID</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>PidTagSubjectPrefix</code> (section 2.2.1.60)</td>
<td>0x0003</td>
<td>0x001F (PtypString)</td>
<td>Empty string</td>
</tr>
<tr>
<td><code>PidTagNormalizedSubject</code> ([MS-OXCMSG] section 2.2.1.10)</td>
<td>0x0E1D</td>
<td>0x001F (PtypString)</td>
<td>&quot;Order Form Issue&quot;</td>
</tr>
</tbody>
</table>

The client then sends a `RopSubmitMessage` ROP request ([MS-OXCROPS] 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-OXCROPS] section 2.2.15.3) to release the E-mail object.

For more details about the ROPs used in this example, see [MS-OXCROPS], [MS-OXCMSG], 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 2.2.1.74).

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 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 00
0060: 00 00 02 00 00 00 FF-FF 04 00 00 00 00 05 4D
0070: 61 79 62 65 00 49 45 50 4D-74 65 00 05 4D
0080: 59 65 73 00 03 59 65 73 00 05 4D
0090: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 FF
00A0: 04 01 03 59 60 05 00-73 00 03 59 60 05 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 2.2.1.74.

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.

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 (prompt before sending)
Internal5: 0x00000002
ID: 0x00000001
Internal6: 0xFF000000

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.

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.

DisplayNameCount: 0x03

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

00A1: 04 01

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.

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.

DisplayNameCount: 0x03

00A3: 03 59 00 65 00 73 00 03-59 00 65 00 73 00
**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 BB97 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
```

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 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**: 0001-000000000025

### 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 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
```
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. The types in the following table are described in [MS-OXCDATA] section 2.11.1.

<table>
<thead>
<tr>
<th>Property ID</th>
<th>Property name</th>
<th>Type</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00480040</td>
<td>Unspecified property</td>
<td>PtypTime</td>
<td>2008/05/06 17:46:09.035</td>
</tr>
<tr>
<td>0x00470102</td>
<td>PidTagMessageSubmissionId (section 2.2.1.79)</td>
<td>PtypBinary</td>
<td>Error: 0x8004010f (MAPI_E_NOT_FOUND)</td>
</tr>
<tr>
<td>0x0C1A001E</td>
<td>PidTagSenderName (section 2.2.1.51)</td>
<td>PtypString8</td>
<td>&quot;user1&quot;</td>
</tr>
<tr>
<td>0x0C190102</td>
<td>PidTagSenderEntryId (section 2.2.1.50)</td>
<td>PtypBinary</td>
<td>See the data for the PidTagSenderEntryId property following the table (1).</td>
</tr>
<tr>
<td>0x0C1D01</td>
<td>PidTagSenderSearchKey</td>
<td>PtypBinary</td>
<td>See the data for the</td>
</tr>
<tr>
<td>Property ID</td>
<td>Property name</td>
<td>Type</td>
<td>Data</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------------------------</td>
<td>-----------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>02</td>
<td>(section 2.2.1.52)</td>
<td></td>
<td><strong>PidTagSenderSearchKey</strong> property following the table (2).</td>
</tr>
<tr>
<td>0x0042001</td>
<td><strong>PidTagSentRepresentingName</strong> (section 2.2.1.57)</td>
<td>PtypString8</td>
<td>&quot;user1&quot;</td>
</tr>
<tr>
<td>0x0041010</td>
<td><strong>PidTagSentRepresentingEntryId</strong> (section 2.2.1.56)</td>
<td>1. PtypBinary</td>
<td>See the data for the <strong>PidTagSentRepresentingEntryId</strong> property following the table (3).</td>
</tr>
<tr>
<td>0x003B010</td>
<td><strong>PidTagSentRepresentingSearchKey</strong> (section 2.2.1.58)</td>
<td>2. PtypBinary</td>
<td>See the data for the <strong>PidTagSentRepresentingSearchKey</strong> property following the table (4).</td>
</tr>
</tbody>
</table>

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

Size: 124

0000: 00 00 00 00 DC A7 40 C8=00 42 10 1A B4 B9 08 00 ......@..B......
0010: 2B 2F E1 82 01 00 00 00 00 00 00 00 00 2F 4F 3D 46 +/............0-F
0020: 49 52 53 54 20 4F 52 47=41 4E 49 5A 41 54 49 4F IRST ORGANIZATION
0030: 4E 2F 4F 55 3D 45 58 43=48 41 4E 47 45 20 41 44 H/OU=EXCHANGE AD
0040: 4D 49 4E 54 52 41 54 49 4F 49 56 45 20 47 52 4F MINISTRATIVE GRO
0050: 55 50 20 28 46 59 44=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=RECIPIENT
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=42 4F 48 46 32 33 52 4F MINISTRATIVE GRO
0040: 4F 48 46 32 33 53 50 44=4C 54 29 2F 43 4E 3D 52 45 43 49 50 49 45 4E DLT)/CN=RECIPIENT
0050: 54 43 49 50 49 45 4E 54=53 2F 43 4E 3D 55 53=45 52 31 00 R1.
0060: 52 31 00

