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.
<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>Revised and edited technical content.</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>Updated IP notice.</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>8.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>3/18/2011</td>
<td>9.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>8/5/2011</td>
<td>10.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>10/7/2011</td>
<td>11.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>1/20/2012</td>
<td>12.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>4/27/2012</td>
<td>13.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>7/16/2012</td>
<td>14.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>10/8/2012</td>
<td>15.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>2/11/2013</td>
<td>16.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>7/26/2013</td>
<td>17.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>11/18/2013</td>
<td>18.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>2/10/2014</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>4/30/2014</td>
<td>19.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>7/31/2014</td>
<td>19.0</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>10/30/2014</td>
<td>19.1</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>3/16/2015</td>
<td>20.0</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>21.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>9/14/2015</td>
<td>21.0</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>21.0</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>21.1</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>9/19/2017</td>
<td>21.1</td>
<td>None</td>
<td>No changes to the meaning, language, or formatting of the technical content.</td>
</tr>
<tr>
<td>7/24/2018</td>
<td>22.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>10/1/2018</td>
<td>23.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>6/18/2019</td>
<td>23.1</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>4/22/2021</td>
<td>24.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>8/17/2021</td>
<td>25.0</td>
<td>Major</td>
<td>Significantly changed the technical content.</td>
</tr>
<tr>
<td>8/15/2023</td>
<td>25.1</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>11/13/2023</td>
<td>25.2</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>2/20/2024</td>
<td>25.3</td>
<td>Minor</td>
<td>Clarified the meaning of the technical content.</td>
</tr>
<tr>
<td>4/16/2024</td>
<td>26.0</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 .................................................................................... 14  
1.2.2 Informative References .................................................................................. 15  
1.3 Overview ............................................................................................................. 16  
1.3.1 Message Objects ............................................................................................ 16  
1.3.2 FAI Messages .................................................................................................. 16  
1.3.3 Message Recipients ......................................................................................... 16  
1.3.4 Message Attachments ..................................................................................... 16  
1.4 Relationship to Other Protocols .......................................................................... 16  
1.5 Prerequisites/Preconditions .................................................................................. 17  
1.6 Applicability Statement ....................................................................................... 17  
1.7 Versioning and Capability Negotiation ............................................................... 17  
1.8 Vendor-Extensible Fields ...................................................................................... 17  
1.9 Standards Assignments ....................................................................................... 17  

2 Messages.................................................................................................................. 18  
2.1 Transport ............................................................................................................ 18  
2.2 Message Syntax .................................................................................................. 18  
2.2.1 Message Object Properties ............................................................................. 18  
2.2.1.1 General Properties ..................................................................................... 18  
2.2.1.2 PidTagHasAttachments Property ............................................................. 18  
2.2.1.3 PidTagMessageClass Property .................................................................. 19  
2.2.1.4 PidTagMessageCodepage Property ........................................................... 19  
2.2.1.5 PidTagMessageLocaleId Property .............................................................. 19  
2.2.1.6 PidTagMessageFlags Property ................................................................. 19  
2.2.1.7 PidTagMessageSize Property ................................................................. 20  
2.2.1.8 PidTagMessageStatus Property ............................................................... 20  
2.2.1.9 PidTagSubjectPrefix Property ................................................................. 21  
2.2.1.10 PidTagNormalizedSubject Property ....................................................... 21  
2.2.1.11 PidTagImportance Property ..................................................................... 21  
2.2.1.12 PidTagPriority Property .......................................................................... 21  
2.2.1.13 PidTagSensitivity Property ...................................................................... 22  
2.2.1.14 PidLidSmartNoAttach Property .............................................................. 22  
2.2.1.15 PidLidPrivate Property .......................................................................... 22  
2.2.1.16 PidLidSideEffects Property .................................................................... 22  
2.2.1.17 PidNameKeywords Property .................................................................. 23  
2.2.1.18 PidLidCommonStart Property .................................................................. 23  
2.2.1.19 PidLidCommonEnd Property .................................................................. 23  
2.2.1.20 PidTagAutoForwarded Property ............................................................. 24  
2.2.1.21 PidTagAutoForwardComment Property ............................................... 24  
2.2.1.22 PidLidCategories Property .................................................................... 24  
2.2.1.23 PidLidClassification .............................................................................. 24  
2.2.1.24 PidLidClassificationDescription Property .......................................... 24  
2.2.1.25 PidLidClassified Property ....................................................................... 24  
2.2.1.26 PidTagInternetReferences Property ....................................................... 24  
2.2.1.27 PidLidInfoPathFormName Property ....................................................... 24  
2.2.1.28 PidTagMimeSkeleton Property ................................................................ 25  
2.2.1.29 PidTagTnefCorrelationKey Property ...................................................... 25  
2.2.1.30 PidTagAddressBookDisplayNamePrintable Property ................................ 25  
2.2.1.31 PidTagCreatorEntryId Property ............................................................ 25  
2.2.1.32 PidTagLastModifierEntryId Property ...................................................... 25  
2.2.1.33 PidLidAgingDoNotAgeMe Property ......................................................... 25  
2.2.1.34 PidLidCurrentVersion Property ............................................................. 26
2.2.1.35  PidLidCurrentVersionName Property .............................................. 26
2.2.1.36  PidTagAlternateRecipientAllowed Property ........................................ 26
2.2.1.37  PidTagResponsibility Property .......................................................... 26
2.2.1.38  PidTagRowid Property ......................................................................... 26
2.2.1.39  PidTagHasNamedProperties Property ...................................................... 26
2.2.1.40  PidTagRecipientOrder Property ........................................................... 26
2.2.1.41  PidNameContentType Property ............................................................. 26
2.2.1.42  PidNameAcceptLanguage Property ...................................................... 27
2.2.1.43  PidTagPurportedSenderDomain Property ............................................. 27
2.2.1.44  PidTagStoreEntryId Property ................................................................ 27
2.2.1.45  PidTagTrustSender .............................................................................. 27
2.2.1.46  PidTagSubject Property ...................................................................... 27
2.2.1.47  PidTagMessageRecipients Property ....................................................... 28
2.2.1.48  PidNameContentClass Property ............................................................ 28
2.2.1.49  PidTagLocalCommitTime Property ......................................................... 28
2.2.1.50  PidNameContentType Property ............................................................. 28
2.2.1.51  PidTagCreatorName Property .................................................................. 28
2.2.1.52  PidTagMessageAttachments Property ................................................... 28
2.2.1.53  PidTagRead Property ........................................................................... 29
2.2.1.54  PidTagRecipientDisplay_Name Property ............................................... 29
2.2.1.55  PidTagRecipientEntryId Property ......................................................... 29
2.2.1.56  PidNameMsiLabels Property .................................................................. 29
2.2.1.57  PidTagClientActiveUntil Property ......................................................... 29
2.2.1.58  Body Properties ..................................................................................... 29
  2.2.1.58.1  PidTagBody Property ....................................................................... 29
  2.2.1.58.2  PidTagNativeBody Property .............................................................. 29
  2.2.1.58.3  PidTagBodyHtml Property .................................................................. 30
  2.2.1.58.4  PidTagRtfCompressed Property .......................................................... 30
  2.2.1.58.5  PidTagRtfInSync Property .................................................................. 30
  2.2.1.58.6  PidTagInternetCodepage Property ..................................................... 30
  2.2.1.58.7  PidTagBodyContentId Property .......................................................... 30
  2.2.1.58.8  PidTagBodyContentLocation Property .............................................. 30
  2.2.1.58.9  PidTagHtml Property ....................................................................... 31
2.2.1.59  Contact Linking Properties .................................................................... 31
  2.2.1.59.1  PidLidContactLinkEntry Property ...................................................... 31
  2.2.1.59.2  PidLidContacts Property ................................................................... 31
  2.2.1.59.3  PidLidContactLinkName Property ...................................................... 32
  2.2.1.59.4  PidLidContactLinkSearchKey Property ............................................. 32
2.2.1.60  Retention and Archive Properties .......................................................... 32
  2.2.1.60.1  PidTagArchiveTag Property ............................................................... 32
  2.2.1.60.2  PidTagPolicyTag Property ................................................................. 32
  2.2.1.60.3  PidTagRetentionPeriod Property ...................................................... 33
  2.2.1.60.4  PidTagStartDateEtc Property ............................................................. 33
  2.2.1.60.5  PidTagRetentionDate Property .......................................................... 33
  2.2.1.60.6  PidTagRetentionFlags Property ........................................................ 34
  2.2.1.60.7  PidTagArchivePeriod Property .......................................................... 34
  2.2.1.60.8  PidTagArchiveDate Property ............................................................. 34
2.2.2  Attachment Object Properties ................................................................... 35
  2.2.2.1  General Properties ................................................................................ 35
  2.2.2.2  PidTagLastModificationTime Property .................................................. 35
  2.2.2.3  PidTagCreationTime Property ................................................................. 35
  2.2.2.4  PidTagDisplayName Property .................................................................. 35
  2.2.2.5  PidTagAttachSize Property ................................................................... 35
  2.2.2.6  PidTagAttachNumber Property ............................................................... 35
  2.2.2.7  PidTagAttachDataBinary Property ........................................................ 36
  2.2.2.8  PidTagAttachDataObject Property ........................................................ 36
  2.2.2.9  PidTagAttachMethod Property ............................................................... 36
  2.2.2.10 PidTagAttachLongFilename Property .................................................... 36
2.2.2.11 PidTagAttachFilename Property .................................................. 37
2.2.2.12 PidTagAttachExtension Property .................................................. 37
2.2.2.13 PidTagAttachLongPathname Property .......................................... 37
2.2.2.14 PidTagAttachPathname Property .................................................. 37
2.2.2.15 PidTagAttachTag Property .......................................................... 37
2.2.2.16 PidTagRenderingPosition Property ................................................. 37
2.2.2.17 PidTagAttachRendering Property .................................................. 38
2.2.2.18 PidTagAttachFlags Property ....................................................... 38
2.2.2.19 PidTagAttachTransportName Property .......................................... 38
2.2.2.20 PidTagAttachEncoding Property .................................................. 38
2.2.2.21 PidTagAttachAdditionalInformation Property .................................. 38
2.2.2.22 PidTagAttachmentLinkId Property ................................................. 39
2.2.2.23 PidTagAttachmentFlags Property ................................................ 39
2.2.2.24 PidTagAttachmentHidden Property .............................................. 39
2.2.2.25 PidTagTextAttachmentCharset Property ....................................... 39
2.2.2.26 PidNameAttachmentProviderType ............................................... 39
2.2.2.27 PidNameAttachmentOriginalPermissionType .................................. 39
2.2.2.28 PidNameAttachmentPermissionType ............................................. 40
2.2.2.29 MIME Properties ......................................................................... 40
2.2.3 Message Object ROPs ................................................................. 41
  2.2.3.1 RopOpenMessage ROP ............................................................. 41
    2.2.3.1.1 RopOpenMessage ROP Request Buffer .................................. 41
    2.2.3.1.2 RopOpenMessage ROP Response Buffer ................................ 42
  2.2.3.2 RopCreateMessage ROP ............................................................. 43
    2.2.3.2.1 RopCreateMessage ROP Request Buffer ................................ 43
    2.2.3.2.2 RopCreateMessage ROP Response Buffer ............................... 43
  2.2.3.3 RopSaveChangesMessage ROP .................................................... 43
    2.2.3.3.1 RopSaveChangesMessage ROP Request Buffer ...................... 44
    2.2.3.3.2 RopSaveChangesMessage ROP Response Buffer .................... 44
  2.2.3.4 RopRemoveAllRecipients ROP ..................................................... 44
    2.2.3.4.1 RopRemoveAllRecipients ROP Request Buffer ...................... 44
    2.2.3.4.2 RopRemoveAllRecipients ROP Response Buffer .................... 45
  2.2.3.5 RopModifyRecipients ROP ........................................................... 45
    2.2.3.5.1 RopModifyRecipients ROP Request Buffer .......................... 45
    2.2.3.5.2 RopModifyRecipients ROP Response Buffer ....................... 46
  2.2.3.6 RopReadRecipients ROP ............................................................. 46
    2.2.3.6.1 RopReadRecipients ROP Request Buffer .......................... 46
    2.2.3.6.2 RopReadRecipients ROP Response Buffer .......................... 46
  2.2.3.7 RopReloadCachedInformation ROP ............................................... 46
    2.2.3.7.1 RopReloadCachedInformation ROP Request Buffer ................ 46
    2.2.3.7.2 RopReloadCachedInformation ROP Response Buffer ............. 47
  2.2.3.8 RopSetMessageStatus ROP .......................................................... 47
    2.2.3.8.1 RopSetMessageStatus ROP Request Buffer .......................... 47
    2.2.3.8.2 RopSetMessageStatus ROP Response Buffer ....................... 47
  2.2.3.9 RopGetMessageStatus ROP .......................................................... 47
    2.2.3.9.1 RopGetMessageStatus ROP Request Buffer ........................ 47
    2.2.3.9.2 RopGetMessageStatus ROP Response Buffer ....................... 48
  2.2.3.10 RopSetReadFlags ROP ............................................................. 48
    2.2.3.10.1 RopSetReadFlags ROP Request Buffer .......................... 48
    2.2.3.10.2 RopSetReadFlags ROP Response Buffer .......................... 49
  2.2.3.11 RopSetMessageReadFlag ROP .................................................... 49
    2.2.3.11.1 RopSetMessageReadFlag ROP Request Buffer .................... 49
    2.2.3.11.2 RopSetMessageReadFlag ROP Response Buffer ................... 49
  2.2.3.12 RopOpenAttachment ROP .......................................................... 50
    2.2.3.12.1 RopOpenAttachment ROP Request Buffer ........................ 50
    2.2.3.12.2 RopOpenAttachment ROP Response Buffer ....................... 50
  2.2.3.13 RopCreateAttachment ROP ........................................................ 50
    2.2.3.13.1 RopCreateAttachment ROP Request Buffer ....................... 50
3 Protocol Details ........................................................................................................54
  3.1 Client Details ...........................................................................................................54
    3.1.1 Abstract Data Model .......................................................................................54
      3.1.1.1 Per Global ...............................................................................................54
      3.1.1.2 Per Mailbox .............................................................................................54
      3.1.1.3 Per Message Object ...............................................................................54
    3.1.2 Timers ...............................................................................................................54
    3.1.3 Initialization .......................................................................................................54
    3.1.4 Higher-Layer Triggered Events .......................................................................55
      3.1.4.1 Opening a Message Object .....................................................................55
      3.1.4.2 Creating a Message Object .....................................................................55
      3.1.4.3 Saving Changes on a Message Object ....................................................55
      3.1.4.4 Removing All Recipients .......................................................................55
      3.1.4.5 Adding, Deleting, or Modifying a Recipient .........................................55
      3.1.4.6 Reading Recipients .................................................................................55
      3.1.4.7 Reload Message Object Header Info .....................................................55
      3.1.4.8 Setting Message Status ............................................................................55
      3.1.4.9 Getting Message Status ..........................................................................55
      3.1.4.10 Setting Message Object Read State .......................................................55
      3.1.4.11 Opening an Attachment .......................................................................56
      3.1.4.12 Creating an Attachment .......................................................................56
      3.1.4.13 Deleting an Attachment .......................................................................56
      3.1.4.14 Setting Attachment Object Content .....................................................56
      3.1.4.15 Saving Changes on an Attachment Object ............................................56
      3.1.4.16 Opening an Embedded Message Object ..............................................56
      3.1.4.17 Accessing the Attachments Table ..........................................................56
      3.1.4.18 Creating an Embedded Message ............................................................56
      3.1.4.19 Saving an Embedded Message ..............................................................56
      3.1.4.20 Linking a Contact Object .....................................................................57
  3.1.5 Message Processing Events and Sequencing Rules ........................................57
      3.1.5.1 Sending a RopOpenMessage ROP Request ........................................57
      3.1.5.2 Sending a RopSaveChangesMessage ROP Request .............................57
      3.1.5.3 Sending a RopCreateMessage ROP Request ........................................57
      3.1.5.4 Sending a RopRemoveAllRecipients ROP Request ...............................58
      3.1.5.5 Sending a RopModifyRecipients ROP Request ......................................58
      3.1.5.6 Sending a RopReadRecipients ROP Request ..........................................58
      3.1.5.7 Sending a RopSetMessageStatus ROP Request ....................................58
      3.1.5.8 Sending a RopGetMessageStatus ROP Request .....................................58
      3.1.5.9 Sending a RopSetReadFlags ROP Request .............................................59
      3.1.5.10 Sending a RopOpenAttachment ROP Request ....................................59
      3.1.5.11 Sending a RopCreateAttachment ROP Request ..................................59

[MS-OXCMMSG] - v20240416
Message and Attachment Object Protocol
Copyright © 2024 Microsoft Corporation
Release: April 16, 2024
3.1.5.12 Sending a RopSetProperties ROP Request ................................... 59
3.1.5.13 Sending a RopGetPropertiesSpecific ROP Request .......................... 60
3.1.5.14 Sending a RopSaveChangesAttachment ROP Request ......................... 60
3.1.5.15 Sending a RopOpenEmbeddedMessage ROP Request .......................... 60
3.1.5.16 Sending a RopGetAttachmentTable ROP Request .............................. 60
3.1.6 Timer Events .................................................................................. 60
3.1.7 Other Local Events ......................................................................... 60
3.2 Server Details ...................................................................................... 60
3.2.1 Abstract Data Model ........................................................................ 60
3.2.1.1 Per Global .................................................................................. 61
3.2.1.2 Per Mailbox ............................................................................... 61
3.2.1.3 Per Message Object .................................................................... 61
3.2.2 Timers ............................................................................................. 61
3.2.3 Initialization .................................................................................... 61
3.2.4 Higher-Layer Triggered Events .......................................................... 61
3.2.4.1 Requesting Body Properties ......................................................... 61
3.2.5 Message Processing Events and Sequencing Rules ................................. 62
3.2.5.1 Receiving a RopOpenMessage ROP Request .................................... 62
3.2.5.2 Receiving a RopCreateMessage ROP Request .................................. 62
3.2.5.3 Receiving a RopSaveChangesMessage ROP Request ........................ 64
3.2.5.4 Receiving a RopRemoveAllRecipients ROP Request ........................... 64
3.2.5.5 Receiving a RopModifyRecipients ROP Request ................................ 65
3.2.5.6 Receiving a RopReadRecipients ROP Request ................................... 66
3.2.5.7 Receiving a RopReloadCachedInformation ROP Request .................... 66
3.2.5.8 Receiving a RopSetMessageStatus ROP Request ............................... 66
3.2.5.9 Receiving a RopGetMessageStatus ROP Request ............................... 67
3.2.5.10 Receiving a RopSetReadFlags ROP Request .................................... 67
3.2.5.11 Receiving a RopSetMessageReadFlag ROP Request .......................... 68
3.2.5.12 Receiving a RopOpenAttachment ROP Request ............................... 68
3.2.5.13 Receiving a RopCreateAttachment ROP Request ............................. 69
3.2.5.14 Receiving a RopSaveChangesAttachment ROP Request .................... 69
3.2.5.15 Receiving a RopDeleteAttachment ROP Request .............................. 70
3.2.5.16 Receiving a RopOpenEmbeddedMessage ROP Request .................... 70
3.2.5.17 Receiving a RopGetAttachmentTable ROP Request .......................... 70
3.2.6 Timer Events .................................................................................... 71
3.2.7 Other Local Events ......................................................................... 71

4 Protocol Examples .................................................................................. 72
4.1 Create Message .................................................................................... 72
4.1.1 RopCreateMessage Request Buffer ............................................... 72
4.1.2 RopCreateMessage Response Buffer .............................................. 72
4.2 Name to Id Mapping ............................................................................ 72
4.3 Get Attachment Table .......................................................................... 72
4.3.1 RopGetAttachmentTable Request Buffer ........................................ 72
4.3.2 RopGetAttachmentTable Response Buffer ....................................... 73
4.4 Insert HTML Embedded Image .............................................................. 73
4.4.1 RopCreateAttachment Request Buffer ......................................... 73
4.4.2 RopCreateAttachment Response Buffer ....................................... 73
4.4.3 Setting Properties ........................................................................... 73
4.4.4 RopSaveChangesAttachment Request Buffer .................................. 74
4.4.5 RopSaveChangesAttachment Response Buffer ............................... 74
4.4.6 Releasing Attachment Object ........................................................... 75
4.5 Attach Text File .................................................................................... 75
4.5.1 RopCreateAttachment Request Buffer ......................................... 75
4.5.2 RopCreateAttachment Response Buffer ....................................... 75
4.5.3 Setting Properties ........................................................................... 75
4.5.4 RopSaveChangesAttachment Request Buffer .................................. 76
4.5.5 RopSaveChangesAttachment Response Buffer ............................... 76
1 Introduction

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

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:

8.3 name: A file name string restricted in length to 12 characters that includes a base name of up to eight characters, one character for a period, and up to three characters for a file name extension. For more information on 8.3 file names, see [MS-CIFS] section 2.2.1.1.

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

Address Book object: An entity in an address book that contains a set of attributes, each attribute with a set of associated values.

archive policy: A feature that determines when items are moved into an alternate mailbox for archival purposes.

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

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.

attachments table: A **Table object** whose rows represent the Attachment objects that are attached to a Message object.

blind carbon copy (Bcc) recipient: An address 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.

category: A subdivision of items into useful groups such as geographical regions. For example, categories that represent geographical regions could be North, South, East, and West.

character set: The range of characters used to represent textual data within a **MIME body part**, as described in [RFC2046].

Classification, labeling, and protection: Classification is determining that an email has sensitive or otherwise interesting content in it. Labeling is tagging the email with an administrator-defined sensitivity label that travels with the email. Protection is enforcing administrator-defined outcomes based on the sensitivity label.
clear-signed body: A message body that was promoted from a clear-signed S/MIME message, as described in [MS-OXOSMIME].

code page: An ordered set of characters of a specific script in which a numerical index (code-point value) is associated with each character. Code pages are a means of providing support for character sets and keyboard layouts used in different countries. Devices such as the display and keyboard can be configured to use a specific code page and to switch from one code page (such as the United States) to another (such as Portugal) at the user's request.

contact: A presence entity (presentity) whose presence information can be tracked.

Contact object: A Message object that contains properties pertaining to a contact.

contents table: A Table object whose rows represent the Message objects that are contained in a Folder object.

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).

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

Draft Message object: A Message object that has not been sent.

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

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

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.