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

Size: 124

0000: 00 00 00 00 DC A7 40 C8=00 42 10 1A B4 B9 08 00 ......@..B......
0010: 2B 2F E1 82 01 00 00 00 00 00 00 00 00 2F 4F 3D 46 +/............0-F
0020: 49 52 53 54 20 4F 52 47=41 4E 49 5A 41 54 49 4F IRST ORGANIZATION
0030: 4E 2F 4F 55 3D 45 58 43=48 41 4E 47 45 20 41 44 H/OU=EXCHANGE AD
0040: 4D 49 4E 54 52 41 54 49 4F 49 56 45 20 47 52 4F MINISTRATIVE GRO
0050: 55 50 20 28 46 59 44=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=RECIPIENT
0070: 54 53 2F 43 4E 3D 55 53=45 52 31 00 TS/CN=USER1.

Data for the **PidTagSentRepresentingSearchKey** property (4)
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.
Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include updates to those products.

- Microsoft Exchange Server 2003
- Microsoft Exchange Server 2007
- Microsoft Exchange Server 2010
- Microsoft Exchange Server 2013
- Microsoft Exchange Server 2016
- Microsoft Exchange Server 2019
- Microsoft Office Outlook 2003
- Microsoft Office Outlook 2007
- Microsoft Outlook 2010
- Microsoft Outlook 2013
- Microsoft Outlook 2016
- Microsoft Outlook 2019
- Microsoft Outlook 2021

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates 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 2.2.1.3: Exchange 2007, Exchange 2010, Office Outlook 2007, Outlook 2010, Outlook 2013, Outlook 2016, and Outlook 2019 compute the PidTagConversationIndex property for messages using a different algorithm. When computing the Current FILETIME field, the 24 low bits of the high part and the 16 high bits of the low part of the FILETIME are included in Current FILETIME high part and Current FILETIME low part, as shown in the following table.

<table>
<thead>
<tr>
<th>Eight most significant bits</th>
<th>40 bits</th>
<th>16 least significant bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Excluded</td>
<td>Included</td>
<td>Excluded</td>
</tr>
</tbody>
</table>

<3> Section 2.2.1.3: Exchange 2013, Exchange 2016, and Exchange 2019 set the Delta Code field to 1 and do not calculate the Time Delta field based on TimeDiff.
<4> Section 2.2.1.5: Exchange 2003 and Exchange 2007 do not set the PidTagConversationTopic property (section 2.2.1.5).

<5> Section 2.2.1.6: Exchange 2003 uses only the PidTagDeferredDeliveryTime property. Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016, and Exchange 2019 use only the PidTagDeferredSendTime property (section 2.2.3.4).

<6> Section 2.2.4.4: Outlook 2010, Outlook 2013, Outlook 2016, and Outlook 2019 do not send a RopOptionsData ROP request ([MS-OXCROPS] section 2.2.7.9).

<7> Section 3.1.1.1: Exchange 2003, Exchange 2007, Exchange 2010, the initial release version of Exchange 2013, Office Outlook 2003, Office Outlook 2007, Outlook 2010, and the initial release version of Outlook 2013 do not support the session context cookie. The session context cookie was introduced in Microsoft Exchange Server 2013 Service Pack 1 (SP1) and Microsoft Outlook 2013 Service Pack 1 (SP1).

<8> Section 3.2.4.6.2: Office Outlook 2007 also uses the PidLidVerbStream property (section 2.2.1.74) 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, Microsoft Office Outlook 2007 Service Pack 1 ignores VoteOption structures with unknown VerbTypes.

<9> 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.

<10> Section 3.2.4.8: Office Outlook 2003 and Office Outlook 2007 set the PidTagNextSendAcct property (section 2.2.1.65) 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.

<11> Section 3.2.5.3: Office Outlook 2003 and Office Outlook 2007 examine the PidTagNextSendAcct property (section 2.2.1.65).

<12> Section 3.3.5.3: Microsoft Exchange Server 2007 Service Pack 2 (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).

<13> 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.

<14> Section 3.3.5.6: Exchange 2010, Exchange 2013, Exchange 2016, and Exchange 2019 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.

<15> 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 this document since the last release. Changes are classified as Major, Minor, or None.

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.
- A document revision that captures changes to protocol functionality.