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

**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].

**Inbox folder:** A special folder that is the default location for **Message objects** received by a user or resource.

**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.

**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.

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

**message body:** The content within an HTTP message, as described in [RFC2616] section 4.3.

**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.

**metafile:** A file that stores an image as graphical objects, such as lines, circles, and polygons, instead of pixels. A metafile preserves an image more accurately than pixels when an image is resized.

**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].

**named property:** A property that is identified by both a GUID and either a string name or a 32-bit identifier.

**non-Unicode:** A character set that has a restricted set of glyphs, such as Shift_JIS or ISO-2022-JP.

**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.

**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.

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

**property tag:** A 32-bit value that contains a property type and a property ID. The low-order 16 bits represent the property type. The high-order 16 bits represent the property ID.
public folder: A Folder object that is stored in a location that is publicly available.

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).

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].

restriction: A filter used to map some domain into a subset of itself, by passing only those items from the domain that match the filter. Restrictions can be used to filter existing Table objects or to define new ones, such as search folder or rule criteria.

retention policy: A policy that specifies the length of time during which data, documents, and other records must be available for recovery.

retention tag: An element that contains information about the retention policy of a Message object or folder.

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

soft delete: A process that removes an item from the system, but not permanently. If an item is soft deleted, a server retains a back-up copy of the item and a client can access, restore, or permanently delete the item. See also hard delete.

Store object: An object that is used to store mailboxes and public folder content.

table object: A group of shapes that are arranged in rows and columns to form a table.

To recipient: See primary recipient.

transaction: The process of opening or creating an object on a server, and the subsequent committing of changes to the object by calling the required save function, at which time all changes to that instance of the object are either saved to the server, or discarded if a failure occurs before saving is finished successfully. Until successfully saved, changes are invisible to any other instances of the object.
**Transport Neutral Encapsulation Format (TNEF):** A binary type-length-value encoding that is used to encode properties for transport, as described in [MS-OXTNEF].

**undefined body:** A body with no defined content.

**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].

**Web Distributed Authoring and Versioning Protocol (WebDAV):** The Web Distributed Authoring and Versioning Protocol, as described in [RFC2518] or [RFC4918].

**Windows metafile format (WMF):** A file format used by Windows that supports the definition of images, including a format for clip art in word-processing documents.

**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-LCID] Microsoft Corporation, "Windows Language Code Identifier (LCID) Reference".

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

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

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

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

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


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


[MS-OXCSTOR] Microsoft Corporation, "Store Object Protocol".
 Informative References

[MS-OXCFOLD] Microsoft Corporation, "Folder Object Protocol".
[MS-OXOCNTC] Microsoft Corporation, "Contact Object Protocol".
[MS-OXOCAL] Microsoft Corporation, "Appointment and Meeting Object Protocol".
[MS-OXOCFG] Microsoft Corporation, "Configuration Information Protocol".
[MS-OXOMSG] Microsoft Corporation, "Email Object Protocol".
[MS-OXOSFLD] Microsoft Corporation, "Special Folders Protocol".
[MS-OXTNEF] Microsoft Corporation, "Transport Neutral Encapsulation Format (TNEF) Data Algorithm".
[MS-WMF] Microsoft Corporation, "Windows Metafile Format".


[RFC2110] Palme, J., and Hopmann, A., "MIME E-mail Encapsulation of Aggregate Documents, such as HTML (MHTML)", RFC 2110, March 1997, http://www.rfc-editor.org/rfc/rfc2110.txt


1.2.2 Informative References

[MS-OXOCAL] Microsoft Corporation, "Appointment and Meeting Object Protocol".
[MS-OXOCFG] Microsoft Corporation, "Configuration Information Protocol".
[MS-OXOMSG] Microsoft Corporation, "Email Object Protocol".
[MS-OXOSFLD] Microsoft Corporation, "Special Folders Protocol".
[MS-OXTNEF] Microsoft Corporation, "Transport Neutral Encapsulation Format (TNEF) Data Algorithm".
[MS-WMF] Microsoft Corporation, "Windows Metafile Format".


[RFC2110] Palme, J., and Hopmann, A., "MIME E-mail Encapsulation of Aggregate Documents, such as HTML (MHTML)", RFC 2110, March 1997, http://www.rfc-editor.org/rfc/rfc2110.txt


1.3 Overview

The Message and Attachment Object Protocol consists of a set of properties and remote operation (ROP) procedure calls for adding, modifying, and deleting Message objects, folder associated information (FAI) messages, and Attachment objects associated with messages.

1.3.1 Message Objects

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

1.3.2 FAI Messages

FAI messages contain auxiliary data needed by the client or server. FAI messages are persisted in the same way as Message objects, but cannot be sent.

1.3.3 Message Recipients

Message objects allow clients to associate one or more recipients (2) to a message.

1.3.4 Message Attachments

An Attachment object is used by a client to associate files, Object Linking and Embedding (OLE) objects, other messages, or binary data with a particular Message object. Because Attachment objects are created, maintained, and accessed only in the context of a message, they are considered subobjects. Operations that affect the location of a Message object also apply to its attachments. Clients retrieve information about attachments in a message via an attachments table, which is a Table object, as described in [MS-OXCTABL].

1.4 Relationship to Other Protocols

The Message and Attachment Object Protocol relies on folders, tables, and properties, as described in [MS-OXCFOLD], [MS-OXOSFLD], [MS-OXCTABL], and [MS-OXCPRTT], as well as the underlying ROPs transport, described in [MS-OXCROPS].

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

- Appointment and Meeting Object Protocol, as described in [MS-OXOCAL]
- Contact Object Protocol, as described in [MS-OXOCNTC]
- Email Object Protocol, as described in [MS-OXOMSG]
- Task–Related Objects Protocol, as described in [MS-OXOTASK]
- Note Object Protocol, as described in [MS-OXONOTE]
1.5 Prerequisites/Preconditions

The Message and Attachment Object Protocol assumes the client has previously logged on to the server and has acquired a handle to the Folder object upon which it needs to operate. For more information on Folder objects, see [MS-OXCFCFOLD]. For more information on folder storage and organization, see [MS-OXCFSTOR].

1.6 Applicability Statement

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

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

A third-party application can create its own set of named properties on a Message object as described in [MS-OXCPRPT]. A third-party application can also extend the Message and Attachment Object Protocol to implement its own object type by changing the PidTagMessageClass property (section 2.2.1.3). For a simple example that extends this protocol to implement an electronic representation of a "Sticky Note", see [MS-OXONOTE].

1.9 Standards Assignments

None.
2 Messages

2.1 Transport

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

2.2 Message Syntax

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

Clients operate on Message objects using the ROPs as specified in section 2.2.3, and the Property and Stream Object Protocol, as specified in [MS-OXCPRPT] section 2.2.

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

When a property is referred to as "read-only for the client", the server can return Success but not persist any changes to read-only properties.\(^1\) Read-only properties that could not be saved are enumerated in the PropertyProblems field of the RopSetProperties ROP response, as specified in [MS-OXCROPS] section 2.2.8.6.2.

2.2.1 Message Object Properties

2.2.1.1 General Properties

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

- **PidTagAccess** ([MS-OXCPRPT] section 2.2.1.1)
- **PidTagAccessLevel** ([MS-OXCPRPT] section 2.2.1.2)
- **PidTagChangeKey** ([MS-OXCFXICS] section 2.2.1.2.7)
- **PidTagCreationTime** (section 2.2.2.3)
- **PidTagLastModificationTime** (section 2.2.2)
- **PidTagLastModifierName** ([MS-OXCPRPT] section 2.2.1.5)
- **PidTagObjectType** \(^2\) ([MS-OXCPRPT] section 2.2.1.7)
- **PidTagRecordKey** \(^3\) ([MS-OXCPRPT] section 2.2.1.8)
- **PidTagSearchKey** ([MS-OXCPRPT] section 2.2.1.9)

2.2.1.2 PidTagHasAttachments Property

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

The PidTagHasAttachments property ([MS-OXPROPS] section 2.717) indicates whether the Message object contains at least one attachment. This property is read-only for the client.
The server computes this property from the \texttt{mfHasAttach} flag of the \texttt{PidTagMessageFlags} property ([MS-OXPROPS] section 2.792).

\subsection*{2.2.1.3 PidTagMessageClass Property}

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

The \texttt{PidTagMessageClass} property ([MS-OXPROPS] section 2.788) denotes the specific type of the \textit{Message object}. It determines the set of properties defined for the message, the kind of information the message conveys, and how to handle the message.

All characters in this property MUST be from the ASCII characters \texttt{0x20} through \texttt{0x7F}. It MUST NOT end with a period (ASCII character \texttt{0x2E}), and its length MUST be greater than zero and less than 256 characters. Furthermore, its length SHOULD be fewer than 128 characters because some operations require extending the value of the \texttt{PidTagMessageClass} property.

Any equality or matching operations performed against the value of this property MUST be case-insensitive.

The value of this property is interpreted in groups of characters separated by periods (\texttt{"."}). Each group specifies a type of object. A \textit{message class} of "IPM.Note" denotes a standard \textit{Message object}, and a message class of "Remote.IPM.Note" indicates a \textit{header message object}.

\subsection*{2.2.1.4 PidTagMessageCodepage Property}

Type: \texttt{PtypInteger32}, unsigned

The \texttt{PidTagMessageCodepage} property ([MS-OXPROPS] section 2.789) specifies the \textit{code page} used to encode the non-Unicode string properties on this \textit{Message object}. The \textit{Folder object} code page is used if this property is set to \texttt{0x00000000}.

\subsection*{2.2.1.5 PidTagMessageLocaleId Property}

Type: \texttt{PtypInteger32}, unsigned

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

\subsection*{2.2.1.6 PidTagMessageFlags Property}

Type: \texttt{PtypInteger32} ([MS-OXCDATA] section 2.11.1)

The \texttt{PidTagMessageFlags} property ([MS-OXPROPS] section 2.792) specifies the status of the \textit{Message object}. Set to zero or to a bitwise OR of one or more of the values from the following tables.

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

<table>
<thead>
<tr>
<th>Flag name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mfRead</td>
<td>0x00000001</td>
<td>The message is marked as having been read.</td>
</tr>
<tr>
<td>mfUnsent</td>
<td>0x00000008</td>
<td>The message is still being composed and is treated as a \textit{Draft Message object}. This bit is cleared by the server when responding to the \texttt{RopSubmitMessage} ROP ([MS-OXCROPS] section 2.2.7.1) with a success code.</td>
</tr>
<tr>
<td>Flag name</td>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>mfResend</td>
<td>0x00000080</td>
<td>The message includes a request for a resend operation with a non-delivery report. For more details, see [MS-OXOMSG] section 3.2.4.5.</td>
</tr>
</tbody>
</table>

These flags are always read-only for the client.

<table>
<thead>
<tr>
<th>Flag name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mfUnmodified</td>
<td>0x00000002</td>
<td>The message has not been modified since it was first saved (if unsent) or it was delivered (if sent).</td>
</tr>
<tr>
<td>mfSubmitted</td>
<td>0x00000004</td>
<td>The message is marked for sending as a result of a call to the RopSubmitMessage ROP</td>
</tr>
<tr>
<td>mfHasAttach</td>
<td>0x00000010</td>
<td>The message has at least one attachment. This flag corresponds to the message’s PidTagHasAttachments property (section 2.2.1.2).</td>
</tr>
<tr>
<td>mfFromMe</td>
<td>0x00000020</td>
<td>The user receiving the message was also the user who sent the message.</td>
</tr>
<tr>
<td>mfFAI</td>
<td>0x00000040</td>
<td>The message is an FAI message.</td>
</tr>
<tr>
<td>mfNotifyRead</td>
<td>0x00000100</td>
<td>The user who sent the message has requested notification when a recipient (1) first reads it.</td>
</tr>
<tr>
<td>mfNotifyUnread</td>
<td>0x00000200</td>
<td>The user who sent the message has requested notification when a recipient (1) deletes it before reading or the Message object expires.</td>
</tr>
<tr>
<td>mfEverRead</td>
<td>0x00000400</td>
<td>The message has been read at least once. This flag is set or cleared by the server whenever the mfRead flag is set or cleared. Clients SHOULD ignore this flag.</td>
</tr>
<tr>
<td>mfInternet</td>
<td>0x00002000</td>
<td>The incoming message arrived over the Internet and originated either outside the organization or from a source the gateway does not consider trusted.</td>
</tr>
<tr>
<td>mfUntrusted</td>
<td>0x00008000</td>
<td>The incoming message arrived over an external link other than X.400 or the Internet. It originated either outside the organization or from a source the gateway does not consider trusted.</td>
</tr>
</tbody>
</table>

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

### 2.2.1.7 PidTagMessageSize Property

Type: PtypInteger32, unsigned

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

### 2.2.1.8 PidTagMessageStatus Property

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

Specifies the status of a message in a contents table. Contains a bitwise OR of zero or more of the following values.
<table>
<thead>
<tr>
<th>Flag name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>msRemoteDownload</td>
<td>0x00001000</td>
<td>The message has been marked for downloading from the remote message store to the local client.</td>
</tr>
<tr>
<td>msInConflict</td>
<td>0x00000800</td>
<td>This is a conflict resolve message as specified in [MS-OXCFXICS] section 3.1.5.6.2.1. This is a read-only value for the client.</td>
</tr>
<tr>
<td>msRemoteDelete</td>
<td>0x00002000</td>
<td>The message has been marked for deletion at the remote message store without downloading to the local client.</td>
</tr>
</tbody>
</table>

More details about setting and obtaining the value of this property are specified in section 3.1.5.7 and section 3.1.5.8.

### 2.2.1.9 PidTagSubjectPrefix Property

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

The PidTagSubjectPrefix property ([MS-OXPROPS] section 2.1035) contains the prefix for the subject of the message. This property is set by the client but can be an empty string if there is no subject. The sum of the lengths of the PidTagNormalizedSubject property (section 2.2.1.10) and the PidTagSubjectPrefix property MUST be less than 254 characters.

More details about the processing of this property are specified in section 3.1.5.13.

### 2.2.1.10 PidTagNormalizedSubject Property

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

The PidTagNormalizedSubject property ([MS-OXPROPS] section 2.813) contains the normalized subject of the message, as specified in [MS-OXCMAIL] section 2.2.3.2.6.1. This property is set by the client but can be an empty string if there is no subject. The sum of the lengths of the PidTagNormalizedSubject property and the PidTagSubjectPrefix property (section 2.2.1.9) MUST be less than 254 characters.

More details about obtaining the value of this property are specified in section 3.1.5.13.

### 2.2.1.11 PidTagImportance Property

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

The PidTagImportance property ([MS-OXPROPS] section 2.739) indicates the level of importance assigned by the end user to the Message object. This property MUST be set to one of the following values.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000000</td>
<td>Low importance.</td>
</tr>
<tr>
<td>0x00000001</td>
<td>Normal importance.</td>
</tr>
<tr>
<td>0x00000002</td>
<td>High importance.</td>
</tr>
</tbody>
</table>

### 2.2.1.12 PidTagPriority Property

Type: PtypInteger32 ([MS-OXCDATA] section 2.11.1)
The **PidTagPriority** property ([MS-OXPROPS] section 2.872) indicates the client's request for the priority at which the message is to be sent by the messaging system. This property is set to one of the following values.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000001</td>
<td>Urgent</td>
</tr>
<tr>
<td>0x00000000</td>
<td>Normal</td>
</tr>
<tr>
<td>0xFFFFFFFF</td>
<td>Not urgent</td>
</tr>
</tbody>
</table>

### 2.2.1.13 PidTagSensitivity Property

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

The **PidTagSensitivity** property ([MS-OXPROPS] section 2.1011) indicates the sender's assessment of the sensitivity of the **Message object**. The value of this property is one of the following.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000000</td>
<td>Normal</td>
</tr>
<tr>
<td>0x00000001</td>
<td>Personal</td>
</tr>
<tr>
<td>0x00000002</td>
<td>Private</td>
</tr>
<tr>
<td>0x00000003</td>
<td>Confidential</td>
</tr>
</tbody>
</table>

### 2.2.1.14 PidLidSmartNoAttach Property

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

The **PidLidSmartNoAttach** property ([MS-OXPROPS] section 2.301) is set to TRUE (0x01) if the **Message object** has no attachments that are visible to the end user. If this property is unset, a default value of FALSE (0x00) is used.

### 2.2.1.15 PidLidPrivate Property

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

The **PidLidPrivate** property ([MS-OXPROPS] section 2.211) is set to TRUE (0x01) if the end user wants this **Message object** to be hidden from other users who have access to the **Message object**.

### 2.2.1.16 PidLidSideEffects Property

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

The **PidLidSideEffects** property ([MS-OXPROPS] section 2.299) controls how a **Message object** is handled by the client in relation to certain user interface actions by the user, such as deleting a message. This property is set to a bitwise OR of zero or more of the following **flags**.
<table>
<thead>
<tr>
<th>Flag name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>seOpenToDelete</code></td>
<td>0x00000001</td>
<td>The client opens the Message object when deleting.</td>
</tr>
<tr>
<td><code>seNoFrame</code></td>
<td>0x00000008</td>
<td>No UI is associated with the Message object.</td>
</tr>
<tr>
<td><code>seCoerceToInbox</code></td>
<td>0x00000010</td>
<td>The client moves the Message object to the Inbox folder when moving or</td>
</tr>
<tr>
<td></td>
<td></td>
<td>copying to a Folder object with the PidTagContainerClass property.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>([MS-OXOCAL] section 2.2.11.1) set to “IPF.Note”. For more details about</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the PidTagContainerClass property, see [MS-OXOSFLD] section 2.2.8.</td>
</tr>
<tr>
<td><code>seOpenToCopy</code></td>
<td>0x00000020</td>
<td>The client opens the Message object when copying to another folder.</td>
</tr>
<tr>
<td><code>seOpenToMove</code></td>
<td>0x00000040</td>
<td>The client opens the Message object when moving to another folder.</td>
</tr>
<tr>
<td><code>seOpenForCtxMenu</code></td>
<td>0x00000100</td>
<td>The client opens the Message object when displaying context-sensitive</td>
</tr>
<tr>
<td></td>
<td></td>
<td>commands, such as a context menu, to the end user.</td>
</tr>
<tr>
<td><code>seCannotUndoDelete</code></td>
<td>0x00000400</td>
<td>The client cannot undo a delete operation; this flag MUST NOT be set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>unless the <code>seOpenToDelete</code> flag is set.</td>
</tr>
<tr>
<td><code>seCannotUndoCopy</code></td>
<td>0x00000800</td>
<td>The client cannot undo a copy operation; this flag MUST NOT be set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>unless the <code>seOpenToCopy</code> flag is set.</td>
</tr>
<tr>
<td><code>seCannotUndoMove</code></td>
<td>0x00001000</td>
<td>The client cannot undo a move operation; this flag MUST NOT be set</td>
</tr>
<tr>
<td></td>
<td></td>
<td>unless the <code>seOpenToMove</code> flag is set.</td>
</tr>
<tr>
<td><code>seHasScript</code></td>
<td>0x00002000</td>
<td>The Message object contains end-user script.</td>
</tr>
<tr>
<td><code>seOpenToPermDelete</code></td>
<td>0x00004000</td>
<td>The client opens the Message object to permanently delete it.</td>
</tr>
</tbody>
</table>

### 2.2.1.17 PidNameKeywords Property

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

The `PidNameKeywords` property ([MS-OXPROPS] section 2.451) contains keywords or categories for the Message object. The length of each string within the multivalue string is less than 256 characters.

### 2.2.1.18 PidLidCommonStart Property

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

The `PidLidCommonStart` property ([MS-OXPROPS] section 2.63) indicates the start time for the Message object. The value of this property is less than or equal to the value of the `PidLidCommonEnd` property (section 2.2.1.19). This time is interpreted as Coordinated Universal Time (UTC).

### 2.2.1.19 PidLidCommonEnd Property

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

The `PidLidCommonEnd` property ([MS-OXPROPS] section 2.62) indicates the end time for the Message object. The value of this property MUST be greater than or equal to the value of the `PidLidCommonStart` property (section 2.2.1.18). This time is interpreted as UTC.
2.2.1.20  **PidTagAutoForwarded Property**

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

The **PidTagAutoForwarded** property ([MS-OXPROPS] section 2.614) indicates that this message has been automatically generated or automatically forwarded. If this property is unset, a default value of 0x00 is assumed.

2.2.1.21  **PidTagAutoForwardComment Property**

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

The **PidTagAutoForwardComment** property ([MS-OXPROPS] section 2.613) contains a comment added by the autoforwarding agent.

2.2.1.22  **PidLidCategories Property**

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

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

2.2.1.23  **PidLidClassification**

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

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

2.2.1.24  **PidLidClassificationDescription Property**

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

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

2.2.1.25  **PidLidClassified Property**

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

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

2.2.1.26  **PidTagInternetReferences Property**

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

The **PidTagInternetReferences** property ([MS-OXPROPS] section 2.750) contains a list of message IDs (MIDs), as specified in [MS-OXCDATA] section 2.2.1.2, that specify the messages to which this reply is related. The format of this property is specified in [RFC2822].

2.2.1.27  **PidLidInfoPathFormControlName Property**

Type: **PtypString** ([MS-OXCDATA] section 2.11.1)
The **PidLidInfoPathFormName** property ([MS-OXPROPS] section 2.149) contains the name of the form associated with this message, if one exists. The relationship between this property and the **Content-Class MIME header** is specified in [MS-OXCMAIL] sections 2.1.3.2.2 and 2.2.3.2.15.

### 2.2.1.28 PidTagMimeSkeleton Property

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

The **PidTagMimeSkeleton** property ([MS-OXPROPS] section 2.804) contains all the top level **MIME** message headers, all MIME message body part headers, and body part content that is not already converted to Message object properties, including attachments.

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

### 2.2.1.29 PidTagTnefCorrelationKey Property

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

The **PidTagTnefCorrelationKey** property ([MS-OXPROPS] section 2.1048) contains a value that correlates a **Transport Neutral Encapsulation Format (TNEF)** attachment with a message. This property determines whether or not an inbound TNEF file belongs to the message it is attached to. It is used primarily by transport providers and gateways.

### 2.2.1.30 PidTagAddressBookDisplayNamePrintable Property

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

The **PidTagAddressBookDisplayNamePrintable** property ([MS-OXPROPS] section 2.514) contains the printable string version of the display name.

### 2.2.1.31 PidTagCreatorEntryId Property

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

The **PidTagCreatorEntryId** property ([MS-OXPROPS] section 2.656) specifies the original author of the message according to their address book EntryID. The format of an address book EntryID data type is specified in [MS-OXCDATA] section 2.2.5.2.

### 2.2.1.32 PidTagLastModifierEntryId Property

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

The **PidTagLastModifierEntryId** property ([MS-OXPROPS] section 2.766) specifies the last user to modify the contents of the message according to their address book EntryID. The format of an address book EntryID data type is specified in [MS-OXCDATA] section 2.2.5.2.

### 2.2.1.33 PidLidAgingDontAgeMe Property

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

The **PidLidAgingDontAgeMe** property ([MS-OXPROPS] section 2.4) specifies whether the message is to be automatically archived. This property is set to "TRUE" if the message will not be automatically archived; otherwise, "FALSE".
2.2.1.34  **PidLidCurrentVersion Property**
Type:  PtypInteger32 ([MS-OXCDATA] section 2.11.1)

The **PidLidCurrentVersion** property ([MS-OXPROPS] section 2.88) specifies the build number of the client application that sent the message.

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

The **PidLidCurrentVersionName** property ([MS-OXPROPS] section 2.89) specifies the name of the client application that sent the message.

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

The **PidTagAlternateRecipientAllowed** property ([MS-OXPROPS] section 2.577) specifies whether the sender permits the message to be autoforwarded. This property is set to "TRUE" if autoforwarding is allowed.

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

The **PidTagResponsibility** property ([MS-OXPROPS] section 2.932) specifies whether another mail agent has ensured that the message will be delivered. This property is set to "TRUE" if another agent has accepted responsibility; otherwise, "FALSE".

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

The **PidTagRowid** property ([MS-OXPROPS] section 2.940) contains a unique identifier for a recipient (2) in the message's recipient table. This is a temporary identifier that is valid only for the life of the **Table object**.

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

The **PidTagHasNamedProperties** property ([MS-OXPROPS] section 2.719) specifies whether this **Message object** supports named properties.

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

The **PidTagRecipientOrder** property ([MS-OXPROPS] section 2.903) specifies the location of the current recipient (2) in the recipient table.

2.2.1.41  **PidNameContentBase Property**
Type:  PtypString ([MS-OXCDATA] section 2.11.1)
The `PidNameContentBase` property ([MS-OXPROPS] section 2.412) specifies the value of the `Content-Base` header, which defines the base Uniform Resource Identifier (URI) for resolving relative URLs contained within the message body.

### 2.2.1.42 PidNameAcceptLanguage Property

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

The `PidNameAcceptLanguage` property ([MS-OXPROPS] section 2.366) contains the value of the `Accept-Language` header, which defines the natural languages in which the sender prefers to receive a response. The format of this property is specified in [RFC3282] section 3. The relationship between this property and the `Accept-Language` header is specified in [MS-OXCMAIL] sections 2.1.3.2.16 and 2.2.3.2.11.

### 2.2.1.43 PidTagPurportedSenderDomain Property

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

The `PidTagPurportedSenderDomain` property ([MS-OXPROPS] section 2.877) contains the domain name of the last sender responsible for transmitting the current message.

### 2.2.1.44 PidTagStoreEntryId Property

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

The `PidTagStoreEntryId` property ([MS-OXPROPS] section 2.1029) contains the unique `EntryID` of the message store where an object resides. The format of this property is specified in [MS-OXCDATA] section 2.2.4.

### 2.2.1.45 PidTagTrustSender

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

The `PidTagTrustSender` property ([MS-OXPROPS] section 2.1052) specifies whether the message was delivered through a trusted transport channel.

<5> This property is a Boolean integer. Valid values are given in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000000</td>
<td>Message was not delivered through a trusted transport channel.</td>
</tr>
<tr>
<td>0x00000001</td>
<td>Message was delivered through a trusted transport channel.</td>
</tr>
</tbody>
</table>

### 2.2.1.46 PidTagSubject Property

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

The `PidTagSubject` property ([MS-OXPROPS] section 2.1034) contains the full subject of an e-mail message. The full subject is a concatenation of the subject prefix, as identified by the `PidTagSubjectPrefix` property (section 2.2.1.9), and the normalized subject, as identified by the `PidTagNormalizedSubject` property (section 2.2.1.10). If the `PidTagSubjectPrefix` property is not set or is set to an empty string, then the values of the `PidTagSubject` and `PidTagNormalizedSubject` properties are equal.
2.2.1.47  PidTagMessageRecipients Property

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

The PidTagMessageRecipients property ([MS-OXPROPS] section 2.796) contains a restriction table that can be used to find all messages containing the recipients (2) in a contents table. The restriction is a SubObjectRestriction structure, as specified in [MS-OXCDATA] section 2.12.10. Messages can be searched with this restriction using the RopSetSearchCriteria ROP ([MS-OXCROPS] section 2.2.4.4) and the RopRestrict ROP ([MS-OXCROPS] section 2.2.5.3).

2.2.1.48  PidNameContentClass Property

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

The PidNameContentClass property ([MS-OXPROPS] section 2.413) contains a string that identifies the type of content of a Message object. The value of this property is as follows:

- For an E-mail object, the value is as specified in [MS-OXCMAIL] section 2.2.3.2.15.
- For other Message objects, the value is as specified in the technical document that defines the particular Message object.
- For the Web Distributed Authoring and Versioning Protocol (WebDAV), the value is as specified in the technical document that specifies the particular WebDAV extension.

2.2.1.49  PidTagLocalCommitTime Property

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

The PidTagLocalCommitTime property ([MS-OXPROPS] section 2.773) specifies the time, in UTC, that the message was last changed. The changes include any change to the read state of the message.

2.2.1.50  PidNameContentType Property

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

The PidNameContentType property ([MS-OXPROPS] section 2.414) contains the value of the Content-Type header, which defines the type of the body part's content. For details about the Content-Type header, see [MS-OXCMAIL].

2.2.1.51  PidTagCreatorName Property

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

The PidTagCreatorName property ([MS-OXPROPS] section 2.657) specifies the name of the Message object's creator.

2.2.1.52  PidTagMessageAttachments Property

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

The PidTagMessageAttachments property ([MS-OXPROPS] section 2.786) contains identifiers that correspond to a message's attachments.
2.2.1.53  
**PidTagRead Property**

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

The **PidTagRead** property ([MS-OXPROPS] section 2.879) indicates whether a message has been read.

2.2.1.54  
**PidTagRecipientDisplayName Property**

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

The **PidTagRecipientDisplayName** property ([MS-OXPROPS] section 2.900) specifies the display name of a recipient (2).

2.2.1.55  
**PidTagRecipientEntryId Property**

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

The **PidTagRecipientEntryId** property ([MS-OXPROPS] section 2.901) contains an **EntryID** that identifies the **Address Book object** associated with a recipient (2).

2.2.1.56  
**PidNameMSIPLabels Property**

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

The **PidNameMSIPLabels** property ([MS-OXPROPS] section 2.460) contains the string that specifies the **CLP Labels** information.

2.2.1.57  
**PidTagClientActivelyEditingUntil Property**

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

The **PidTagClientActivelyEditingUntil** property ([MS-OXPROPS] section 2.635) specifies the date and time, in **UTC**, until which time the client expects to be actively editing the object.

2.2.1.58  
**Body Properties**

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

2.2.1.58.1  
**PidTagBody Property**

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

The **PidTagBody** property ([MS-OXPROPS] section 2.618) contains unformatted text, which is the text/plain MIME format as specified in [RFC1521] section 7.1.2. Processing of the plain text body from the MIME message format is specified in [MS-OXCMAIL] section 2.1.3.3.1.

2.2.1.58.2  
**PidTagNativeBody Property**

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

The **PidTagNativeBody** property ([MS-OXPROPS] section 2.806) indicates the best available format for storing the message body<6>. The value of this property is limited to one of the property values shown in the following table.
<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00000000</td>
<td>Undefined body</td>
</tr>
<tr>
<td>0x00000001</td>
<td>Plain text body</td>
</tr>
<tr>
<td>0x00000002</td>
<td>Rich Text Format (RTF)</td>
</tr>
<tr>
<td>0x00000003</td>
<td>HTML body</td>
</tr>
<tr>
<td>0x00000004</td>
<td>Clear-signed body</td>
</tr>
</tbody>
</table>

2.2.1.58.3 PidTagBodyHtml Property

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

The PidTagBodyHtml property ([MS-OXPROPS] section 2.621) contains the HTML body as specified in [RFC2822] section 2.3.

2.2.1.58.4 PidTagRtfCompressed Property

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

The PidTagRtfCompressed property ([MS-OXPROPS] section 2.942) contains an RTF body compressed as specified in [MS-OXRTFCP].

2.2.1.58.5 PidTagRtfInSync Property

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

The PidTagRtfInSync property ([MS-OXPROPS] section 2.943) is set to "TRUE" (0x01) if the RTF body has been synchronized with the contents in the PidTagBody property (section 2.2.1.58.1).

2.2.1.58.6 PidTagInternetCodepage Property

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

The PidTagInternetCodepage property ([MS-OXPROPS] section 2.747) indicates the code page used for the PidTagBody property (section 2.2.1.58.1) or the PidTagBodyHtml property (section 2.2.1.58.3).

2.2.1.58.7 PidTagBodyContentId Property

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

The PidTagBodyContentId property ([MS-OXPROPS] section 2.619) contains a GUID corresponding to the current message body.

This property corresponds to the Content-ID header. The relationship between this property and the Content-ID header is further specified in [MS-OXCMAIL] sections 2.1.3.4.2.3 and 2.2.3.2.24.

2.2.1.58.8 PidTagBodyContentLocation Property

Type: PtypString ([MS-OXCDATA] section 2.11.1)
The `PidTagBodyContentLocation` property ([MS-OXPROPS] section 2.620) contains a globally unique URI that serves as a label for the current message body. The URI can be either absolute or relative.

This property corresponds to the Content-Location header. The relationship between this property and the Content-Location header is further specified in [MS-OXCMAIL] sections 2.1.3.4.2.3 and 2.2.3.2.26. The format of this property is further specified in [RFC2110].

### 2.2.1.58.9 PidTagHtml Property

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

The `PidTagHtml` property ([MS-OXPROPS] section 2.734) contains the message body text in HTML format.

### 2.2.1.59 Contact Linking Properties

Contact linking properties are a group of related properties that are valid on any Message object containing information about the linked Contact objects.

#### 2.2.1.59.1 PidLidContactLinkEntry Property

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

The `PidLidContactLinkEntry` property ([MS-OXPROPS] section 2.70) contains the list of address book EntryIDs linked to by this Message object.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3 | 0 | 1 |
| AddrBookEntryIdCount |
| FieldSize |
| AddrBookEntryIdData (variable) |
| ... |
| Padding |

**AddrBookEntryIdCount (4 bytes):** The number of address book EntryIDs included in this property.

**FieldSize (4 bytes):** The size of the AddrBookEntryIdCount field, minus 4.

**AddrBookEntryIdData (variable):** The EntryID data. Repeated the number of times specified by the AddrBookEntryIdCount field. Address book EntryIDs are specified in [MS-OXCDATA] section 2.2.5.2.

**Padding (3 bytes):** Between 0 and 3 bytes of padding, up to the amount required to make the FieldSize field a multiple of 4. The value of each padded byte MUST be 0x0000.

#### 2.2.1.59.2 PidLidContacts Property

Type: PtypMultipleString ([MS-OXCDATA] section 2.11.1)
The `PidLidContacts` property ([MS-OXPROPS] section 2.77) contains the `PidTagDisplayName` property ([MS-OXOABK] section 2.2.3.1) of each address book EntryID referenced in the value of the `PidLidContactLinkEntry` property (section 2.2.1.59.1). This property can also include names not referenced in the `PidLidContactLinkEntry` property.

### 2.2.1.59.3 PidLidContactLinkName Property

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

The `PidLidContactLinkName` property ([MS-OXPROPS] section 2.74) contains the elements of the `PidLidContacts` property (section 2.2.1.59.2), separated by a semicolon and a space ("; ").

### 2.2.1.59.4 PidLidContactLinkSearchKey Property

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

The `PidLidContactLinkSearchKey` property ([MS-OXPROPS] section 2.75) contains the list of search keys for the Contact object linked to by this Message object. Search keys are used to find related objects. Search keys for address book data are further specified by the `PidTagSearchKey` property ([MS-OXCPRPT] section 2.2.1.9).

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3 | 0 | 1 |
| SearchKeyCount | SearchKeyData (variable) |

**SearchKeyCount (2 bytes)**: The number of search keys included in this property.

**SearchKeyData (variable)**: The search keys, up to the number indicated by the `ContactEntryCount` field. Search keys might be a variable size and are null-terminated, as specified in [MS-OXOABK] section 2.2.3.5.

### 2.2.1.60 Retention and Archive Properties

Retention and archive properties specify information about the retention policy or archive policy. These properties are valid on e-mail Message objects. Some of these properties are also valid on folders. The retention policy and the archive policy are independent features. The server can enable one of these policies, both of these policies, or neither of them. For details about how the retention policy and archive policy settings are communicated between client and server, see [MS-OXOCFG] section 2.2.5.2.3.

#### 2.2.1.60.1 PidTagArchiveTag Property

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

The `PidTagArchiveTag` property ([MS-OXPROPS] section 2.581) specifies the GUID of an archive tag. The `PidTagArchiveTag` property can be present on both Message objects and folders and can be set by both client and server.

#### 2.2.1.60.2 PidTagPolicyTag Property

**Type**: `PtypBinary` ([MS-OXCDATA] section 2.11.1)
The PidTagPolicyTag property ([MS-OXPROPS] section 2.864) specifies the GUID of a retention tag. The PidTagPolicyTag property can be present on both Message objects and folders and can be set by both client and server.

2.2.1.60.3 PidTagRetentionPeriod Property

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

The PidTagRetentionPeriod property ([MS-OXPROPS] section 2.935) specifies the number of days that a Message object can be retained. The PidTagRetentionPeriod property can be present on both Message objects and folders and can be set by both client and server.

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

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

2.2.1.60.4 PidTagStartDateEtc Property

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

The PidTagStartDateEtc property ([MS-OXPROPS] section 2.1027) has the following structure.

<table>
<thead>
<tr>
<th>Length in bytes</th>
<th>Meaning</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Default retention</td>
<td>A default retention tag is applied to a Message object when a regular retention tag is not present on the Message object. The absence of a regular retention tag indicates that the Message object does not have a specific retention policy. The application of the default tag is based on the Message object's message class.</td>
</tr>
<tr>
<td>8</td>
<td>Start date</td>
<td>The date, in UTC, from which the age of the Message object is calculated.</td>
</tr>
</tbody>
</table>

The PidTagStartDateEtc property can be present only on Message objects.

2.2.1.60.5 PidTagRetentionDate Property

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

The PidTagRetentionDate property ([MS-OXPROPS] section 2.933) specifies the date, in UTC, after which a Message object is expired by the server. The PidTagRetentionDate property can be present only on Message objects, not on folders. If the property is not present, the Message object never expires. The PidTagRetentionDate property can be set by both client and server.

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

- When the Message object has a specific retention policy:

  \[ \text{PidTagRetentionDate} = \text{PidTagMessageDeliveryTime} \quad (\text{[MS-OXOMSG] section 2.2.3.9}) + \text{PidTagRetentionPeriod} \quad (\text{section 2.2.1.60.3}) \]

  If the PidTagMessageDeliveryTime property does not exist, the PidTagCreationTime property (section 2.2.2.2) is used.

- When the Message object has the default retention policy:
\[ \text{PidTagRetentionDate} = \text{PidTagMessageDeliveryTime} + \text{default retention period. If the} \]
\[ \text{PidTagMessageDeliveryTime} \text{ property does not exist, the} \text{PidTagCreationTime} \text{ property is used.} \]

### 2.2.1.60.6 PidTagRetentionFlags Property

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

The PidTagRetentionFlags property ([MS-OXPROPS] section 2.934) contains flags that specify the status or nature of an item's retention tag or archive tag. The PidTagRetentionFlags property can be present on both Message objects and folders and can be set by both client and server.

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

<table>
<thead>
<tr>
<th>Flag name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExplicitTag</td>
<td>0x00000001</td>
<td>The retention tag on the folder is explicitly set.</td>
</tr>
<tr>
<td>UserOverride</td>
<td>0x00000002</td>
<td>The retention tag was not changed by the end user.</td>
</tr>
<tr>
<td>AutoTag</td>
<td>0x00000004</td>
<td>The retention tag on the Message object is an autotag, which is predicted by the system.</td>
</tr>
<tr>
<td>PersonalTag</td>
<td>0x00000008</td>
<td>The retention tag on the folder is of a personal type and can be made available to the end user.</td>
</tr>
<tr>
<td>ExplicitArchiveTag</td>
<td>0x00000010</td>
<td>The archive tag on the folder is explicitly set.</td>
</tr>
<tr>
<td>KeepInPlace</td>
<td>0x00000020</td>
<td>The Message object remains in place and is not archived.</td>
</tr>
<tr>
<td>SystemData</td>
<td>0x00000040</td>
<td>The Message object or folder is system data.</td>
</tr>
<tr>
<td>NeedsRescan&lt;7&gt;</td>
<td>0x00000080</td>
<td>The folder needs to be rescanned.</td>
</tr>
<tr>
<td>PendingRescan&lt;8&gt;</td>
<td>0x00000100</td>
<td>The folder is being rescanned.</td>
</tr>
</tbody>
</table>

### 2.2.1.60.7 PidTagArchivePeriod Property

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

The PidTagArchivePeriod property ([MS-OXPROPS] section 2.580) specifies the number of days that a Message object can remain unarchived. The PidTagArchivePeriod property can be present on both Message objects and folders and can be set by both client and server.

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

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

### 2.2.1.60.8 PidTagArchiveDate Property

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

The PidTagArchiveDate property ([MS-OXPROPS] section 2.579) specifies the date, in UTC, after which a Message object is archived by the server. The PidTagArchiveDate property can be present on only Message objects, not on folders, and can be set by both client and server. If the PidTagArchiveDate property is not present, the Message object is never archived by the server.
The value of the **PidTagArchiveDate** property is calculated from the values of other properties as follows:

\[
\text{PidTagArchiveDate} = \text{start date} + \text{PidTagArchivePeriod} \quad (\text{section 2.2.1.60.7})
\]

The start date is obtained from the last eight bytes of the **PidTagStartDateEtc** property (section 2.2.1.60.4).

### 2.2.2 Attachment Object Properties

#### 2.2.2.1 General Properties

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

- **PidTagAccessLevel** ([MS-OXCPRPT] section 2.2.1.2)
- **PidTagObjectType** <9> ([MS-OXCPRPT] section 2.2.1.7)
- **PidTagRecordKey** ([MS-OXCPRPT] section 2.2.1.8)

#### 2.2.2.2 PidTagLastModificationTime Property

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

The **PidTagLastModificationTime** property ([MS-OXPROPS] section 2.765) indicates the last time the file referenced by the Attachment object was modified, or the last time the Attachment object itself was modified.

#### 2.2.2.3 PidTagCreationTime Property

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

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

#### 2.2.2.4 PidTagDisplayName Property

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

The **PidTagDisplayName** property ([MS-OXPROPS] section 2.677) contains the name of the attachment as input by the end user. This property is set to the same value as the **PidTagAttachLongFilename** property (section 2.2.2.13).

#### 2.2.2.5 PidTagAttachSize Property

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

The **PidTagAttachSize** property ([MS-OXPROPS] section 2.608) contains the size in bytes consumed by the Attachment object on the server. This property is read-only for the client.

#### 2.2.2.6 PidTagAttachNumber Property

Type: **PtypInteger32**, unsigned ([MS-OXCDATA] section 2.11.1)
The `PidTagAttachNumber` property ([MS-OXPROPS] section 2.603) identifies the Attachment object within its Message object. The value of this property MUST be unique among the Attachment objects in a message.

### 2.2.2.7 PidTagAttachDataBinary Property

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

The `PidTagAttachDataBinary` property ([MS-OXPROPS] section 2.589) contains the contents of the file to be attached.

### 2.2.2.8 PidTagAttachDataObject Property

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

The `PidTagAttachDataObject` property ([MS-OXPROPS] section 2.590) contains the binary representation of the Attachment object in an application-specific format.

### 2.2.2.9 PidTagAttachMethod Property

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

The `PidTagAttachMethod` property ([MS-OXPROPS] section 2.601) represents the way the contents of an attachment are accessed. This property is set to one of the following values.

<table>
<thead>
<tr>
<th>Flag name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>afNone</td>
<td>0x00000000</td>
<td>The attachment has just been created.</td>
</tr>
<tr>
<td>afByValue</td>
<td>0x00000001</td>
<td>The <code>PidTagAttachDataBinary</code> property (section 2.2.2.7) contains the attachment data.</td>
</tr>
<tr>
<td>afByReference</td>
<td>0x00000002</td>
<td>The <code>PidTagAttachLongPathname</code> property (section 2.2.2.13) contains a fully qualified path identifying the attachment To recipients with access to a common file server.</td>
</tr>
<tr>
<td>afByReferenceOnly</td>
<td>0x00000004</td>
<td>The <code>PidTagAttachLongPathname</code> property contains a fully qualified path identifying the attachment.</td>
</tr>
<tr>
<td>afEmbeddedMessage</td>
<td>0x00000005</td>
<td>The attachment is an embedded message that is accessed via the <code>RopOpenEmbeddedMessage</code> ROP ([MS-OXCROPS] section 2.2.6.16).</td>
</tr>
<tr>
<td>afStorage</td>
<td>0x00000006</td>
<td>The <code>PidTagAttachDataObject</code> property (section 2.2.2.8) contains data in an application-specific format.</td>
</tr>
<tr>
<td>afByWebReference</td>
<td>0x00000007</td>
<td>The <code>PidTagAttachLongPathname</code> property contains a fully qualified path identifying the attachment. The <code>PidNameAttachmentProviderType</code> defines the web service API manipulating the attachment.</td>
</tr>
</tbody>
</table>

### 2.2.2.10 PidTagAttachLongFilename Property

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

The `PidTagAttachLongFilename` property ([MS-OXPROPS] section 2.595) contains the full file name and extension of the Attachment object.
2.2.2.11  PidTagAttachFilename Property
Type: PtypString ([MS-OXCDATA] section 2.11.1)

The PidTagAttachFilename property ([MS-OXPROPS] section 2.593) contains the **8.3 name** of the value of the PidTagAttachLongFilename property (section 2.2.2.10).