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 **None** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

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

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Revision class</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 Appendix A: Product Behavior</td>
<td>Updated list of supported products.</td>
<td>major</td>
</tr>
</tbody>
</table>
8 Index

A

Aborting a message submission example overview 80
ROP request buffer 80
ROP response buffer 80
Abstract data model
client 65
server 70
Abstract data model - server per E-mail object 71
User ADM data type 71
Abstract data model - server per Message object 71
Applicability 18
Attachments E-mail object properties 40

C

Capability negotiation 18
Categories and keywords E-mail object properties 40
Change tracking 95
Client
abstract data model 65
initialization 66
other local events 70
timer events 70
timers 66
Client - higher layer triggered events
deferring message send 66
delivering mail to the server 68
optimizing send 67
processing E-mail objects in the spooler queue 68
resending a message 67
sending a message 66
sending a message with expiry time 66
sending mail through a specific server 68
soliciting votes from voters 67
Client - sequencing rules
sending a RopAbortSubmit ROP request 70
sending a RopSpoolerLockMessage ROP request 70
sending a RopSubmitMessage ROP request 70
sending a RopTransportNewMail ROP request 70
Common
abstract data model 64
higher-layer triggered events 65
initialization 65
message processing events and sequencing rules 65
other local events 65
timer events 65
timers 65
Contacts E-mail object properties 41
Controlling sending and delivery of mail overview 17
Creating, opening, and saving e-mail objects overview 16

D

Data model - abstract
Examples - sending mail to a specific server
determining the transport folder overview 87
determining the transport folder ROP request
buffer 87
determining the transport folder ROP response
buffer 88
initialization overview 86
initialization ROP request buffer 86
initialization ROP response buffer 86
locking the message in the spooler queue folder for
processing overview 87
locking the message in the spooler queue folder for
processing ROP request buffer 87
locking the message in the spooler queue folder for
processing ROP response buffer 87
marking the message as ready for post-send
server processing overview 91
marking the message as ready for post-send
server processing ROP request buffer 91
marking the message as ready for post-send
server processing ROP response buffer 91
overview 86
sending the message overview 88
sending the message ROP request buffer 88
sending the message ROP response buffer 88
submitting the message to the spooler queue
folder 86
Examples - submitting a deferred message
overview 79
ROP request buffer 79
ROP response buffer 79
Examples - submitting a message
overview 78
ROP request buffer 78
ROP response buffer 78

F
Fields - vendor-extensible 18
Flags E-mail object properties 41

G
Glossary 10

H
Higher layer triggered events - client
deferring message send 66
delivering mail to the server 68
optimizing send 67
processing E-mail objects in the spooler queue 68
resending a message 67
sending a message 66
sending a message with expiry time 66
sending mail through a specific server 68
soliciting votes from voters 67
Higher-layer triggered events
server 72

I
Implementer - security considerations 92
Index of security parameters 92
Informative references 16
Introduction

Initialization
client 66
server 72

Message delivery ROPs
RopAbortSubmit ROP 59
RopGetAddressTypes ROP 59
RopOptionsData ROP 60
RopSubmitMessage ROP 58

Message processing - client
sending a RopAbortSubmit ROP request 70
sending a RopSpoolerLockMessage ROP request 70
sending a RopSubmitMessage ROP request 70
sending a RopTransportNewMail ROP request 70

Message processing - server
receiving a RopAbortSubmit ROP request 75
receiving a RopGetAddressTypes ROP request 75
receiving a RopGetTransportFolder ROP request 76
receiving a RopOptionsData ROP request 77
receiving a RopSetSpooler ROP request 76
receiving a RopSpoolerLockMessage ROP request 77
receiving a RopSubmitMessage ROP request 77
receiving a RopTransportNewMail ROP request 77
receiving a RopTransportSend ROP request 77

Message status reports properties
PidTagMessageClass property 47
PidTagOriginalDeliveryTime property 48
PidTagOriginalDisplayBcc property 48
PidTagOriginalDisplayCc property 48
PidTagOriginalDisplayTo property 48
PidTagOriginalSenderAddressType property 49
PidTagOriginalSenderEmailAddress property 49
PidTagOriginalSenderEntryId property 49
PidTagOriginalSenderName property 49
PidTagOriginalSenderSearchKey property 49
PidTagOriginalSentRepresentingAddressType property 49
PidTagOriginalSentRepresentingEmailAddress property 49
PidTagOriginalSentRepresentingEntryId property 50
PidTagOriginalSentRepresentingName property 50
PidTagOriginalSentRepresentingSearchKey property 50
PidTagOriginalSubject property 50
PidTagOriginalSubmitTime property 50
PidTagParentKey property 50
PidTagReadReceiptAddressType property 53
PidTagReadReceiptEmailAddress property 53
PidTagReadReceiptEntryId property 53
PidTagReadReceiptName property 54
PidTagReadReceiptSearchKey property 54
PidTagReportEntryId property 51
PidTagReportName property 51
PidTagReportSearchKey property 51
PidTagReportTag property 51
PidTagReportText property 53