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

The PidTagAttachExtension property ([MS-OXPROPS] section 2.592) contains a file name extension that indicates the document type of an attachment.

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

The PidTagAttachLongPathname property ([MS-OXPROPS] section 2.596) contains the fully qualified path and file name with extension.

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

The PidTagAttachPathname property ([MS-OXPROPS] section 2.604) contains the **8.3 name** of the value of the PidTagAttachLongPathname property (section 2.2.2.13).

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

The PidTagAttachTag property ([MS-OXPROPS] section 2.609) contains the identifier information for the application that supplied the Attachment object’s data. This property can be left unset; if set, it MUST be one of the following.

<table>
<thead>
<tr>
<th>Definition</th>
<th>Data</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>TNEF</td>
<td>{0x2A,86,48,86,F7,14,03,0A,01}</td>
<td>The TNEF format is specified in [MS-OXTNEF].</td>
</tr>
<tr>
<td>afStorage</td>
<td>{0x2A,86,48,86,F7,14,03,0A,03,02,01}</td>
<td>Data is in an application-specific format.</td>
</tr>
<tr>
<td>MIME</td>
<td>{0x2A,86,48,86,F7,14,03,0A,04}</td>
<td>Conversion between Message object and MIME formats is specified in [MS-OXCMAIL].</td>
</tr>
</tbody>
</table>

2.2.2.16  PidTagRenderingPosition Property
Type: PtypInteger32, unsigned ([MS-OXCDATA] section 2.11.1)

The PidTagRenderingPosition property ([MS-OXPROPS] section 2.915) represents an offset, in rendered characters, to use when rendering an attachment within the main message text.

The values specify a relative ordering of the rendered attachment in the text. If a message has three attachments with values of 200, 100, and 500 for the PidTagRenderingPosition property, these will be rendered in the same order as if the attachments had the values 2, 1, and 5. A detailed example of this property is provided in [MS-OXRTFEX] section 3.2.
The value 0xFFFFFFFF indicates a hidden attachment that is not to be rendered in the main text.

### 2.2.2.17 PidTagAttachRendering Property

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

The PidTagAttachRendering property ([MS-OXPROPS] section 2.607) contains a Windows Metafile Format (WMF) metafile as specified in [MS-WMF] for the Attachment object.

### 2.2.2.18 PidTagAttachFlags Property

Type: PtypInteger32, as a bit field ([MS-OXCDATA] section 2.11.1)

The PidTagAttachFlags property ([MS-OXPROPS] section 2.594) indicates which body formats might reference this attachment when rendering data. This property contains a bitwise OR of zero or more of the following flags. If this property is absent or its value is 0x00000000, the attachment is available to be rendered in any format.

<table>
<thead>
<tr>
<th>Flag name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>attInvisibleInHtml</td>
<td>0x00000001</td>
<td>The Attachment object is not available to be rendered in HTML.</td>
</tr>
<tr>
<td>attInvisibleInRtf</td>
<td>0x00000002</td>
<td>The Attachment object is not available to be rendered in RTF.</td>
</tr>
<tr>
<td>attRenderedInBody</td>
<td>0x00000004</td>
<td>The Attachment object is referenced and rendered within the HTML body of the associated Message object. More details are specified in the PidTagBodyHtml property (section 2.2.1.58.3).</td>
</tr>
</tbody>
</table>

### 2.2.2.19 PidTagAttachTransportName Property

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

The PidTagAttachTransportName property ([MS-OXPROPS] section 2.610) contains the name of an attachment file, modified so that it can be correlated with TNEF messages, as specified in [MS-OXTNEF].

### 2.2.2.20 PidTagAttachEncoding Property

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

The PidTagAttachEncoding property ([MS-OXPROPS] section 2.591) contains encoding information about the Attachment object. If the attachment is in MacBinary format, this property is set to "{0x2A,86,48,86,F7,14,03,0B,01}"; otherwise, it is unset. This property is used to indicate that the attachment content, which is the value of the PidTagAttachDataBinary property (section 2.2.2.7), MUST be encoded in the MacBinary format, as specified in [MS-OXCMAIL]. Clients SHOULD correctly detect MacBinary I, MacBinary II, and MacBinary III formats.

### 2.2.2.21 PidTagAttachAdditionalInformation Property

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

The PidTagAttachAdditionalInformation property ([MS-OXPROPS] section 2.585) MUST be set to an empty string if the PidTagAttachEncoding property (section 2.2.2.20) is unset. If the PidTagAttachEncoding property is set, the PidTagAttachAdditionalInformation property MUST
be set to a string of the format ":CREA:TYPE", where ":CREA" is the four-letter Macintosh file creator code, and ":TYPE" is a four-letter Macintosh type code.

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

The PidTagAttachmentLinkId property ([MS-OXPROPS] section 2.600) is the type of Message object to which this attachment is linked. This property MUST be set to 0x00000000 unless overridden by other protocols that extend the Message and Attachment Object Protocol as noted in section 1.4.

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

The PidTagAttachmentFlags property ([MS-OXPROPS] section 2.598) indicates special handling for this Attachment object. This property MUST be set to 0x00000000 unless overridden by other protocols that extend the Message and Attachment Object Protocol as noted in section 1.4.

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

The PidTagAttachmentHidden property ([MS-OXPROPS] section 2.599) is set to TRUE (0x01) if this Attachment object is hidden from the end user.

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

The PidTagTextAttachmentCharset property ([MS-OXPROPS] section 2.1045) specifies the character set of messages for messages with a text body. This property corresponds to the charset parameter of the Content-Type header, as specified in [MS-OXCMAIL] section 2.2.3.4.1.2.

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

The PidNameAttachmentProviderType property ([MS-OXPROPS] section 2.372) contains the type of web service manipulating the attachment.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OneDrivePro</td>
<td>The web reference attachment belongs to a OneDrive for Business service.</td>
</tr>
<tr>
<td>OneDriveConsumer</td>
<td>The web reference attachment belongs to a OneDrive Consumer service.</td>
</tr>
</tbody>
</table>

2.2.2.27 PidNameAttachmentOriginalPermissionType
Type: PtypInteger32 ([MS-OXCDATA] section 2.11.1)
The **PidNameAttachmentOriginalPermissionType** property ([MS-OXPROPS] section 2.370) contains the original permission type data associated with a web reference attachment.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>None. User has no permissions to share.</td>
</tr>
<tr>
<td>1</td>
<td>View. User can only read the web reference attachment.</td>
</tr>
<tr>
<td>2</td>
<td>Edit. User can edit the web reference attachment.</td>
</tr>
</tbody>
</table>

### 2.2.2.28 PidNameAttachmentPermissionType

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

The **PidNameAttachmentPermissionType** property ([MS-OXPROPS] section 2.371) contains the permission type data associated with a web reference attachment.

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>None. User has no permissions to share.</td>
</tr>
<tr>
<td>1</td>
<td>View. User can only read the web reference attachment.</td>
</tr>
<tr>
<td>2</td>
<td>Edit. User can edit the web reference attachment.</td>
</tr>
</tbody>
</table>

### 2.2.2.29 MIME Properties

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

<table>
<thead>
<tr>
<th>Type</th>
<th>Property name</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>PtypString</td>
<td>PidTagAttachMimeTag ([MS-OXPROPS] section 2.602)</td>
<td>The Content-Type header.</td>
</tr>
<tr>
<td>PtypString</td>
<td>PidTagAttachContentId ([MS-OXPROPS] section 2.587)</td>
<td>A content identifier unique to this Message object that matches a corresponding &quot;cid:&quot; URI scheme reference in the HTML body of the Message object.</td>
</tr>
<tr>
<td>PtypString</td>
<td>PidTagAttachContentLocation ([MS-OXPROPS] section 2.588)</td>
<td>A relative or full URI that matches a corresponding reference in the HTML body of the Message object.</td>
</tr>
<tr>
<td>PtypString</td>
<td>PidTagAttachContentBase ([MS-OXPROPS] section 2.586)</td>
<td>The base of a relative URI. MUST be set if the PidTagAttachContentLocation property contains a relative URI.</td>
</tr>
<tr>
<td>PtypString</td>
<td>PidTagAttachPayloadClass ([MS-OXPROPS] section 2.605)</td>
<td>The class name of an object that can display the contents of the message.</td>
</tr>
<tr>
<td>PtypString</td>
<td>PidTagAttachPayloadProviderGuidString ([MS-OXPROPS] section 2.606)</td>
<td>The GUID of the software application that can display the contents of the message.</td>
</tr>
</tbody>
</table>
2.2.3 Message Object ROPs

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

2.2.3.1 RopOpenMessage ROP

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

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

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].

2.2.3.1.1 RopOpenMessage ROP Request Buffer

The following descriptions define valid fields for the request buffer of the RopOpenMessage ROP ([MS-OXCROPS] section 2.2.6.1).

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

**FolderID**: 8 bytes containing the FID ([MS-OXCDATA] section 2.2.1.1) of the folder from which the message is to be opened.

**OpenModeFlags**: 1 byte. The values given in the following table are valid; if other bits are set, they are ignored.

<table>
<thead>
<tr>
<th>Flag name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadOnly</td>
<td>0x00</td>
<td>Message will be opened as read-only.</td>
</tr>
<tr>
<td>ReadWrite</td>
<td>0x01</td>
<td>Message will be opened for both reading and writing.</td>
</tr>
<tr>
<td>BestAccess</td>
<td>0x03</td>
<td>Open for read/write if the user has write permissions for the folder, read-only if not.&lt;12&gt;</td>
</tr>
<tr>
<td>OpenSoftDeleted</td>
<td>0x04</td>
<td>Open a soft deleted Message object if available.</td>
</tr>
</tbody>
</table>
2.2.3.1.2 RopOpenMessage ROP Response Buffer

The following descriptions define valid fields for the response buffer of the RopOpenMessage ROP ([MS-OXCROPS] section 2.2.6.1).

**HasNamedProperties:** 1 byte.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>No named properties are defined for this Message object.</td>
</tr>
<tr>
<td>Nonzero</td>
<td>Named properties are defined for this Message object and can be obtained through a RopGetPropertiesAll ROP request ([MS-OXCROPS] section 2.2.8.4). Named properties can consist of custom properties added and maintained by third-party applications.</td>
</tr>
</tbody>
</table>

**SubjectPrefix:** A TypedString structure ([MS-OXCDATA] section 2.11.7) specifying the prefix for the subject of the Message object. The SubjectPrefix field contains the value of the PidTagSubjectPrefix property (section 2.2.1.9).

**NormalizedSubject:** A TypedString structure specifying the normalized subject of the Message object. The NormalizedSubject field contains the value of the PidTagNormalizedSubject property (section 2.2.1.10).

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

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

**RecipientColumns:** An array of PropertyTag structures ([MS-OXCDATA] section 2.9) with the number of elements specified in the ColumnCount field. Each PropertyTag value is valid for a recipient (2) as specified in [MS-OXPROPS].

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

**RecipientRows:** An array of OpenRecipientRow structures whose number is equal to the value of the RowCount field.

The value of the RecipientType field specified in [MS-OXCROPS] section 2.2.6.1.2.1 is a bitwise OR of zero or one value from the Types table with zero or more values from the flags table. Valid values for the RecipientType field are given in the following table.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x01</td>
<td>Primary recipient</td>
</tr>
<tr>
<td>0x02</td>
<td>Carbon copy (Cc) recipient</td>
</tr>
<tr>
<td>0x03</td>
<td>Blind carbon copy (Bcc) recipient</td>
</tr>
</tbody>
</table>

Valid flags are as follows.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x10</td>
<td>When resending a previous failure, this flag indicates that this recipient (1) did not successfully receive the message on the previous attempt.</td>
</tr>
<tr>
<td>0x80</td>
<td>When resending a previous failure, this flag indicates that this recipient (1) did successfully receive the message on the previous attempt.</td>
</tr>
</tbody>
</table>
2.2.3.2 RopCreateMessage ROP

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

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

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].

2.2.3.2.1 RopCreateMessage ROP Request Buffer

The following descriptions define valid fields for the request buffer of the RopCreateMessage ROP ([MS-OXCROPS] section 2.2.6.2).

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

- **FolderId**: 8 bytes containing the FID ([MS-OXCDATA] section 2.2.1.1) for the Folder object in which the Message object is to be created.

- **AssociatedFlag**: 1 byte Boolean value.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>Is not an FAI message.</td>
</tr>
<tr>
<td>Nonzero</td>
<td>Is an FAI message.</td>
</tr>
</tbody>
</table>

2.2.3.2.2 RopCreateMessage ROP Response Buffer

The following descriptions define valid fields for the response buffer of the RopCreateMessage ROP ([MS-OXCROPS] section 2.2.6.2).

- **HasMessageId**: 1 byte.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>This is the last byte in the buffer.</td>
</tr>
<tr>
<td>Nonzero</td>
<td>The MessageId field follows beginning with the next byte in the buffer.</td>
</tr>
</tbody>
</table>

- **MessageId**: 8 bytes containing the MID ([MS-OXCDATA] section 2.2.1.2) for the newly created Message object.

2.2.3.3 RopSaveChangesMessage ROP

The RopSaveChangesMessage ROP ([MS-OXCROPS] section 2.2.6.3) commits the changes made to the Message object.
For this ROP, the value of the **ResponseHandleIndex** field references the containing **Folder object** or, for an embedded message, the **Embedded Message object**. The value of the **InputHandleIndex** field references a Message object.

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].

### 2.2.3.3.1 RopSaveChangesMessage ROP Request Buffer

The following descriptions define valid fields for the request buffer of the **RopSaveChangesMessage ROP** ([MS-OXCROPS] section 2.2.6.3).

**SaveFlags**: 1 byte indicating the server save behavior; this field MUST be set to none or one of the flags specified in the following table. Other flags are not valid and are ignored by the server.

<table>
<thead>
<tr>
<th>Flag name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KeepOpenReadOnly</td>
<td>0x01</td>
<td>The client requests that the server commit the changes. The server either returns an error and leaves the Message object open with unchanged access level or returns a success code and keeps the Message object open with read-only access. More details about access levels are specified in [MS-OXCPRT] section 2.2.1.2.</td>
</tr>
<tr>
<td>KeepOpenReadWrite</td>
<td>0x02</td>
<td>The client requests that the server commit the changes. The server either returns an error and leaves the Message object open with unchanged access level or returns a success code and keeps the Message object open with read/write access.</td>
</tr>
<tr>
<td>ForceSave</td>
<td>0x04</td>
<td>The client requests that the server commit the changes. The server either returns an error and leaves the Message object open with unchanged access level or returns a success code and keeps the Message object open with read/write access. The <strong>ecObjectModified</strong> error code is not valid when this flag is set; the server overwrites any changes instead.</td>
</tr>
</tbody>
</table>

### 2.2.3.3.2 RopSaveChangesMessage ROP Response Buffer

The following descriptions define valid fields for the response buffer of the **RopSaveChangesMessage ROP** ([MS-OXCROPS] section 2.2.6.3).

**MessageId**: 8 bytes containing the MID ([MS-OXCDATA] section 2.2.1.2) for the saved Message object.

### 2.2.3.4 RopRemoveAllRecipients ROP

The client sends the **RopRemoveAllRecipients ROP request** ([MS-OXCROPS] section 2.2.6.4) to delete all recipients (2) from a message.

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

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].

### 2.2.3.4.1 RopRemoveAllRecipients ROP Request Buffer
The following descriptions define valid fields for the request buffer of the **RopRemoveAllRecipients ROP** ([MS-OXCROPS] section 2.2.6.4).

**Reserved**: 4 bytes; unspecified value.

### 2.2.3.4.2 RopRemoveAllRecipients ROP Response Buffer

The syntax of the **RopRemoveAllRecipients ROP response buffer** is specified in [MS-OXCROPS] section Error! Hyperlink reference not valid..

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

### 2.2.3.5 RopModifyRecipients ROP

The **RopModifyRecipients ROP** ([MS-OXCROPS] section 2.2.6.5) modifies recipients (2) associated with the **Message object**.

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

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].

#### 2.2.3.5.1 RopModifyRecipients ROP Request Buffer

The following descriptions define valid fields for the request buffer of the **RopModifyRecipients ROP** ([MS-OXCROPS] section 2.2.6.5).

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

**RecipientColumns**: An array of **PropertyTag** structures whose number is equal to the value of the **ColumnCount** field. Each element is valid for a recipient (2) as specified in [MS-OXPROPS]. The client MUST NOT include property tags for any properties that are part of standard property values of the **RecipientRow** field, as specified in [MS-OXCDATA] section 2.8.3:

- **PidTagAddressType** ([MS-OXOABK] section 2.2.3.13)
- **PidTagDisplayName** (section 2.2.2.4)
- **PidTagEmailAddress** ([MS-OXOABK] section 2.2.3.14)
- **PidTagEntryId** ([MS-OXCPERM] section 2.2.4)
- **PidTagInstanceKey** ([MS-OXOABK] section 2.2.3.6)
- **PidTagRecipientType** ([MS-OXOMSG] section 2.2.3.1)
- **PidTagSearchKey** ([MS-OXCPRT] section 2.2.1.9)
- **PidTagSendRichInfo** ([MS-OXOABK] section 2.2.3.18)
- **PidTagTransmittableDisplayName** ([MS-OXOABK] section 2.2.3.8)

**RowCount**: 2 bytes containing the number of elements in the **RecipientRows** field. The value of this field is greater than or equal to 0x0000 and less than 0x7FEF.

**RecipientRow**: An array of **ModifyRecipientRow** structures ([MS-OXCROPS] section 2.2.6.5.1.1) whose length equals the value of the **RowCount** field.
2.2.3.5.2 RopModifyRecipients ROP Response Buffer

The syntax of the RopModifyRecipients ROP response buffer is specified in [MS-OXCROPS] section 2.2.6.5.
This protocol adds no additional field information to the RopModifyRecipients ROP response buffer.

2.2.3.6 RopReadRecipients ROP

The RopReadRecipients ROP ([MS-OXCROPS] section 2.2.6.6) retrieves the recipients (2) associated with the Message object.

For this ROP, the value of the InputHandleIndex field references a Message object.

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].

2.2.3.6.1 RopReadRecipients ROP Request Buffer

The following descriptions define valid fields for the request buffer of the RopReadRecipients ROP ([MS-OXCROPS] section 2.2.6.6).

RowId: 4 bytes containing the starting index for the recipients (2) to be retrieved.

Reserved: 2 bytes; MUST be 0x0000.

2.2.3.6.2 RopReadRecipients ROP Response Buffer

The following descriptions define valid fields for the response buffer of the RopReadRecipients ROP ([MS-OXCROPS] section 2.2.6.6).

RowCount: 1 byte containing the number of elements in the RecipientRows field. Is greater than or equal to 0x00 and less than 0xFF.

RecipientRows: An array of ReadRecipientRow structures whose number of elements equals the RowCount field. The ReadRecipientRow structure is specified in [MS-OXCROPS] section 2.2.6.6.2.1.

2.2.3.7 RopReloadCachedInformation ROP

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

For this ROP, the value of the InputHandleIndex field references a Message object.

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].

2.2.3.7.1 RopReloadCachedInformation ROP Request Buffer

The following descriptions define valid fields for the request buffer of the RopReloadCachedInformation ROP ([MS-OXCROPS] section 2.2.6.7).

Reserved: 2 bytes; MUST be 0x0000.
2.2.3.7.2 RopReloadCachedInformation ROP Response Buffer

The syntax of the RopReloadCachedInformation ROP response buffer ([MS-OXCROPS] section 2.2.6.7) is identical to the syntax of the RopOpenMessage ROP response buffer ([MS-OXCROPS] section Error! Hyperlink reference not valid.).

2.2.3.8 RopSetMessageStatus ROP

The RopSetMessageStatus ROP ([MS-OXCROPS] section 2.2.6.8) sets the PidTagMessageStatus property ([MS-OXPROPS] section 2.799) on a message in a folder without the need to open or save the Message object.

For this ROP, the value of the InputHandleIndex field references a Folder object.

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].

2.2.3.8.1 RopSetMessageStatus ROP Request Buffer

The following descriptions define valid fields for the request buffer of the RopSetMessageStatus ROP ([MS-OXCROPS] section 2.2.6.8).

MessageId: 8 bytes containing the MID ([MS-OCXCDATA] section 2.2.1.2) for the Message object to modify.

MessageStatusFlags: 4 bytes containing the PidTagMessageStatus property ((section 2.2.1.8), which defines the status of the message in a contents table. Valid values are specified in section 2.2.1.8. This value of this field is combined with the value of the MessageStatusMask field to yield the new message status.

MessageStatusMask: 4 bytes indicating which status flags are to be set and which are to be cleared. This field contains a bitwise OR of zero or more values from the table in section 2.2.1.8. Processing information for this field is specified in section 3.2.5.8.

2.2.3.8.2 RopSetMessageStatus ROP Response Buffer

The following descriptions define valid fields for the response buffer of the RopSetMessageStatus ROP ([MS-OXCROPS] section 2.2.6.8).

MessageStatusFlags: 4 bytes indicating the status flags that were set on the Message object before processing this request. This field MUST contain a bitwise OR of zero or more values from the table in section 2.2.1.8.

2.2.3.9 RopGetMessageStatus ROP

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

For this ROP, the value of the InputHandleIndex field references a Folder object.

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].

2.2.3.9.1 RopGetMessageStatus ROP Request Buffer
The following descriptions define valid fields for the request buffer of the *RopGetMessageStatus ROP* ([MS-OXCROPS] section 2.2.6.9).

**MessageId**: 8 bytes containing the MID ([MS-OXCDATA] section 2.2.1.2) for the *Message object* in which to operate.

### 2.2.3.9.2 RopGetMessageStatus ROP Response Buffer

The following descriptions define valid fields for the request buffer of the *RopGetMessageStatus ROP* ([MS-OXCROPS] section 2.2.6.9).

**MessageStatusFlags**: 4 bytes indicating the status of the *Message object*. This field contains a bitwise OR of zero or more values from the table in section 2.2.1.8.

### 2.2.3.10 RopSetReadFlags ROP

The *RopSetReadFlags ROP* ([MS-OXCROPS] section 2.2.6.10) changes the state of the *PidTagMessageFlags* property (section 2.2.1.6) on one or more *Message objects* within a *Folder object*. It also triggers the sending of read receipts, as specified in [MS-OXOMSG].

For this ROP, the value of the *InputHandleIndex* field references a Folder object.

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].

#### 2.2.3.10.1 RopSetReadFlags ROP Request Buffer

The following descriptions define valid fields for the request buffer of the *RopSetReadFlags ROP* ([MS-OXCROPS] section 2.2.6.10).

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

**ReadFlags**: 1 byte containing a bitwise OR of zero or more values from the following table. The server modifies bits on the *PidTagMessageFlags* property (section 2.2.1.6). The flags, *rfGenerateReceiptOnly*, *rfsuppressReceipt*, and *rfClearReadFlag*, (rfClearNotifyRead or rfClearNotifyUnread), are mutually exclusive.

<table>
<thead>
<tr>
<th>Flag name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rfDefault</td>
<td>0x00</td>
<td>The server sets the read flag and sends the receipt.</td>
</tr>
<tr>
<td>rfSuppressReceipt</td>
<td>0x01</td>
<td>The user requests that any pending read receipt be canceled; the server sets the mfRead bit.</td>
</tr>
<tr>
<td>rfReserved</td>
<td>0x0A</td>
<td>Ignored by the server.</td>
</tr>
<tr>
<td>rfClearReadFlag</td>
<td>0x04</td>
<td>Server clears the mfRead bit; the client MUST include the rfSuppressReceipt bit with this flag.</td>
</tr>
<tr>
<td>rfGenerateReceiptOnly</td>
<td>0x10</td>
<td>The server sends a read receipt if one is pending, but does not change the mfRead bit.</td>
</tr>
<tr>
<td>rfClearNotifyRead</td>
<td>0x20</td>
<td>The server clears the mfNotifyRead bit but does not send a read receipt.</td>
</tr>
<tr>
<td>rfClearNotifyUnread</td>
<td>0x40</td>
<td>The server clears the mfNotifyUnread bit but does not send a nonread receipt.</td>
</tr>
</tbody>
</table>
**MessageIdCount**: 2 bytes containing the number of elements in the **MessageIds** field.

**MessageIds**: An array of MIDs ([MS-OXCDATA] section 2.2.1.2) whose length is equal to the value of the **MessageIdCount** field.

### 2.2.3.10.2 RopSetReadFlags ROP Response Buffer

The following descriptions define valid fields for the response buffer of the **RopSetReadFlags ROP** ([MS-OXCROPS] section 2.2.6.10).

**PartialCompletion**: 1 byte boolean flag. A nonzero value indicates the server was unable to modify one or more of the **Message objects** represented in the **MessageIds** field.<15>

### 2.2.3.11 RopSetMessageReadFlag ROP

The **RopSetMessageReadFlag ROP** ([MS-OXCROPS] section 2.2.6.11) changes the state of the **PidTagMessageFlags** property (section 2.2.1.6) for the **Message object**. It also triggers the sending of **read receipts**, as specified in [MS-OXOMSG].

In this section, "in public folder mode" means that the logon associated with the value of the **LogonID** field from the request was created with the **Private** flag unset.

For this ROP, the value of the **ResponseHandleIndex** field references a **Folder object**, and the value of the **InputHandleIndex** field references a Message object.

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].

### 2.2.3.11.1 RopSetMessageReadFlag ROP Request Buffer

The following descriptions define valid fields for the request buffer of the **RopSetMessageReadFlag ROP** ([MS-OXCROPS] section 2.2.6.11).

**ReadFlags**: 1 byte containing a bitwise OR of one or more values from the **ReadFlags** field table in section 2.2.3.10.1.

**ClientData**: A 24 byte LongTermID, as specified in [MS-OXCDATA] section 2.2.1.3.1, that represents the message read when in public folder mode; 0 bytes otherwise.

### 2.2.3.11.2 RopSetMessageReadFlag ROP Response Buffer

The following descriptions define valid fields for the response buffer of the **RopSetMessageReadFlag ROP** ([MS-OXCROPS] section 2.2.6.11).

**ReadStatusChanged**: 1 byte containing one of the following values.

<table>
<thead>
<tr>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>The read status on the <strong>Message object</strong> was unchanged, or the logon is not in public folder mode.</td>
</tr>
<tr>
<td>Nonzero</td>
<td>The read status on the Message object changed, and the logon is in public folder mode.</td>
</tr>
</tbody>
</table>

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

**ClientData**: 24 bytes containing the **ClientData** field from the request when the value in the **ReadStatusChanged** field is nonzero; 0 bytes otherwise.
2.2.3.12 RopOpenAttachment ROP

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

For this ROP, the value of the InputHandleIndex field references a Message object, and the value of the OutputHandleIndex field references an Attachment object.

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].

2.2.3.12.1 RopOpenAttachment ROP Request Buffer

The following descriptions define valid fields for the request buffer of the RopOpenAttachment ROP ([MS-OXCROPS] section 2.2.6.12).

OpenAttachmentFlags: 1 byte containing one of the following values.

<table>
<thead>
<tr>
<th>Value name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadOnly</td>
<td>0x00</td>
<td>Attachment will be opened as read-only.</td>
</tr>
<tr>
<td>ReadWrite</td>
<td>0x01</td>
<td>Attachment will be opened for both reading and writing.</td>
</tr>
<tr>
<td>BestAccess</td>
<td>0x03</td>
<td>Attachment will be opened for read/write if the user has write permissions for the attachment; opened for read-only if not.</td>
</tr>
</tbody>
</table>

AttachmentID: 4 bytes containing the ID of the Attachment object to be opened, as specified by the PidTagAttachNumber property (section 2.2.2.6).

2.2.3.12.2 RopOpenAttachment ROP Response Buffer

The syntax of the RopOpenAttachment ROP response buffer is specified in [MS-OXCROPS] section 2.2.6.12.

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

2.2.3.13 RopCreateAttachment ROP

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

For this ROP, the value of the InputHandleIndex field references a Message object, and the value of the OutputHandleIndex field references an Attachment object.

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].

2.2.3.13.1 RopCreateAttachment ROP Request Buffer

The syntax of the RopCreateAttachment ROP request buffer is specified in [MS-OXCROPS] section 2.2.6.13.

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

2.2.3.13.2 RopCreateAttachment ROP Response Buffer
The syntax of the **RopCreateAttachment ROP response buffer** is specified in [MS-OXCROPS] section 2.2.6.13.

The field specified in this section is part of the **RopCreateAttachment** ROP response buffer.

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

### 2.2.3.14 RopDeleteAttachment ROP

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

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

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].

#### 2.2.3.14.1 RopDeleteAttachment ROP Request Buffer

The following descriptions define valid fields for the request buffer of the **RopDeleteAttachment ROP** ([MS-OXCROPS] section 2.2.6.14).

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

#### 2.2.3.14.2 RopDeleteAttachment ROP Response Buffer

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

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

### 2.2.3.15 RopSaveChangesAttachment ROP

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

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

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

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].

#### 2.2.3.15.1 RopSaveChangesAttachment ROP Request Buffer

The following descriptions define valid fields for the request buffer of the **RopSaveChangesAttachment ROP** ([MS-OXCROPS] section 2.2.6.15).

**SaveFlags**: As specified in section 2.2.3.3.1.

#### 2.2.3.15.2 RopSaveChangesAttachment ROP Response Buffer

The syntax of the **RopSaveChangesAttachment ROP response buffer** is specified in [MS-OXCROPS] section 2.2.6.15.
This protocol adds no additional field information to the `RopSaveChangesAttachment` ROP response buffer.

### 2.2.3.16 RopOpenEmbeddedMessage ROP

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

For this ROP, the value of the `InputHandleIndex` field references an Attachment object and the value of the `OutputHandleIndex` field references a Message object.

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].

#### 2.2.3.16.1 RopOpenEmbeddedMessage ROP Request Buffer

The following descriptions define valid fields for the request buffer of the `RopOpenEmbeddedMessage` ROP ([MS-OXCROPS] section 2.2.6.16).

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

**OpenModeFlags**: 1 byte. The following values are valid for this flag.

<table>
<thead>
<tr>
<th>Value name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReadOnly</td>
<td>0x00</td>
<td>Open the message as read-only.</td>
</tr>
<tr>
<td>ReadWrite</td>
<td>0x01</td>
<td>Open the message for both reading and writing.</td>
</tr>
<tr>
<td>Create</td>
<td>0x02</td>
<td>Create the attachment if it does not already exist and open the message for both reading and writing.</td>
</tr>
</tbody>
</table>

#### 2.2.3.16.2 RopOpenEmbeddedMessage ROP Response Buffer

The following descriptions define valid fields for the response buffer of the `RopOpenEmbeddedMessage` ROP ([MS-OXCROPS] section 2.2.6.16).

**MessageId**: 8 bytes containing the MID ([MS-OXCDATA] section 2.2.1.2) for the `Message object`.

The following fields are as specified in section 2.2.3.1.2: `HasNamedProperties`, `SubjectPrefix`, `NormalizedSubject`, `RecipientCount`, `ColumnType`, `RecipientColumns`, `RowCount`, and `RecipientRows`. The presence of data in the `RecipientRows` field is indeterminate, even when the embedded message exists.

### 2.2.3.17 RopGetAttachmentTable ROP

The `RopGetAttachmentTable` ROP ([MS-OXCROPS] section 2.2.6.17) retrieves a handle to a `Table object` that represents the attachments stored on the `Message object`. For more details on Table objects, see [MS-OXCTABL].

For this ROP, the value of the `InputHandleIndex` field references a Message object, and the value of the `OutputHandleIndex` field references a Table object.
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].

### 2.2.3.17.1 RopGetAttachmentTable ROP Request Buffer

The following descriptions define valid fields for the request buffer of the RopGetAttachmentTable ROP ([MS-OXCROPS] section 2.2.6.17).

**TableFlags:** 1 byte. The following values are valid for this flag.

<table>
<thead>
<tr>
<th>Value name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard</td>
<td>0x00</td>
<td>Open the table.</td>
</tr>
<tr>
<td>Unicode</td>
<td>0x40</td>
<td>Open the table. Also requests that the columns containing string data be returned in Unicode format.</td>
</tr>
</tbody>
</table>

### 2.2.3.17.2 RopGetAttachmentTable ROP Response Buffer

The syntax of the RopGetAttachmentTable ROP response buffer is specified in [MS-OXCROPS] section 2.2.6.17.

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

### 2.2.3.18 RopGetValidAttachments ROP

The RopGetValidAttachments ROP ([MS-OXCROPS] section 2.2.6.18) gets the attachment IDs for all attachments that have previously been saved and have been assigned a valid numeric identifier.\(^{<18}\)

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].

#### 2.2.3.18.1 RopGetValidAttachments ROP Request Buffer

The syntax of the RopGetValidAttachments ROP request buffer is specified in [MS-OXCROPS] section 2.2.6.18.1.

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

#### 2.2.3.18.2 RopGetValidAttachments ROP Response Buffer

The syntax of the RopGetValidAttachments ROP response buffer is specified in [MS-OXCROPS] section 2.2.6.18.2.

This protocol adds no additional field information to the RopGetValidAttachments ROP response buffer.
3 Protocol Details

3.1 Client Details

3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

The following abstract data model (ADM) data types are defined in this section:

Global

Mailbox

Message Object

3.1.1.1 Per Global

This protocol includes the following ADM elements for the client:

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

Session context cookie, as specified in [MS-OXCMAP1HTTP] section 3.1.1.<19>

3.1.1.2 Per Mailbox

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


3.1.1.3 Per Message Object

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

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

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

3.1.2 Timers

None.

3.1.3 Initialization

None.
3.1.4 Higher-Layer Triggered Events

3.1.4.1 Opening a Message Object
A client obtains a handle to an existing Message object by sending a RopOpenMessage ROP request ([MS-OXCROPS] section 2.2.6.1).

3.1.4.2 Creating a Message Object
A client creates a new Message object by sending a RopCreateMessage ROP request ([MS-OXCROPS] section 2.2.6.2).

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

3.1.4.4 Removing All Recipients
A client clears all recipients (2) from a Message object by sending a RopRemoveAllRecipients ROP request ([MS-OXCROPS] section 2.2.6.4).

3.1.4.5 Adding, Deleting, or Modifying a Recipient
A client modifies recipients (2) of the Message object by sending a RopModifyRecipients ROP request ([MS-OXCROPS] section 2.2.6.5).

3.1.4.6 Reading Recipients
A client retrieves a list of all recipients (2) on the Message object by sending a RopReadRecipients ROP request ([MS-OXCROPS] section 2.2.6.6).

3.1.4.7 Reload Message Object Header Info
A client retrieves the current state of the data returned in a RopOpenMessage ROP ([MS-OXCROPS] section 2.2.6.1) by sending a RopReloadCachedInformation ROP request ([MS-OXCROPS] section 2.2.6.7).

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

3.1.4.9 Getting Message Status
A client checks the status of a header message object by sending a RopGetMessageStatus ROP request ([MS-OXCROPS] section 2.2.6.9).

3.1.4.10 Setting Message Object Read State
A client marks one or more Message objects as read or unread without opening the Message objects by sending a RopSetReadFlags ROP request ([MS-OXCROPS] section 2.2.6.10).
When a user marks or unmarks a single opened Message object as read, the client sends a RopSetMessageReadFlag ROP request ([MS-OXCROPS] section 2.2.6.11).

### 3.1.4.11 Opening an Attachment

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

### 3.1.4.12 Creating an Attachment

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

### 3.1.4.13 Deleting an Attachment

A client deletes an attachment from a Message object by sending a RopDeleteAttachment ROP request ([MS-OXCROPS] section 2.2.6.14).

### 3.1.4.14 Setting Attachment Object Content

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

### 3.1.4.15 Saving Changes on an Attachment Object

A client saves changes to an Attachment object by sending a RopSaveChangesAttachment ROP request ([MS-OXCROPS] section 2.2.6.15).

### 3.1.4.16 Opening an Embedded Message Object

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

### 3.1.4.17 Accessing the Attachments Table

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

### 3.1.4.18 Creating an Embedded Message

A client creates an embedded message by sending a RopCreateAttachment ROP request ([MS-OXCROPS] section 2.2.6.13) to create an attachment on a message.

### 3.1.4.19 Saving an Embedded Message

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

To link a Contact object with another Message object, the client sets the following properties. Additional details are specified in section 2.2.1.59.

- PidLidContactLinkEntry (section 2.2.1.59.1)
- PidLidContactLinkName (section 2.2.1.59.3)
- PidLidContactLinkSearchKey (section 2.2.1.59.4)
- PidLidContacts (section 2.2.1.59.2)

3.1.5 Message Processing Events and Sequencing Rules

3.1.5.1 Sending a RopOpenMessage ROP Request

To send the RopOpenMessage ROP request ([MS-OXCROPS] section 2.2.6.1), the client first obtains the MID ([MS-OXCDATA] section 2.2.1.2) for the Message object to be opened, and either the FID ([MS-OXCDATA] section 2.2.1.1) or the LogonID. The MID is accessible from the contents table of the Folder object that contains the Message object by including the PidTagMid property ([MS-OXCFXICS] section 2.2.1.2.1) in a RopSetColumns ROP request ([MS-OXCROPS] section 2.2.5.1), as specified in [MS-OXCTABL] section 2.2.2.2.

When the client receives the response buffer, it can cache the data from the NormalizedSubject and SubjectPrefix fields to minimize further calls to the server; it then updates the cache when issuing a RopSetProperties ROP request ([MS-OXCROPS] section 2.2.8.6) for the PidTagNormalizedSubject property (section 2.2.1.10) and the PidTagSubjectPrefix property (section 2.2.1.9) and uses the cached values.

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

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

If a client does not recognize a message class, it reverts to acting on all but the last group, recursively, until a recognized form remains.

3.1.5.2 Sending a RopSaveChangesMessage ROP Request

The client controls the access level of the Message object after saving changes by calling the RopSaveChangesMessage ROP ([MS-OXCROPS] section 2.2.6.3) by setting the proper flags as specified in section 2.2.3.1.1.

3.1.5.3 Sending a RopCreateMessage ROP Request

After calling the RopCreateMessage ROP ([MS-OXCROPS] section 2.2.6.2), the client sends a RopSaveChangesMessage ROP request ([MS-OXCROPS] section 2.2.6.3) to commit the new Message object and uses the opened Message object in subsequent ROPs. It MUST eventually send a RopRelease ROP request ([MS-OXCROPS] section 2.2.15.3) on the Message object and, after doing so, MUST NOT use the Message object for any subsequent ROPs.
3.1.5.4 Sending a RopRemoveAllRecipients ROP Request

After calling the RopRemoveAllRecipients ROP ([MS-OXCROPS] section 2.2.6.4), the client commits the changes by sending a RopSaveChangesMessage ROP ([MS-OXCROPS] section 2.2.6.3) for the Message object associated with the removed recipients (2).

3.1.5.5 Sending a RopModifyRecipients ROP Request

To modify an existing recipient (2) using the RopModifyRecipients ROP ([MS-OXCROPS] section 2.2.6.5), the client sets the RowId field of the RecipientRows field to the row ID of the recipient (2) to be modified and sets all of the ModifyRecipientRow data to the desired values for that recipient (2), including any additional property information for the recipients (2). Additional property information is set by adding values for the PropertyTag field to the RecipientColumns field and including the property values in the RecipientRows field.

To delete an existing recipient (2), the client sets the RowId field to the row ID of the recipient (2) to be deleted and sets the RecipientRowSize field to 0x0000.

To add a new recipient (2), the client sets the RowId field to a value greater than the largest row ID for any recipient (2) that already exists on the Message object. The client sets all of the data in the ModifyRecipientRow field to the desired values for that recipient (2), including any additional property information.

To commit the changes, the client sends a RopSaveChangesMessage ROP ([MS-OXCROPS] section 2.2.6.3) for the Message object associated with the added recipients (2).

3.1.5.6 Sending a RopReadRecipients ROP Request

If the count of recipients and the count of recipient rows in the RopOpenMessage ROP response buffer ([MS-OXCROPS] section 2.2.6.1) are the same, then the client can use the information in the RecipientRow field from the RopOpenMessage ROP instead of sending a RopReadRecipients ROP request ([MS-OXCROPS] section 2.2.6.6). If the counts are not equal, the client MUST issue a series of RopReadRecipients ROP requests to retrieve all the recipients associated with the Message object.

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

3.1.5.7 Sending a RopSetMessageStatus ROP Request

The client uses the RopSetMessageStatus ROP request ([MS-OXCROPS] section 2.2.6.8) to set the value of the PidTagMessageStatus property (section 2.2.1.8). Additionally, the PidTagMessageStatus property can be set as a column on a contents table, as specified in [MS-OXCTABL] section 2.2.2.2.

To modify the status of a header message object, clients:

1. Obtain the message's MID ([MS-OXCDATA] section 2.2.1.2), as specified in section 3.1.5.1.
2. Send the RopSetMessageStatus ROP request ([MS-OXCROPS] section 2.2.6.8), setting the mask and status appropriately.

3.1.5.8 Sending a RopGetMessageStatus ROP Request

The client uses the RopGetMessageStatus ROP request ([MS-OXCROPS] section 2.2.6.9) to obtain the value of the PidTagMessageStatus property (section 2.2.1.8).
To retrieve the status of a header message object, clients:

1. Obtain the message's MID ([MS-OXCDATA] section 2.2.1.2), as specified in section 3.1.5.1.

2. Send the RopGetMessageStatus ROP request; if the request succeeds, the value of the header message object's PidTagMessageStatus property ([MS-OXPROPS] section 2.799) is returned in the response buffer.

### 3.1.5.9 Sending a RopSetReadFlags ROP Request

The client obtains a list of MIDs ([MS-OXCDATA] section 2.2.1.2) using a contents table, as specified in section 3.1.4.1, and uses the list of MIDs in the RopSetReadFlags ROP request ([MS-OXCROPS] section 2.2.6.10).

The client controls whether the Message object is marked as read or unread, as well as the sending of read receipts, by setting the appropriate flags as specified in section 2.2.3.10.1.

### 3.1.5.10 Sending a RopOpenAttachment ROP Request

When sending a RopOpenAttachment ROP request ([MS-OXCROPS] section 2.2.6.12) to open an attachment, the client MUST use a valid value for the AttachmentID field. For more information about opening an Embedded Message object, see section 3.1.4.17.

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

### 3.1.5.11 Sending a RopCreateAttachment ROP Request

After creating a new attachment with a call to the RopCreateAttachment ROP ([MS-OXCROPS] section 2.2.6.13), the client sends a RopSaveChangesAttachment ROP request ([MS-OXCROPS] section 2.2.6.15) to commit the new Attachment object and uses the newly created Attachment object in subsequent ROPS. The client eventually sends a RopRelease ROP request ([MS-OXCROPS] section 2.2.15.3) on the Attachment object and, after doing so, MUST NOT use the Attachment object for any subsequent ROPS.

To create an Embedded Message object, the client uses the RopProperties ROP ([MS-OXCROPS] section 2.2.8.6) to set the afEmbeddedMessage flag on the PidTagAttachMethod property (section 2.2.2.9). Finally the client sends a RopOpenEmbeddedMessage ROP ([MS-OXCROPS] section 2.2.6.16) on the attachment to get a Message object handle.

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

### 3.1.5.12 Sending a RopSetProperties ROP Request

Depending on the type of Attachment object being created by sending the RopSetProperties ROP request ([MS-OXCROPS] section 2.2.8.6), the client sets the appropriate value for the PidTagAttachMethod property, as specified in section 2.2.2.9.

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

To set the PidTagMessageStatus property (section 2.2.1.8), the client does not include it in a RopSetProperties ROP request ([MS-OXCROPS] section 2.2.8.6), as specified in [MS-OXCPRPT] section 2.2.5. Instead, the client calls the RopSetMessageStatus ROP ([MS-OXCROPS] section 2.2.6.8), as specified in section 2.2.3.8.
3.1.5.13 Sending a RopGetPropertiesSpecific ROP Request

To get the PidTagMessageStatus property (section 2.2.1.8), the client does not include it in a RopGetPropertiesSpecific ROP request ([MS-OXCROPS] section 2.2.8.3). Instead, the client calls the RopGetMessageStatus ROP ([MS-OXCROPS] section 2.2.6.9), as specified in section 2.2.3.9.

The client does not include the PidTagSubjectPrefix property (section 2.2.1.9) in a RopGetPropertiesSpecific ROP request ([MS-OXCROPS] section 2.2.8.3). Instead, the client uses the SubjectPrefix field from the RopOpenMessage ROP response buffer ([MS-OXCROPS] section 2.2.6.1).