Message syntax
E-Mail Object Properties 23
E-Mail Submission Properties 55
Spooler and Transport ROPs 61
syntax 19
transport 19

Non-delivery receipt overview 17
Non-read receipt overview 17
Normative references 15

Other local events
client 70
server 77

Overview
controlling sending and delivery of mail 17
report messages 17
voting and tracking 17

Overview (synopsis) 16

Parameters - security index 92
PidLidAutoProcessState E-mail object property 41
PidLidInternetAccountName E-mail object property 40
PidLidInternetAccountStamp E-mail object property 40
PidLidUseTnf E-mail object property 40
PidLidVerbResponse E-mail object property 44
PidLidVerbStream E-mail object property 41
PidTagAutoResponseSuppress E-mail object property 45

PidTagBlockStatus E-mail object property 23
PidTagClientSubmitTime e-mail submission property 58
PidTagConversationId E-mail object property 24
PidTagConversationIndex E-mail object property 24
PidTagConversationIndexTracking E-mail object property 26
PidTagDeferredDeliveryTime E-mail object property 26
PidTagDeferredSendNumber e-mail submission property 56
PidTagDeferredSendTime e-mail submission property 56
PidTagDeferredSendUnits e-mail submission property 56
PidTagDelegatedByRule E-mail object property 47
PidTagDeleteAfterSubmit e-mail submission property 57
PidTagDisplayBcc E-mail object property 26
PidTagDisplayCc E-mail object property 26
PidTagDisplayTo E-mail object property 27
PidTagExpiryNumber e-mail submission property 57
PidTagExpiryTime e-mail submission property 57
PidTagExpiryUnits e-mail submission property 57
PidTagIconIndex E-mail object property 27
PidTagInReplyTold E-mail object property 29
PidTagInternetMailOverrideFormat E-mail object property 28
PidTagInternetMessageId E-mail object property 29
PidTagLastVerbExecuted E-mail object property 29
Sending mail to a specific server example – locking the message in the spooler queue folder for processing
overview 87
ROP request buffer 87
ROP response buffer 87
Sending mail to a specific server example – marking the message as ready for post-send server processing
overview 91
ROP request buffer 91
ROP response buffer 91
Sending mail to a specific server example - overview 86
Sending mail to a specific server example – sending the message
overview 88
ROP request buffer 88
ROP response buffer 88
Sending mail to a specific server example – submitting the message to the spooler queue folder 86
Sending messages overview 16
Sequencing rules - client
sending a RopAbortSubmit ROP request 70
sending a RopSpoolerLockMessage ROP request 70
sending a RopSubmitMessage ROP request 70
sending a RopTransportNewMail ROP request 70
Sequencing rules - server
receiving a RopAbortSubmit ROP request 75
receiving a RopGetAddressTypes ROP request 75
receiving a RopGetTransportFolder ROP request 76
receiving a RopOptionsData ROP request 77
receiving a RopSetSpooler ROP request 76
receiving a RopSpoolerLockMessage ROP request 76
receiving a RopSubmitMessage ROP request 72
receiving a RopTransportNewMail ROP request 77
receiving a RopTransportSend ROP request 77
Server
abstract data model 70
higher-layer triggered events 72
initialization 72
other local events 77
timer events 77
timers 71
Server - abstract data model
per E-mail object 71
per Message object 71
User ADM data type 71
Server - sequencing rules
receiving a RopAbortSubmit ROP request 75
receiving a RopGetAddressTypes ROP request 75
receiving a RopGetTransportFolder ROP request 76
receiving a RopOptionsData ROP request 77
receiving a RopSetSpooler ROP request 76
receiving a RopSpoolerLockMessage ROP request 76
receiving a RopSubmitMessage ROP request 72
receiving a RopTransportNewMail ROP request 77
receiving a RopTransportSend ROP request 77
Spooler and transport ROPs
RopGetTransportFolder ROP 61
RopSetSpooler ROP 61
RopSpoolerLockMessage ROP 62
RopTransportNewMail ROP 63
RopTransportSend ROP 62
Spooler and Transport ROPs message 61
Standards assignments 18
Submitting a deferred message example
overview 79
ROP request buffer 79
ROP response buffer 79
Submitting a message example
overview 78
ROP request buffer 78
ROP response buffer 78

T

Timer events
client 70
server 77
Timers
client 66
server 71
Tracking changes 95
Transport 19
Triggered events - client
deferring message send 66
delivering mail to the server 68
optimizing send 67
processing E-mail objects in the spooler queue 68
resending a message 67
sending a message 66
sending a message with expiry time 66
sending mail through a specific server 68
soliciting votes from voters 67
Triggered events - higher-layer
server 72

V

Vendor-extensible fields 18
Versioning 18
Voting and tracking overview 17