The client does not include the PidTagNormalizedSubject property (section 2.2.1.10) in a RopGetPropertiesSpecific ROP request ([MS-OXCROPS] section 2.2.8.3). Instead, the client uses the NormalizedSubject field from the RopOpenMessage ROP response buffer ([MS-OXCROPS] section 2.2.6.1).

3.1.5.14 Sending a RopSaveChangesAttachment ROP Request

After saving changes to an attachment with a call to the RopSaveChangesAttachment ROP ([MS-OXCROPS] section 2.2.6.15), the client sends a RopSaveChangesMessage ROP request ([MS-OXCROPS] section 2.2.6.3) to commit the Attachment object changes to the Message object.

3.1.5.15 Sending a RopOpenEmbeddedMessage ROP Request

The client uses the Message object opened by the RopOpenEmbeddedMessage ROP ([MS-OXCROPS] section 2.2.6.16) in subsequent ROPs; it eventually sends a RopRelease ROP request ([MS-OXCROPS] section 2.2.15.3) on the Message object and, after doing so, MUST NOT use the Message object for any subsequent ROPs.

3.1.5.16 Sending a RopGetAttachmentTable ROP Request

When a client calls the RopGetAttachmentTable ROP ([MS-OXCROPS] section 2.2.6.17), the server returns a table of properties for each Attachment object associated with the Message object, as specified in [MS-OXCTABL]. To retrieve the attachment ID, the client includes the PidTagAttachNumber property (section 2.2.2.6) when sending a RopSetColumns ROP request ([MS-OXCROPS] section 2.2.5.1).

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Server Details

3.2.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.
The following ADM data types are defined in this section.

**Global**

**Mailbox**

**MessageObject**

**3.2.1.1 Per Global**

This protocol includes the following ADM elements for the server:


- **Session context cookie**, as specified in [MS-OXCMAPIHTTP] section 3.2.1.<20>

**3.2.1.2 Per Mailbox**

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

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

**3.2.1.3 Per Message Object**

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

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

- **MessageObject.Transactions**: A record of the open **Global.Handle** ADM elements (section 3.2.1.1) against a **MessageObject**. Each **Global.Handle** ADM element is given its own **transaction**. If a change to a **MessageObject** is committed on one **Global.Handle** ADM element, the server prevents changes from being saved against other open **Global.Handle** ADM elements on the same **MessageObject**, as specified in section 3.2.5.3.

**3.2.2 Timers**

None.

**3.2.3 Initialization**

None.

**3.2.4 Higher-Layer Triggered Events**

None.

**3.2.4.1 Requesting Body Properties**

When a client requests the **Message body**, it can either use the best body algorithm as specified in [MS-OXBBODY] or directly request one of the body properties specified in section 2.2.1.58. In the second case, the server SHOULD convert whatever body property it has for the message into the requested format. For example, if the message contains the body in the **PidTagBodyHtml** property (section 2.2.1.58.3) and the client requests the **PidTagBody** property (section 2.2.1.58.1), the server
converts the **PidTagBodyHtml** property to a **plain text** representation and returns this converted value.

The semantics for converting from one body format to another are implementation-dependent.

### 3.2.5 Message Processing Events and Sequencing Rules

#### 3.2.5.1 Receiving a RopOpenMessage ROP Request

The **Message object** returned by the **RopOpenMessage ROP** ([MS-OXCROPS] section 2.2.6.1) is used in subsequent ROPs, such as a **RopGetPropertiesSpecific ROP request** ([MS-OXCROPS] section 2.2.8.3). For information about which ROPs operate on Message objects, see the specific ROPs in [MS-OXCROPS].

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

A **RopOpenMessage** ROP MUST NOT succeed if a Message object with the specified ID does not exist or if the client has insufficient access rights to the folder in which the Message object is stored.

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

The response field **RecipientCount** indicates the current number of **recipients** (2) in the message. In addition, the server returns data for as many recipients (2) as will fit in the response buffer, in the order of the value of the **RowId** field. The data for each recipient (2) is encoded as an **OpenRecipientRow** structure in the **RecipientRows** field. The response field **RowCount** indicates how many recipients (2) are present in the **RecipientRows** field.

If a server does not recognize a **message class**, it reverts to acting on all but the last group, recursively, until a recognized form remains.

The following specific error codes apply to this ROP.

<table>
<thead>
<tr>
<th>Error code name</th>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ecNotFound</td>
<td>0x8004010F</td>
<td>The MID ([MS-OXCDATA] section 2.2.1.2) does not correspond to a message in the database. The folder corresponding to the FID ([MS-OXCDATA] section 2.2.1.1) entered in the <strong>ROP request buffer</strong> does not contain a message with the entered MID. The message is soft deleted and the client has not specified the <strong>OpenSoftDeleted</strong> flag as part of the <strong>OpenModeFlag</strong> field.</td>
</tr>
<tr>
<td>ecNullObject</td>
<td>0x000004b9</td>
<td>The value of the <strong>InputHandleIndex</strong> field on which this ROP was called does not refer to a <strong>Folder object</strong> or a <strong>Store object</strong>.</td>
</tr>
<tr>
<td>ecAccessDenied</td>
<td>0x80070005</td>
<td>The user does not have rights to open the message.</td>
</tr>
</tbody>
</table>

#### 3.2.5.2 Receiving a RopCreateMessage ROP Request

When processing the **RopCreateMessage ROP** ([MS-OXCROPS] section 2.2.6.2), the server MUST NOT commit the new **Message object** until it receives a **RopSaveChangesMessage ROP request** ([MS-OXCROPS] section 2.2.6.3).
The server SHOULD initialize the following properties before responding.

<table>
<thead>
<tr>
<th>Property name</th>
<th>Initial data</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PidTagImportance</strong> (section 2.2.1.11)</td>
<td>0x00000001</td>
</tr>
<tr>
<td><strong>PidTagMessageClass</strong> (section 2.2.1.3)</td>
<td>“IPM.Note”</td>
</tr>
<tr>
<td><strong>PidTagSensitivity</strong> (section 2.2.1.13)</td>
<td>0x00000000</td>
</tr>
<tr>
<td><strong>PidTagDisplayBcc</strong> ([MS-OXOMSG] section 2.2.1.7)</td>
<td>**</td>
</tr>
<tr>
<td><strong>PidTagDisplayCc</strong> ([MS-OXOMSG] section 2.2.1.8)</td>
<td>**</td>
</tr>
<tr>
<td><strong>PidTagDisplayTo</strong> ([MS-OXOMSG] section 2.2.1.9)</td>
<td>**</td>
</tr>
</tbody>
</table>
| **PidTagMessageFlags** (section 2.2.1.6) | 0x00000009; will be 0x00000409 (the mfEverRead flag combined by using the bitwise OR operation with the value 0x00000009) if the client does not explicitly set the read state. |<22>
| **PidTagMessageSize** (section 2.2.1.7) | See the **PidTagMessageSize** property in section 2.2.1.7                   |
| **PidTagHasAttachments** (section 2.2.1.2) | 0x00                                                                    |
| **PidTagTrustSender** (section 2.2.1.45) | 0x00000001                                                                  |
| **PidTagAccess** ([MS-OXCPRPT] section 2.2.1.1) | 0x00000003; will be 0x00000409 (the mfEverRead flag combined by using the bitwise OR operation with the value 0x00000009) if the client does not explicitly set the read state. |<23>
| **PidTagAccessLevel** ([MS-OXCPRPT] section 2.2.1.2) | 0x00000001                                                                  |
| **PidTagCreationTime** (section 2.2.2.3) | The time the RopCreateMessage ROP ([MS-OXCROPS] section 2.2.6.2) was processed |
| **PidTagLastModificationTime** (section 2.2.2.2) | Same as the **PidTagCreationTime** property                                  |
| **PidTagSearchKey** ([MS-OXCPRPT] section 2.2.1.9) | Server generated search key                                                  |
| **PidTagMessageLocaleId** (section 2.2.1.5) | The Logon object LocaleID.                                                   |
| **PidTagCreatorName** ([MS-OXPROPS] section 2.657) | Name of the creator.                                                         |
| **PidTagCreatorEntryId** (section 2.2.1.31) | Address book EntryID of the creator                                          |
| **PidTagLastModifierName** ([MS-OXCPRPT] section 2.2.1.5) | Same as the **PidTagCreatorName** property                                  |
| **PidTagLastModifierEntryId** (section 2.2.1.32) | Same as the **PidTagCreatorEntryId** property                               |
| **PidTagHasNamedProperties** (section 2.2.1.39) | 0x00                                                                       |
### Property name | Initial data
--- | ---
PidTagLocaleId ([MS-OXPROPS] section 2.775) | Same as the PidTagMessageLocaleId property

The following specific error code applies to this ROP.

<table>
<thead>
<tr>
<th>Error code name</th>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ecAccessDenied</td>
<td>0x80070005</td>
<td>The user does not have permissions to create this message. &lt;24&gt;</td>
</tr>
</tbody>
</table>

#### 3.2.5.3 Receiving a RopSaveChangesMessage ROP Request

After processing the RopSaveChangesMessage ROP ([MS-OXCR OPS] section 2.2.6.3), when the message has been successfully saved and all changes committed to the message store, the server determines the status of the Message object after the commit by the value of the SaveFlags flag as documented in section 2.2.3.3.1.

The response contains the MID ([MS-OXCDATA] section 2.2.1.2) of the committed message.

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

When the server receives multiple requests to open the same Message object, it returns a different handle and maintains a separate transaction for each. Any changes made on one transaction MUST NOT be visible to another transaction until the changes are committed via the RopSaveChangesMessage ROP. Once a transaction on one handle has been committed, the server MUST return "ecObjectModified" for RopSaveChangesMessage ROP requests on other handles and MUST NOT<25> allow those transactions to be committed, unless the client instructs the server to override previous changes with the ForceSave flag.

If pending changes include changes to read-only properties, the server MAY<26> return an error. The server sets the PidTagLocalCommitTime property (section 2.2.1.49) when the RopSaveChangesMessage ROP is processed.

The following specific error codes apply to this ROP.

<table>
<thead>
<tr>
<th>Error code name</th>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ecError</td>
<td>0x80004005 &lt;27&gt;</td>
<td>The message has been opened or previously saved as read only; changes cannot be saved.</td>
</tr>
<tr>
<td>ecObjectModified</td>
<td>0x80040109</td>
<td>The underlying data for this Message object was changed through another transaction context.</td>
</tr>
</tbody>
</table>

#### 3.2.5.4 Receiving a RopRemoveAllRecipients ROP Request

When processing the RopRemoveAllRecipients ROP ([MS-OXCR OPS] section 2.2.6.4), the server ignores the value of the Reserved field.
Until the server receives a **RopSaveChangesMessage ROP request** ([MS-OXCROPS] section 2.2.6.3) from the client, the server adheres to the following:

- The **PidTagRowid** property (section 2.2.1.38) and associated data of removed **recipients** (2) MUST NOT be returned as part of any subsequent handling of ROPs for the opened **Message object** on the same Message object handle.

- The changes made to the recipients (2) MUST NOT be included in the response buffer returned for ROP requests that apply to recipients (2) on different Message object handles.

The call to the **RopRemoveAllRecipients** ROP succeeds even if the Message object on which it is executed has no recipients (2).

The following specific error code applies to this ROP.

<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 value of the <strong>InputHandleIndex</strong> field on which this ROP was called does not refer to a Message object.</td>
</tr>
</tbody>
</table>

### 3.2.5.5 Receiving a **RopModifyRecipients** ROP Request

When processing the **RopModifyRecipients** ROP ([MS-OXCROPS] section 2.2.6.5), for each **recipient** (2) provided, the server locates its representation of the recipient (2) based on the value of the **RowId** field within the **RecipientRows** field. If the recipient (2) indicated by the value of the **RowId** field does not exist, the server creates a new recipient (2) with that **RowId** field value and applies the data from the request.

If the recipient (2) currently exists on the **Message object** and the value of **RecipientRowSize** field in the request buffer is nonzero, the server replaces all existing properties of the recipient (2) with the property values supplied in the request. If the value of the **RecipientRowSize** field in the **ModifyRecipientRow** structure within the **RecipientRows** field of the request buffer is 0x0000, then the server deletes the recipient (2) from the Message object.

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

- If a recipient (2) was deleted, its **RowId** field and associated data MUST NOT be returned as part of any subsequent handling of ROPs for the opened Message object.

- Any changes made to the recipients (2) MUST be included in the response buffer for any subsequent ROP requests that apply to recipients (2) for the same Message object handle.

- The changes made to the recipients (2) MUST NOT be included in the response buffer returned for ROP requests that apply to recipients (2) on different Message object handles.

The following specific error code applies to this ROP.

<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 value of the <strong>InputHandleIndex</strong> field on which this ROP was called does not refer to a Message object.</td>
</tr>
</tbody>
</table>
3.2.5.6 Receiving a RopReadRecipients ROP Request

The RopReadRecipients ROP ([MS-OXCROPS] section 2.2.6.6) is used to obtain information for all recipients (2) in the Message object, regardless of the number of recipients (2) on the message.

The server provides the recipient (2) information starting with the recipient (2) specified by the RowId field. If there is a recipient (2) with the given value of the RowId field, the server provides the information for that recipient (2) and as many recipients (2) as possible, limited by the number of actual recipients (2) in the message and the amount of recipient (2) information that fits in the response buffer.

When the value of the RowId field is 0x00000000, the server returns all recipients (2) for the message, beginning with the first recipient (2) and filling the response buffer with as many RecipientRow structures ([MS-OXCDATA] section 2.8.3) as will fit. If the message does not have recipients (2), the server returns the error ecNotFound.

The following specific error codes apply to this ROP.

<table>
<thead>
<tr>
<th>Error code name</th>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ecNotFound</td>
<td>0x8004010F</td>
<td>Recipient row RowId does not exist on the message.</td>
</tr>
<tr>
<td>ecBufferTooSmall</td>
<td>0x0000047D</td>
<td>Unable to fit at least one recipient (2) in the response buffer.</td>
</tr>
<tr>
<td>ecNullObject</td>
<td>0x000004B9</td>
<td>The InputHandleIndex on which this ROP was called does not refer to a Message object.</td>
</tr>
</tbody>
</table>

3.2.5.7 Receiving a RopReloadCachedInformation ROP Request

The following specific error code applies to the RopReloadCachedInformation ROP ([MS-OXCROPS] section 2.2.6.7).

<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 value of the InputHandleIndex field on which this ROP was called does not refer to a Message object.</td>
</tr>
</tbody>
</table>

3.2.5.8 Receiving a RopSetMessageStatus ROP Request

When processing the RopSetMessageStatus ROP ([MS-OXCROPS] section 2.2.6.8), the server modifies the bits on the PidTagMessageStatus property (section 2.2.1.8) specified by the MessageStatusMask field, preserving only those flags that are set in both the MessageStatusMask field and the MessageStatusFlags field, and clearing any other flags set only in the MessageStatusMask field.

The server immediately commits the changes to the Message object as if the Message object had been opened and the RopSaveChangesMessage ROP ([MS-OXCROPS] section 2.2.6.3) had been called, except that it changes only the PidTagMessageStatus property, not the PidTagChangeKey property ([MS-OXCFXICS] section 2.2.1.2.7), the PidTagLastModificationTime property (section 2.2.2.2), or any other property that is modified during the RopSaveChangesMessage ROP request. The following specific error code applies to this ROP.
Error code | Value | Meaning
--- | --- | ---
ecNullObject | 0x000004B9 | The value of the InputHandleIndex field on which this ROP was called does not refer to a Folder object.

3.2.5.9 Receiving a RopGetMessageStatus ROP Request

When processing the RopGetMessageStatus ROP ([MS-OXCROPS] section 2.2.6.9), the server MUST NOT require the Message object to be opened.

The following specific error code applies to this ROP.

Error code | Value | Meaning
--- | --- | ---
ecNullObject | 0x000004B9 | The value of the InputHandleIndex field on which this ROP was called does not refer to a Folder object.

3.2.5.10 Receiving a RopSetReadFlags ROP Request

The server immediately commits the changes to the Message objects as if the Message objects had been opened and the RopSaveMessageChanges ROP ([MS-OXCROPS] section 2.2.6.3) had been called, except that it only changes the PidTagMessageFlags property (section 2.2.1.6), not the PidTagChangeKey property ([MS-OXCFXICS] section 2.2.1.2.7), the PidTagLastModificationTime property (section 2.2.2.2), or any other property that is modified during a RopSaveChangesMessage ROP request ([MS-OXCROPS] section 2.2.6.3).

If the WantAsynchronous flag, as specified in section 2.2.3.10.1, is nonzero, the server SHOULD return a RopSetReadFlags ROP response ([MS-OXCROPS] section 2.2.6.10) but MAY return a RopProgress ROP response ([MS-OXCROPS] section 2.2.8.13) instead. If the SUPPORT_PROGRESS flag, as specified in [MS-OXCSTOR] section 2.2.1.1, is not set by the client in the OpenFlags field in the RopLogon ROP ([MS-OXCROPS] section 2.2.3.1), then the server SHOULD disable asynchronous processing of the RopSetReadFlags ROP and SHOULD NOT <28> return the RopProgress ROP whether or not the WantAsynchronous flag is set.

If the server has not received a SUPPORT_PROGRESS flag in the request buffer of the RopLogon ROP ([MS-OXCROPS] section 2.2.3.1), the server MUST disable asynchronous processing for the RopSetReadFlags ROP ([MS-OXCROPS] section 2.2.6.10), overriding any value of the WantAsynchronous flag. In this case, a RopProgress ROP SHOULD NOT <29> be sent. If the client does not pass the SUPPORT_PROGRESS flag, the server will process the entire RopSetReadFlags ROP request before returning a response to the client. If the client does pass a SUPPORT_PROGRESS flag, and the client also passes the WantAsynchronous flag, the server performs asynchronously and returns the RopProgress ROP to inform the client of the status of processing the RopSetReadFlags ROP.

If the server is unable to modify one or more of the Message objects that are specified in the MessageIds field, as specified in section 2.2.3.10.1, of the request buffer, then the server returns the PartialCompletion flag, as specified in section 2.2.3.10.2, in the response buffer.

The following specific error code applies to this ROP.
### 3.2.5.11 Receiving a RopSetMessageReadFlag ROP Request

If the **RopSetMessageReadFlag** ROP ([MS-OXCROPS] section 2.2.6.11) is performed in public folder mode as specified in section 2.2.3.11, the server finds the message associated with the LongTermID structure, as specified in [MS-OXCDATA] section 2.2.1.3.1, which is contained in the ClientData field in the request. The server finds the message by using the method specified in [MS-OXCSTOR] section 3.2.5.9.

The server immediately commits the changes to the **Message object** as if the Message object had been opened and the **RopSaveChangesMessage** ROP ([MS-OXCROPS] section 2.2.6.3) had been called, except that it only changes the PidTagMessageFlags property (section 2.2.1.6), not the PidTagChangeKey property ([MS-OXCFXICS] section 2.2.1.2.7), the PidTagLastModificationTime property (section 2.2.2.2), or any other property that is modified during a **RopSaveChangesMessage ROP request** ([MS-OXCROPS] section 2.2.6.3).

The following specific error code applies to this ROP.

<table>
<thead>
<tr>
<th>Error code name</th>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ecNullObject</td>
<td>0x00004B9</td>
<td>The value of the <strong>InputHandleIndex</strong> field on which this ROP was called does not refer to a <strong>Folder object</strong>.</td>
</tr>
</tbody>
</table>

### 3.2.5.12 Receiving a RopOpenAttachment ROP Request

The **handle** returned by the **RopOpenAttachment** ROP ([MS-OXCROPS] section 2.2.6.12) is used in subsequent ROPs, such as the **RopGetPropertiesSpecific** ROP ([MS-OXCROPS] section 2.2.8.3). For details about which ROPs operate on Attachment objects, see the sections for the ROPs in [MS-OXCROPS].

The following specific error codes apply to this ROP.

<table>
<thead>
<tr>
<th>Error code name</th>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ecNotFound</td>
<td>0x8004010F</td>
<td>The value of the <strong>AttachmentID</strong> field does not correspond to an attachment on the <strong>Message object</strong>.</td>
</tr>
<tr>
<td>ecAccessDenied</td>
<td>0x80070005</td>
<td>The user has insufficient privileges.</td>
</tr>
<tr>
<td>ecNullObject</td>
<td>0x000004B9</td>
<td>The value of the <strong>InputHandleIndex</strong> field on which this ROP was called does not refer to a <strong>Message object</strong>.</td>
</tr>
</tbody>
</table>
3.2.5.13 Receiving a RopCreateAttachment ROP Request

When processing the RopCreateAttachment ROP ([MS-OXCROPS] section 2.2.6.13), the server does not commit the new Attachment object until it receives a call to the RopSaveChangesAttachment ROP ([MS-OXCROPS] section 2.2.6.15).

The server MUST initialize the following properties before responding.

<table>
<thead>
<tr>
<th>Property name</th>
<th>Initial data</th>
</tr>
</thead>
<tbody>
<tr>
<td>PidTagAttachNumber (section 2.2.2.6)</td>
<td>Varies, depending on the number of existing attachments on the Message object</td>
</tr>
<tr>
<td>PidTagAttachSize (section 2.2.2.5)</td>
<td>0x00000040&lt;30&gt;</td>
</tr>
<tr>
<td>PidTagAccessLevel ([MS-OXCPRPT] section 2.2.1.2)</td>
<td>0x00000001&lt;31&gt;</td>
</tr>
<tr>
<td>PidTagRenderingPosition (section 2.2.2.16)</td>
<td>0xFFFFFFFF</td>
</tr>
<tr>
<td>PidTagCreationTime (section 2.2.2.3)</td>
<td>The time the RopCreateAttachment ROP ([MS-OXCROPS] section 2.2.6.13) was processed</td>
</tr>
<tr>
<td>PidTagLastModificationTime (section 2.2.2.2)</td>
<td>Same as the PidTagCreationTime property&lt;32&gt;</td>
</tr>
</tbody>
</table>

The following specific error codes apply to this ROP.

<table>
<thead>
<tr>
<th>Error code name</th>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ecMaxAttachmentExceeded</td>
<td>0x000004DB</td>
<td>The (server defined) maximum number of attachments for a message has been exceeded.</td>
</tr>
<tr>
<td>ecNullObject</td>
<td>0x000004B9</td>
<td>The value of the InputHandleIndex field on which this ROP was called does not refer to a Message object.</td>
</tr>
</tbody>
</table>

3.2.5.14 Receiving a RopSaveChangesAttachment ROP Request

After processing the RopSaveChangesAttachment ROP ([MS-OXCROPS] section 2.2.6.15), the server determines the status of the Attachment object after the commit by the values of the SaveFlags field as specified in section 2.2.3.3.1.

Although the server commits any pending changes to the Attachment object in the context of its containing Message object, the changes MUST NOT be committed to the database until the RopSaveChangesMessage ROP ([MS-OXCROPS] section 2.2.6.3) has been executed on the handle of the Message object.

The following specific error code applies to this ROP.

<table>
<thead>
<tr>
<th>Error code name</th>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
</table>
| ecNotSupported     | 0x80040102| The value of the SaveFlags field is not a supported combination as specified in section 2.2.3.3.1.  
The value of the InputHandleIndex field on which this ROP was called does not refer to an Attachment object. |
3.2.5.15 Receiving a RopDeleteAttachment ROP Request

The server recalculates the PidTagHasAttachments property (section 2.2.1.2) while processing the RopDeleteAttachment ROP ([MS-OXCROPS] section 2.2.6.14).

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

The following specific error codes apply to this ROP.

<table>
<thead>
<tr>
<th>Error code name</th>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ecNotFound</td>
<td>0x8004010F</td>
<td>The value of the AttachmentID field does not correspond to an attachment on the Message object.</td>
</tr>
<tr>
<td>ecAccessDenied</td>
<td>0x80070005</td>
<td>The user has insufficient privileges.</td>
</tr>
<tr>
<td>ecNullObject</td>
<td>0x000004B9</td>
<td>The value of the InputHandleIndex field on which this ROP was called does not refer to a Message object.</td>
</tr>
</tbody>
</table>

3.2.5.16 Receiving a RopOpenEmbeddedMessage ROP Request

If the embedded object does not exist, the client creates an Attachment object following the process specified in section 3.1.4.18. Once the attachment is created and its properties initialized, the client sends a RopOpenEmbeddedMessage ROP ([MS-OXCROPS] section 2.2.6.16) on the attachment to get a Message object handle. The returned handle is used in subsequent ROPs (similar to the one returned by the RopOpenMessage ROP ([MS-OXCROPS] section 2.2.6.1)). The server MUST NOT commit the Message object to the containing Attachment object until the RopSaveChangesMessage ROP ([MS-OXCROPS] section 2.2.6.3) is called with the Embedded Message object's handle.

The following specific error codes apply to this ROP.

<table>
<thead>
<tr>
<th>Error code name</th>
<th>Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ecAccessDenied</td>
<td>0x80070005</td>
<td>The user does not have permission to open or create this message.</td>
</tr>
<tr>
<td>ecNullObject</td>
<td>0x000004B9</td>
<td>The value of the InputHandleIndex field on which this ROP was called does not refer to an Attachment object.</td>
</tr>
<tr>
<td>ecUnknownCodePage</td>
<td>0x000003ef</td>
<td>The code page is unknown.</td>
</tr>
</tbody>
</table>

3.2.5.17 Receiving a RopGetAttachmentTable ROP Request

The Table object returned by the RopGetAttachmentTable ROP ([MS-OXCROPS] section 2.2.6.17) allows access to the properties of Attachment objects.

The following specific error codes apply to this ROP.
<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 value of the InputHandleIndex field on which this ROP was called does not refer to a Message object.</td>
</tr>
<tr>
<td>ecBusy</td>
<td>0x80040108</td>
<td>The server is too busy to complete the request.</td>
</tr>
</tbody>
</table>

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.
4 Protocol Examples

A user creates a new HTML-format e-mail, sets its subject to "abc123sample" and its body to "This is a sample body text". The user also adds two attachments, an HTML embedded image and a text file, adds a recipient (2), and then saves and closes the message.

4.1 Create Message

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

4.1.1 RopCreateMessage Request Buffer

0000: 06 00 00 01 ff 0f 01 00-00 00 00 f0 79 93 00
RopId: 0x06
LogonId: 0x00
InputHandleIndex: 0x00
OutputHandleIndex: 0x01
CodePageId: 0x0FFF
FolderId: 01 00 00 00 00 f0 79 93
AssociatedFlag: 0x00

4.1.2 RopCreateMessage Response Buffer

0000: 06 01 00 00 00 00 00
RopId: 0x06
OutputHandleIndex: 0x01
ReturnValue: 0x00000000
HasMessageId: 0x00

4.2 Name to Id Mapping

Before manipulating named properties on Message objects, the client needs to ask the server to map from the named properties to property identifiers, using the RopGetPropertyIdsFromNames ROP ([MS-OXCROPS] section 2.2.8.1) as described in [MS-OXCPRT] section 2.2.12.

4.3 Get Attachment Table

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

4.3.1 RopGetAttachmentTable Request Buffer

0000:21 00 00 01 00
RopId: 0x21
LogonId: 0x00
InputHandleIndex: 0x00
OutputHandleIndex: 0x01
TableFlags: 0x00 (Standard)

4.3.2 RopGetAttachmentTable Response Buffer
0000:21 01 00 00 00 00
RopId: 0x21
OutputHandleIndex: 0x01
ReturnValue: 0x00000000

4.4 Insert HTML Embedded Image
The client first creates the Attachment object on the Message object, then sets its properties and commits the changes.

4.4.1 RopCreateAttachment Request Buffer
0000: 23 00 00 01
RopId: 0x23
LogonId: 0x00
InputHandleIndex: 0x00
OutputHandleIndex: 0x01

4.4.2 RopCreateAttachment Response Buffer
0000: 23 01 00 00 00 00 00 00-00 00
RopId: 0x23
OutputHandleIndex: 0x01
ReturnValue: 0x00000000
AttachmentID: 0x00000000

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

<table>
<thead>
<tr>
<th>Property tag</th>
<th>Property name</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x37050003</td>
<td>PidTagAttachMethod (section 2.2.9)</td>
<td>0x0000000001</td>
</tr>
<tr>
<td>0x370B0003</td>
<td>PidTagRenderingPosition (section 2.2.16)</td>
<td>0xFFFFFFFF</td>
</tr>
</tbody>
</table>
To set the contents of the embedded image, the client uses four ROPs.

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

The **RopSetStreamSize** ROP ([MS-OXCROPS] section 2.2.9.7) with the size of image file data.

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

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

### 4.4.4 RopSaveChangesAttachment Request Buffer

<table>
<thead>
<tr>
<th>RopId</th>
<th>0x25</th>
</tr>
</thead>
<tbody>
<tr>
<td>LogonId</td>
<td>0x00</td>
</tr>
<tr>
<td>ResponseHandleIndex</td>
<td>0x01</td>
</tr>
<tr>
<td>InputHandleIndex</td>
<td>0x00</td>
</tr>
<tr>
<td>SaveFlags</td>
<td>0x0A (KeepOpenReadWrite)</td>
</tr>
</tbody>
</table>

### 4.4.5 RopSaveChangesAttachment Response Buffer

<table>
<thead>
<tr>
<th>RopId</th>
<th>0x25</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResponseHandleIndex</td>
<td>0x01</td>
</tr>
<tr>
<td>ReturnValue</td>
<td>0x00000000</td>
</tr>
</tbody>
</table>
4.4.6 Releasing Attachment Object

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

4.5 Attach Text File

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

4.5.1 RopCreateAttachment Request Buffer

```
0000:23 00 00 03
RopId: 0x23
LogonId: 0x00
InputHandleIndex: 0x00
OutputHandleIndex: 0x03
```

4.5.2 RopCreateAttachment Response Buffer

```
0000: 23 03 00 00 00 01 00-00 00
RopId: 0x23
OutputHandleIndex: 0x03
ReturnValue: 0x00000000
AttachmentID: 0x00000001
```

4.5.3 Setting Properties

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

<table>
<thead>
<tr>
<th>Property tag</th>
<th>Property name</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x37050003</td>
<td>PidTagAttachMethod</td>
<td>&quot;0x00000001&quot;</td>
</tr>
<tr>
<td>0x370B0003</td>
<td>PidTagRenderingPosition</td>
<td>&quot;0xFFFFFFFF&quot;</td>
</tr>
<tr>
<td>0x7FFD0003</td>
<td>PidTagAttachmentFlags</td>
<td>&quot;0x00000000&quot;</td>
</tr>
<tr>
<td>0x3001001F</td>
<td>PidTagDisplayName</td>
<td>&quot;test.txt&quot;</td>
</tr>
<tr>
<td>0x7FFA0003</td>
<td>PidTagAttachmentLinkId</td>
<td>&quot;0x00000000&quot;</td>
</tr>
<tr>
<td>0x37140003</td>
<td>PidTagAttachFlags</td>
<td>&quot;0x00000000&quot;</td>
</tr>
<tr>
<td>Property tag</td>
<td>Property name</td>
<td>Data</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------------------------------------</td>
<td>---------------------------------------------------------------------</td>
</tr>
<tr>
<td>0x7FFE000B</td>
<td>PidTagAttachmentHidden (section 2.2.2.24)</td>
<td>&quot;0x00&quot;</td>
</tr>
<tr>
<td>0x3707001F</td>
<td>PidTagAttachLongFilename (section 2.2.2.10)</td>
<td>&quot;test.txt&quot;</td>
</tr>
<tr>
<td>0x3704001F</td>
<td>PidTagAttachFilename (section 2.2.2.11)</td>
<td>&quot;test.txt&quot;</td>
</tr>
<tr>
<td>0x3703001F</td>
<td>PidTagAttachExtension (section 2.2.2.12)</td>
<td>&quot;.txt&quot;</td>
</tr>
<tr>
<td>0x30070040</td>
<td>PidTagCreationTime (section 2.2.2.3)</td>
<td>&quot;2008/02/12 22:28:34.636&quot;</td>
</tr>
<tr>
<td>0x30080040</td>
<td>PidTagLastModificationTime (section 2.2.2.2)</td>
<td>&quot;2008/02/12 22:28:50.112&quot;</td>
</tr>
<tr>
<td>0x37090102</td>
<td>PidTagAttachRendering (section 2.2.2.17)</td>
<td>3,512 bytes representing a Windows Metafile Format (WMF) file. For more information on WMF, see [MS-WMF].</td>
</tr>
</tbody>
</table>

To set the contents of the attachment, the client uses four ROPs:

1. The **RopOpenStream** ROP ([MS-OXCROPS] section 2.2.9.1) with **PidTagAttachDataBinary** (section 2.2.2.7).
2. The **RopSetStreamSize** ROP ([MS-OXCROPS] section 2.2.9.7) with the size of the file data.
3. The **RopWriteStream** ROP request ([MS-OXCROPS] section 2.2.9.3) with the actual file contents.
4. The **RopRelease** ROP ([MS-OXCROPS] section 2.2.15.3) for the handle returned from the **RopOpenStream** ROP.

### 4.5.4 RopSaveChangesAttachment Request Buffer

```
0000: 25 00 02 01 0A
```

**RopId**: 0x25

**LogonId**: 0x00

**ResponseHandleIndex**: 0x02

**InputHandleIndex**: 0x01

**SaveFlags**: 0x0A *(KeepOpenReadWrite)*

### 4.5.5 RopSaveChangesAttachment Response Buffer

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

**RopId**: 0x25

**ResponseHandleIndex**: 0x02

**ReturnValue**: 0x00000000
4.5.6 Releasing Attachment Object

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

4.6 Setting Message Properties

The client uses the RopSetProperties ROP ([MS-OXCROPS] section 2.2.8.6) to set all the necessary properties.

The HTML body, stored in the PidTagBodyHtml property (section 2.2.158.3), is the following:

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

4.7 Adding Recipients

4.7.1 RopModifyRecipients Request Buffer

```hex
0000:0e 00 08 0c 00 03 00 fe-ff 03 00 00 39 1f 00 ff

0010:39 1f 00 fe 39 03 00 71-3a 03 00 05 39 1f 00 f6
0020:05 03 00 fd 5f 03 00 ff-5f 03 00 de 5f 03 00 df
0030:5f 02 01 f7 5f 01 00 00-00 00 00 01 27 01 51 06
0040:5a 00 55 73 65 72 32 00-75 00 73 00 65 00 72 00
0050:70-00 00 00 00 00 00 00 00 00 75 00 73 00 65 00
0060:70 00 00 00 00 00 75 00 00 00 00 75 00 00 00 00
0070:00 00 00 00 00 00 00 00 00 00 00 75 00 00 00 00
0080:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0090:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00a0:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00b0:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00c0:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00d0:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00e0:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00f0:00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

RopId: 0x0E

LogonId: 0x00

InputHandleIndex: 0x08
ColumnCount: 0x000C (number of following RecipientColumns)

PidTagObjectType ([MS-OXCPRPT] section 2.2.1.7): 0x0FFE0003

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

PidTagAddressBookDisplayNamePrintable (section 2.2.1.30): 0x39FF001F

PidTagSmtpAddress ([MS-OXOABK] section 2.2.3.21): 0x39FE001F

PidTagSendInternetEncoding ([MS-OXOABK] section 2.2.3.19): 0x3a710003

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

PidTagRecipientDisplayName ([MS-OXPROPS] section 2.900): 0x5FF6001F

PidTagRecipientFlags ([MS-OXOCAL] section 2.2.4.10.1): 0x5FFD0003

PidTagRecipientTrackStatus ([MS-OXOCAL] section 2.2.4.10.2): 0x5FFF0003

Unspecified property: 0x5FDE0003 [PtypInteger32 ([MS-OXCDATA] section 2.11.1)]

PidTagRecipientOrder ([MS-OXPROPS] section 2.903): 0x5FDF0003

PidTagRecipientEntryId ([MS-OXPROPS] section 2.901): 0x5FF70102

RowCount: 0x0001 (number of following ModifyRecipientRows)

RowId: 0x00000000

RecipientType: 0x01 (primary recipient)

RecipientRowSize: 0x0127 (bytes in following RecipientRow)

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

AddressPrefixUsed: 0x5A (present because Type=X500DN)

DisplayType: 0x00 (present because Type=X500DN)

EmailAddress: User2 (present because Type=X500DN)

DisplayName: user2 (present because D is set)

SimpleDisplayName: user2 (present because I is set)

RecipientColumnCount: 0x000C (matches ColumnCount)

StandardPropertyRow:

Flag: 0x00

ValueArray: (property order defined by RecipientColumns)

PidTagObjectType: 0x00000006

PidTagDisplayType: 0x00000000

PidTagAddressBookDisplayNamePrintable (section 2.2.1.30): user2

PidTagSmtpAddress: user2@szfkuk-dom.extest.microsoft.com

PidTagSendInternetEncoding: 0
PidTagDisplayTypeEx: 0x40000000
PidTagRecipientDisplayName: user2
PidTagRecipientFlags: 0x00000001
PidTagRecipientTrackStatus: 0x00000000
Unspecified property (0x5FDE0003 [PtypInteger32]): 0x00000000
PidTagRecipientOrder: 0x00000000
PidTagRecipientEntryId: 0x007C and the subsequent 124 (0x7C) bytes

4.7.2 RopModifyRecipients Response Buffer
0000: 0e 08 00 00 00 00
RopId: 0x0E
InputHandleIndex: 0x08
ReturnValue: 0x000000

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

4.8.1 RopSaveChangesMessage Request Buffer
0000: 0c 00 00 01 0a
RopId: 0x0C
LogonId: 0x00
ResponseHandleIndex: 0x00
InputHandleIndex: 0x01
SaveFlags: 0x0A (KeepOpenReadWrite)

4.8.2 RopSaveChangesMessage Response Buffer
0000: 0c 00 00 00 00 01 01 00 00 00 f0 86 39
RopId: 0x0C
ResponseHandleIndex: 0x00
ReturnValue: 0x00000000
InputHandleIndex: 0x01
MessageId: 01 00 00 00 00 F0 86 39
4.9 Releasing Message Object

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

5.1 Security Considerations for Implementers

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

5.2 Index of Security Parameters

None.
6 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include 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
- Microsoft Outlook 2024 Preview

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: Exchange 2007 changes properties PidTagAccess and PidTagLastModificationTime. Exchange 2010, Exchange 2013, Exchange 2016, and Exchange 2019 change properties PidTagAccess, PidTagAccessLevel, PidTagChangeKey, PidTagCreationTime, PidTagLastModificationTime, PidTagLastModifierName, PidTagSearchKey, PidTagHasAttachments, PidTagMessageFlags, PidTagMessageSize, PidTagMessageStatus, and PidTagAttachSize.

<2> Section 2.2.1.1: Exchange 2010, Exchange 2013, Exchange 2016, and Exchange 2019 do not support the PidTagObjectType property.

<3> Section 2.2.1.1: Exchange 2010, Exchange 2013, Exchange 2016, and Exchange 2019 do not support the PidTagRecordKey property.

<4> Section 2.2.1.6: Exchange 2007 does not set the mfEverRead flag when the mfRead flag is set.

<5> Section 2.2.1.45: The PidTagTrustSender property (section 2.2.1.45) is not supported in Exchange 2010, Exchange 2013, Exchange 2016, and Exchange 2019.

[MS-OXCMSG] - v20240416
Message and Attachment Object Protocol
Copyright © 2024 Microsoft Corporation
Release: April 16, 2024
<6> Section 2.2.1.58.2: Exchange 2003 and Exchange 2007 do not support the PidTagNativeBody property.

<7> Section 2.2.1.60.6: Exchange 2003, Exchange 2007, and the initial release version of Exchange 2010 do not support the NeedsRescan flag.

<8> Section 2.2.1.60.6: Exchange 2003, Exchange 2007, and the initial release version of Exchange 2010 do not support the PendingRescan flag.

<9> Section 2.2.2.1: Exchange 2010, Exchange 2013, Exchange 2016, and Exchange 2019 do not support the PidTagObjectType property.

<10> Section 2.2.2.20: Office Outlook 2003 only correctly detects MacBinary I and MacBinary II.

<11> Section 2.2.3.1: Exchange 2010, Exchange 2013, Exchange 2016, and Exchange 2019 can output unexpected results when using the RopOpenMessage ROP ([MS-OXCROPS] section 2.2.6.1) when Client Access Services are deployed on an Exchange server that does not also have a mailbox message store installed.

<12> Section 2.2.3.1.1: Exchange 2010, Exchange 2013, Exchange 2016, and Exchange 2019 return read/write messages when the user does not have write permissions.

<13> Section 2.2.3.3.1: Exchange 2010, Exchange 2013, Exchange 2016, and Exchange 2019 ignore the KeepOpenReadOnly flag.

<14> Section 2.2.3.3.1: The value of ForceSave is 0x0C in Microsoft Exchange Server 2007 Service Pack 3 (SP3).

<15> Section 2.2.3.10.2: Exchange 2010, Exchange 2013 Exchange 2016, and Exchange 2019 return a zero value.

<16> Section 2.2.3.15: Exchange 2010, Exchange 2013, Exchange 2016, and Exchange 2019 return a GeneralFailure error if pending changes include changes to read-only properties.

<17> Section 2.2.3.16.1: Exchange 2010, Exchange 2013, Exchange 2016, and Exchange 2019 open the message for both reading and writing.

<18> Section 2.2.3.18: Exchange 2010, Exchange 2013, Exchange 2016, and Exchange 2019 do not support the RopGetValidAttachments ROP ([MS-OXCROPS] section 2.2.6.18).

<19> Section 3.1.1.1: 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 Outlook 2013 Service Pack 1 (SP1).

<20> Section 3.2.1.1: Exchange 2003, Exchange 2007, Exchange 2010, the initial release version of Exchange 2013 do not support the session context cookie. The session context cookie was introduced in Microsoft Exchange Server 2013 Service Pack 1 (SP1).

<21> Section 3.2.5.2: Exchange 2013, Exchange 2016, and Exchange 2019 do not initialize the following properties: PidTagCreatorName ([MS-OXPROPS] section 2.657), PidTagCreatorEntryId (section 2.2.1.31), PidTagLastModifierName ([MS-OXCPRPT] section 2.2.1.5), PidTagLastModifierEntryId (section 2.2.1.32), PidTagLastModificationTime (section 2.2.2.2), PidTagMessageLocaleId (section 2.2.1.5) and PidTagLocaleId ([MS-OXPROPS] section 2.775).

<22> Section 3.2.5.2: In Exchange 2007, the initial MUST be 0x00000009.

<23> Section 3.2.5.2: Exchange 2013, Exchange 2016, and Exchange 2019 initialize the PidTagAccess property ([MS-OXCPRPT] section 2.2.1.1) to 0x00000007.

<24> Section 3.2.5.2: In Exchange 2007 and Exchange 2010, the error code is ecNoCreateRight with value 0x0000004F.
Section 3.2.5.3: Exchange 2010, Exchange 2013 and Exchange 2016 return Success for RopSaveChangesMessage ROP requests ([MS-OXCROPS] section 2.2.6.3) when a previous request has already been committed against the Message object, even though the changes to the object are not actually committed to the server message store.

Section 3.2.5.3: Exchange 2010, Exchange 2013, Exchange 2016, and Exchange 2019 return a GeneralFailure error if pending changes include changes to read-only properties PidTagMessageSize, PidTagAccess, PidTagAccessLevel, PidTagObjectType, PidTagRecordKey, PidTagMessageStatus, and PidTagHasAttachments.


Section 3.2.5.13: Exchange 2013, Exchange 2016, and Exchange 2019 set the PidTagAttachSize property (section 2.2.2.5) to 0x00000000.

Section 3.2.5.13: Exchange 2013, Exchange 2016, and Exchange 2019 set the PidTagAccessLevel property ([MS-OXCPRPT] section 2.2.1.2) to 0x00000000.

Section 3.2.5.13: Exchange 2013, Exchange 2016, and Exchange 2019 set the PidTagLastModificationTime property (section 2.2.2.2) to a value that is within 100 nanoseconds of the value of the PidTagCreationTime property (section 2.2.2.3).
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>2.2.1.57 PidTagClientActivelyEditingUntil Property</td>
<td>Added new section for PidTagClientActivelyEditingUntil Property.</td>
<td>Major</td>
</tr>
<tr>
<td>Appendix A: Product Behavior</td>
<td>Updated list of supported products.</td>
<td>Major</td>
</tr>
</tbody>
</table>
8 Index

A

Abstract data model
- client 54
- server 60

Abstract data model types - client
- global 54
- per mailbox 54
- per Message object 54

Abstract data model types - server
- global 61
- per mailbox 61
- per Message object 61

Add recipients example
- RopModifyRecipients request buffer 77
- RopModifyRecipients response buffer 79

Applicability 17

Attach text file example
- overview 75
- releasing Attachment object 77
- RopCreateAttachment request buffer 75
- RopCreateAttachment response buffer 75
- RopSaveChangesAttachment request buffer 76
- RopSaveChangesAttachment response buffer 76
- setting properties 75

Attachment object properties
- general properties 35
- MIME properties 40
- PidTagAttachAdditionalInformation property 38
- PidTagAttachDataBinary property 36
- PidTagAttachDataObject property 36
- PidTagAttachEncoding property 38
- PidTagAttachExtension property 37
- PidTagAttachFilename property 37
- PidTagAttachFlags property 38
- PidTagAttachLongFilename property 36
- PidTagAttachLongPathname property (section 2.2.2.13 37, section 2.2.2.26 39)
- PidTagAttachmentFlags property 39
- PidTagAttachmentHidden property 39
- PidTagAttachmentLinkId property 39
- PidTagAttachMethod property 36
- PidTagAttachNumber property 35
- PidTagAttachPathname property 37
- PidTagAttachRendering property 38
- PidTagAttachSize property 35
- PidTagAttachTag property 37
- PidTagAttachTransportName property 38
- PidTagCreationTime property 35
- PidTagDisplayName property 35
- PidTagLastModificationTime property 35
- PidTagRenderingPosition property 37
- PidTagTextAttachmentCharset property 39

B

Body Message object properties 29

C

Capability negotiation 17

Change tracking 85

Client
- abstract data model 54
- initialization 54
- other local events 60
- timer events 60
- timers 54

Client - abstract data model types
- global 54
- per mailbox 54
- per Message object 54

Client - higher-layer triggered events
- accessing the attachments table 56
- adding, deleting, or modifying a recipient 55
- creating a Message object 55
- creating an attachment 56
- creating an embedded message 56
- deleting an attachment 56
- getting message status 55
- linking a Contact object 57
- opening a Message object 55
- opening an attachment 56
- opening an Embedded Message object 56
- reading recipients 55
- reloading Message object header info 55
- removing all recipients 55
- saving an embedded message 56
- saving changes on a Message object 55
- saving changes on an Attachment object 56
- setting Attachment object content 56
- setting Message object read status 55
- setting message status 55

Client - message processing
- sending a RopCreateAttachment ROP request 59
- sending a RopCreateMessage ROP request 57
- sending a RopGetAttachmentTable ROP request 60
- sending a RopGetMessageStatus ROP request 58
- sending a RopGetPropertiesSpecific ROP request 60
- sending a RopModifyRecipients ROP request 58
- sending a RopOpenAttachment ROP request 59
- sending a RopOpenEmbeddedMessage ROP request 60
- sending a RopOpenMessage ROP request 57
- sending a RopReadRecipients ROP request 58
- sending a RopRemoveAllRecipients ROP request 58
- sending a RopSaveChangesAttachment ROP request 60
- sending a RopSaveChangesMessage ROP request 57
- sending a RopSetMessageStatus ROP request 58
- sending a RopSetProperties ROP request 59
- sending a RopSetReadFlags ROP request 59

Client - sequencing rules
- sending a RopCreateAttachment ROP request 59
- sending a RopCreateMessage ROP request 57
- sending a RopGetAttachmentTable ROP request 60
- sending a RopGetMessageStatus ROP request 58
- sending a RopGetPropertiesSpecific ROP request 60
- sending a RopModifyRecipients ROP request 58
- sending a RopOpenAttachment ROP request 59
sending a RopOpenEmbeddedMessage ROP request 60
sending a RopOpenMessage ROP request 57
sending a RopReadRecipients ROP request 58
sending a RopRemoveAllRecipients ROP request 58
sending a RopSaveChangesAttachment ROP request 60
sending a RopSaveChangesMessage ROP request 57
sending a RopSetMessageStatus ROP request 58
sending a RopSetProperties ROP request 59
sending a RopSetReadFlags ROP request 59

Contact linking Message object properties 31

Create message example
overview 72
RopCreateMessage request buffer 72
RopCreateMessage response buffer 72

Data model - abstract
client 54
server 60

Examples
name to id mapping 72
releasing Message object 80
setting message properties 77

Examples – add recipients
RopModifyRecipients request buffer 77
RopModifyRecipients response buffer 79

Examples – attach text file
overview 75
releasing Attachment object 77
RopCreateAttachment request buffer 75
RopCreateAttachment response buffer 75
RopSaveChangesAttachment request buffer 76
RopSaveChangesAttachment response buffer 76
setting properties 75

Examples – create message
overview 72
RopCreateMessage request buffer 72
RopCreateMessage response buffer 72

Examples – get attachment table
overview 72
RopGetAttachmentTable request buffer 72
RopGetAttachmentTable response buffer 73

Examples – insert HTML embedded image
overview 73
RopCreateAttachment request buffer 73
RopCreateAttachment response buffer 73
RopSaveChangesAttachment request buffer 74
RopSaveChangesAttachment response buffer 74
setting properties 73

Examples – save message
overview 79
RopSaveChanges request buffer 79
RopSaveChanges response buffer 79

FAI messages 16
Fields - vendor-extendible 17

G
General Attachment object properties 35
General Message object properties 18
Get attachment table example
overview 72
RopGetAttachmentTable request buffer 72
RopGetAttachmentTable response buffer 73

Global abstract data model type
client 54
server 61
Glossary 10

Higher-layer triggered events
server 61
Higher-layer triggered events - client
accessing the attachments table 56
adding, deleting, or modifying a recipient 55
creating a Message object 55
creating an attachment 56
creating an embedded message 56
deleting an attachment 56
getting message status 55
linking a Contact object 57
opening a Message object 55
opening an attachment 56
opening an Embedded Message object 56
reading recipients 55
reload Message object header info 55
removing all recipients 55
saving an embedded message 56
saving changes on a Message object 55
saving changes on an Attachment object 56
setting Attachment object content 56
setting Message object read status 55
setting message status 55

Implementer – security considerations 81
Index of security parameters 81
Informative references 15

Initialization
client 54
server 61

Insert HTML embedded image example
overview 73
RopCreateAttachment request buffer 73
RopCreateAttachment response buffer 73
RopSaveChangesAttachment request buffer 74
RopSaveChangesAttachment response buffer 74
setting properties 73
Introduction 10

Message attachments 16
Message object properties
body properties 29
contact linking properties 31
general properties 18
PidLidAgingDontAgeMe property 25
<table>
<thead>
<tr>
<th>Property Name</th>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>PidLidClassification property</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>PidLidClassificationDescription property</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>PidLidClassified property</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>PidLidCommonEnd property</td>
<td>23</td>
<td>23</td>
</tr>
<tr>
<td>PidLidCommonStart property</td>
<td>23</td>
<td>23</td>
</tr>
<tr>
<td>PidLidCurrentVersion property</td>
<td>26</td>
<td>26</td>
</tr>
<tr>
<td>PidLidCurrentVersionName property</td>
<td>26</td>
<td>26</td>
</tr>
<tr>
<td>PidLidInfoPathFormName property</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>PidLidPrivate property</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td>PidLidSideEffects property</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td>PidNameAcceptLanguage property</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>PidNameAddressBookDisplayNamePrintable property</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>PidTagAddressBookDisplayNamePrintable property</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>PidTagAlternateRecipientAllowed property</td>
<td>26</td>
<td>26</td>
</tr>
<tr>
<td>PidTagAutoForward property</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>PidTagAutoForwarded property</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>PidTagCreatorEntryId property</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>PidTagCreatorName property</td>
<td>28</td>
<td>28</td>
</tr>
<tr>
<td>PidTagHasAttachments property</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>PidTagHasNamedProperties property</td>
<td>26</td>
<td>26</td>
</tr>
<tr>
<td>PidTagImportance property</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>PidTagInternetReferences property</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>PidTagLastModifierEntryId property</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>PidTagLocalCommitTime property</td>
<td>28</td>
<td>28</td>
</tr>
<tr>
<td>PidTagMessageAttachments property</td>
<td>28</td>
<td>28</td>
</tr>
<tr>
<td>PidTagMessageClass property</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>PidTagMessageCodepage property</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>PidTagMessageFlags property</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>PidTagMessageLocaleId property</td>
<td>19</td>
<td>19</td>
</tr>
<tr>
<td>PidTagMessageRecipients property</td>
<td>28</td>
<td>28</td>
</tr>
<tr>
<td>PidTagMessageSize property</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>PidTagMessageStatus property</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>PidTagLastModifiedEntryId property</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>PidTagNormalizedSubject property</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>PidTagPriority property</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>PidTagPuportedSenderDomain property</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>PidTagRead property</td>
<td>29</td>
<td>29</td>
</tr>
<tr>
<td>PidTagRecipientDisplayName property</td>
<td>29</td>
<td>29</td>
</tr>
<tr>
<td>PidTagRecipientEntryId property</td>
<td>29</td>
<td>29</td>
</tr>
<tr>
<td>PidTagRecipientOrder property</td>
<td>26</td>
<td>26</td>
</tr>
<tr>
<td>PidTagResponsibility property</td>
<td>26</td>
<td>26</td>
</tr>
<tr>
<td>PidTagRowid property</td>
<td>26</td>
<td>26</td>
</tr>
<tr>
<td>PidTagSensitivity property</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td>PidTagSubject property</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>PidTagSubject property</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>PidTagTnefPrefix property</td>
<td>21</td>
<td>21</td>
</tr>
<tr>
<td>PidTagTrustSender property</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>PidTagTrustSender property</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>PidTagTrustServer property</td>
<td>27</td>
<td>27</td>
</tr>
<tr>
<td>PidTagTrustServer property</td>
<td>27</td>
<td>27</td>
</tr>
</tbody>
</table>

**Message object ROPs**

- RopCreateAttachment ROP: 50
- RopCreateMessage ROP: 43
- RopDeleteAttachment ROP: 51
- RopGetAttachmentTable ROP: 52
- RopGetMessageStatus ROP: 47
- RopGetValidAttachments ROP: 53
- RopModifyRecipients ROP: 45
- RopOpenAttachment ROP: 50
- RopOpenEmbeddedMessage ROP: 52
- RopOpenMessage ROP: 41
- RopReadRecipients ROP: 46
- RopReloadCachedInformation ROP: 46
- RopSaveChangesAttachment ROP: 51
- RopSaveChangesMessage ROP: 43
- RopSetMessageReadFlag ROP: 49
- RopSetMessageStatus ROP: 47
- RopSetReadFlags ROP: 48

**Message Object ROPs message**

- sending a RopCreateAttachment ROP request: 59
- sending a RopCreateMessage ROP request: 57
- sending a RopGetAttachmentTable ROP request: 60
- sending a RopGetMessageStatus ROP request: 58
- sending a RopGetPropertiesSpecific ROP request: 60
- sending a RopModifyRecipients ROP request: 58
- sending a RopOpenAttachment ROP request: 59
- sending a RopOpenEmbeddedMessage ROP request: 60
- sending a RopOpenMessage ROP request: 57
- sending a RopReadRecipients ROP request: 58
- sending a RopRemoveAllRecipients ROP request: 58
- sending a RopSaveChangesAttachment ROP request: 60
- sending a RopSaveChangesMessage ROP request: 57
- sending a RopSetMessageStatus ROP request: 58
- sending a RopSetProperties ROP request: 59
- sending a RopSetReadFlags ROP request: 59

**Message processing - server**

- receiving a RopCreateAttachment ROP request: 69
- receiving a RopCreateMessage ROP request: 62
- receiving a RopDeleteAttachment ROP request: 70
- receiving a RopGetAttachmentTable ROP request: 70
- receiving a RopGetMessageStatus ROP request: 67
- receiving a RopModifyRecipients ROP request: 65
- receiving a RopOpenAttachment ROP request: 68
- receiving a RopOpenEmbeddedMessage ROP request: 70
- receiving a RopOpenMessage ROP request: 62
- receiving a RopReadRecipients ROP request: 66
- receiving a RopReloadCachedInformation ROP request: 66
- receiving a RopRemoveAllRecipients ROP request: 64
- receiving a RopSaveChangesAttachment ROP request: 69
- receiving a RopSaveChangesMessage ROP request: 64
- receiving a RopSetMessageReadFlag ROP request: 68
- receiving a RopSetMessageStatus ROP request: 66
- receiving a RopSetReadFlags ROP request: 67

**Message recipients**

- 16

**Messages**

- FAI messages: 16
- message attachments: 16
- Message Object ROPs: 41
- message recipients: 16
- transport: 18
**MIME Attachment object properties**

**Name to id mapping example**

**Normative references**

**Other local events**

*client* 54

*server* 61

**Overview**

**Overview (synopsis)**

**Parameters - security index**

Per mailbox abstract data model type

*client* 54

*server* 61

Per Message object abstract data model type

*client* 54

*server* 61

PidLiDAgingDonAgeMe Message object property

PidLiDCategories Message object property

PidLiDClassification Message object property

PidLiDClassificationDescription Message object property

PidLiDClassified Message object property

PidLiDCommonEnd Message object property

PidLiDCommonStart Message object property

PidLiDCurrentVersion Message object property

PidLiDCurrentVersionName Message object property

PidLiDInfoPathFormName Message object property

PidLiDPrivate Message object property

PidLiDSideEffects Message object property

PidLiDSmartNoAttach Message object property

PidLiDAcceptLanguage Message object property

PidLiDContentType Message object property

PidLiDContentClass Message object property

PidLiDContentBase Message object property

PidLiDKeywords Message object property

PidTagAddressBookDisplayNamePrintable Message object property

PidTagAlternateRecipientAllowed Message object property

PidTagAttachAdditionalInformation Attachment object property

PidTagAttachDataBinary Attachment object property

PidTagAttachDataObject Attachment object property

PidTagAttachEncoding Attachment object property

PidTagAttachExtension Attachment object property

PidTagAttachFilename Attachment object property

PidTagAttachFlags Attachment object property

PidTagAttachLongFilename Attachment object property

PidTagAttachLongPathname Attachment object property

PidTagAttachmentFlags Attachment object property

PidTagAttachmentHidden Attachment object property

PidTagAttachmentLinkId Attachment object property

PidTagAttachMethod Attachment object property

PidTagAttachNumber Attachment object property

PidTagAttachPathname Attachment object property

PidTagAttachRendering Attachment object property

PidTagAttachSize Attachment object property

PidTagAttachTag Attachment object property

PidTagAttachTransportName Attachment object property

PidTagAutoForwardComment Message object property

PidTagAutoForwarded Message object property

PidTagCreationTime Attachment object property

PidTagCreatorEntryId Message object property

PidTagCreatorName Message object property

PidTagDisplayName Attachment object property

PidTagHasAttachments Message object property

PidTagHasNamedProperties Message object property

PidTagImportance Message object property

PidTagInternetReferences Message object property

PidTagLastModificationTime Attachment object property

PidTagLastModifierEntryId Message object property

PidTagLocalCommitTime Message object property

PidTagMessageAttachments Message object property

PidTagMessageClass Message object property

PidTagMessageCodepage Message object property

PidTagMessageFlags Message object property

PidTagMessageLocaleId Message object property

PidTagMessageFlags Message object property

PidTagMessageRecipients Message object property

PidTagMessageRecipients Message object property

PidTagMessageSize Message object property

PidTagMessageStatus Message object property

PidTagMimeSkeleton Message object property

PidTagNormalizedSubject Message object property

PidTagPriority Message object property

PidTagPriorityMessage object property

PidTagPurposedSenderDomain Message object property

PidTagRecipientOrder Message object property

PidTagRecipientEntryId Message object property

PidTagRecipientDisplayName Message object property

PidTagRead Message object property

PidTagRecipientDisplayMessage object property

PidTagRecipientEntryId Message object property

PidTagRecipientOrder Message object property

PidTagRenderingPosition Attachment object property

PidTagResponsibility Message object property

PidTagRowid Message object property

PidTagSensitivity Message object property
PidTagStoreEntryId Message object property 27
PidTagSubject Message object property 27
PidTagSubjectPrefix Message object property 21
PidTagTextAttachmentCharset Attachment object property 39
PidTagTrustCorrelationKey Message object property 25
PidTagTrustSender Message object property 27
Preconditions 17
Prerequisites 17
Product behavior 82

R

References 14
   informative 15
   normative 14
Relationship to other protocols 16
Releasing Message object example 80
Retention and archive Message object properties 32
RopCreateAttachment Message object ROP 50
RopCreateMessage Message object ROP 43
RopDeleteAttachment Message object ROP 41
RopGetAttachmentTable Message object ROP 52
RopGetMessageStatus Message object ROP 47
RopGetValidAttachments Message object ROP 53
RopModifyRecipients Message object ROP 45
RopOpenAttachment Message object ROP 50
RopOpenEmbeddedMessage Message object ROP 52
RopOpenMessage Message object ROP 41
RopReadRecipients Message object ROP 46
RopReloadCachedInformation Message object ROP 46
RopRemoveAllRecipients Message object ROP 44
RopSaveChangesAttachment Message object ROP 51
RopSaveChangesMessage Message object ROP 43
RopSetMessageReadFlag Message object ROP 49
RopSetMessageStatus Message object ROP 47
RopSetReadFlags Message object ROP 48

S

Save message example
   overview 79
   RopSaveChanges request buffer 79
   RopSaveChanges response buffer 79
Security
   implementer considerations 81
   parameter index 81
Sequencing rules - client
   sending a RopCreateAttachment ROP request 59
   sending a RopCreateMessage ROP request 57
   sending a RopGetAttachmentTable ROP request 60
   sending a RopGetMessageStatus ROP request 58
   sending a RopGetPropertiesSpecific ROP request 60
   sending a RopModifyRecipients ROP request 58
   sending a RopOpenAttachment ROP request 59
   sending a RopOpenEmbeddedMessage ROP request 60
   sending a RopOpenMessage ROP request 57
   sending a RopReadRecipients ROP request 58
   sending a RopRemoveAllRecipients ROP request 58
   sending a RopSaveChangesAttachment ROP request 60

Sequencing rules - server
   receiving a RopCreateAttachment ROP request 69
   receiving a RopCreateMessage ROP request 62
   receiving a RopDeleteAttachment ROP request 70
   receiving a RopGetAttachmentTable ROP request 70
   receiving a RopGetMessageStatus ROP request 67
   receiving a RopModifyRecipients ROP request 65
   receiving a RopOpenAttachment ROP request 68
   receiving a RopOpenEmbeddedMessage ROP request 70
   receiving a RopOpenMessage ROP request 62
   receiving a RopReadRecipients ROP request 66
   receiving a RopReloadCachedInformation ROP request 66
   receiving a RopRemoveAllRecipients ROP request 64
   receiving a RopSaveChangesAttachment ROP request 69
   receiving a RopSaveChangesMessage ROP request 64
   receiving a RopSetMessageReadFlag ROP request 64
   receiving a RopSetMessageStatus ROP request 68
   receiving a RopSetProperties ROP request 66
   receiving a RopSetReadFlags ROP request 70
   sending a RopSaveChangesMessage ROP request 57
   sending a RopSetMessageStatus ROP request 58
   sending a RopSetProperties ROP request 59
   sending a RopSetReadFlags ROP request 59

Server abstract data model 60
   higher-layer triggered events 61
   initialization 61
   other local events 71
   time events 71
   timers 61
Server - abstract data model types
   global 61
   per mailbox 61
   per Message object 61
Server - message processing
   receiving a RopCreateAttachment ROP request 69
   receiving a RopCreateMessage ROP request 62
   receiving a RopDeleteAttachment ROP request 70
   receiving a RopGetAttachmentTable ROP request 70
   receiving a RopGetMessageStatus ROP request 67
   receiving a RopModifyRecipients ROP request 65
   receiving a RopOpenAttachment ROP request 68
   receiving a RopOpenEmbeddedMessage ROP request 70
   receiving a RopOpenMessage ROP request 62
   receiving a RopReadRecipients ROP request 66
   receiving a RopReloadCachedInformation ROP request 66
   receiving a RopRemoveAllRecipients ROP request 64
   receiving a RopSaveChangesAttachment ROP request 69
   receiving a RopSaveChangesMessage ROP request 64
   receiving a RopSetMessageReadFlag ROP request 64
   receiving a RopSetMessageStatus ROP request 68
   receiving a RopSetProperties ROP request 66
   receiving a RopSetReadFlags ROP request 70
   sending a RopSaveChangesMessage ROP request 57
   sending a RopSetMessageStatus ROP request 58
   sending a RopSetProperties ROP request 59
   sending a RopSetReadFlags ROP request 59

[MS-OXCMSP] - v20240416
Message and Attachment Object Protocol
Copyright © 2024 Microsoft Corporation
Release: April 16, 2024

90 / 91
receiving a RopSetReadFlags ROP request 67
receiving a RopCreateAttachment ROP request 69
receiving a RopCreateMessage ROP request 62
receiving a RopDeleteAttachment ROP request 70
receiving a RopGetAttachmentTable ROP request 70
receiving a RopGetMessageStatus ROP request 67
receiving a RopModifyRecipients ROP request 65
receiving a RopOpenAttachment ROP request 68
receiving a RopOpenEmbeddedMessage ROP request 70
receiving a RopOpenMessage ROP request 62
receiving a RopReadRecipients ROP request 66
receiving a RopReloadCachedInformation ROP request 66
receiving a RopRemoveAllRecipients ROP request 64
receiving a RopSaveChangesAttachment ROP request 69
receiving a RopSaveChangesMessage ROP request 64
receiving a RopSetMessageReadFlag ROP request 68
receiving a RopSetMessageStatus ROP request 66
receiving a RopSetReadFlags ROP request 67

Setting message properties example 77

Standards assignments 17

T

Timer events
client 60
server 71

Timers
client 54
server 61

Tracking changes 85

Transport 18

Triggered events - client
accessing the attachments table 56
adding, deleting, or modifying a recipient 55
creating a Message object 55
creating an attachment 56
creating an embedded message 56
deleting an attachment 56
getting message status 55
linking a Contact object 57
opening a Message object 55
opening an attachment 56
opening an Embedded Message object 56
reading recipients 55
reload Message object header info 55
removing all recipients 55
saving an embedded message 56
saving changes on a Message object 55
saving changes on an Attachment object 56
setting Attachment object content 56
setting Message object read status 55
setting message status 55

Triggered events - higher-layer
server 61

V