[MS-OXCMSG]:

Message and Attachment Object Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.

- **Copyrights**. This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.

- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.

- **Patents**. Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft Open Specification Promise or the Community Promise. If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.

- **Trademarks**. The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.

- **Fictitious Names**. The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications do not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them. Certain Open Specifications are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

Preliminary Documentation. This Open Specification provides documentation for past and current releases and/or for the pre-release version of this technology. This Open Specification is final documentation for past or current releases as specifically noted in the document, as applicable; it is preliminary documentation for the pre-release versions. Microsoft will release final documentation in connection with the commercial release of the updated or new version of this technology. As the documentation may change between this preliminary version and the final version of this technology, there are risks in relying on preliminary documentation. To the extent that you incur additional

development obligations or any other costs as a result of relying on this preliminary documentation, you do so at your own risk.

Revision Summary

Date	Revision History	Revision Class	Comments
4/4/2008	0.1		Initial Availability.
4/25/2008	0.2		Revised and updated property names and other technical content.
6/27/2008	1.0		Initial Release.
8/6/2008	1.01		Revised and edited technical content.
9/3/2008	1.02		Revised and edited technical content.
10/1/2008	1.03		Revised and edited technical content.
12/3/2008	1.04		Updated IP notice.
4/10/2009	2.0		Updated technical content and applicable product releases.
7/15/2009	3.0	Major	Revised and edited for technical content.
11/4/2009	4.0.0	Major	Updated and revised the technical content.
2/10/2010	5.0.0	Major	Updated and revised the technical content.
5/5/2010	6.0.0	Major	Updated and revised the technical content.
8/4/2010	7.0	Major	Significantly changed the technical content.
11/3/2010	8.0	Major	Significantly changed the technical content.
3/18/2011	9.0	Major	Significantly changed the technical content.
8/5/2011	10.0	Major	Significantly changed the technical content.
10/7/2011	11.0	Major	Significantly changed the technical content.
1/20/2012	12.0	Major	Significantly changed the technical content.
4/27/2012	13.0	Major	Significantly changed the technical content.
7/16/2012	14.0	Major	Significantly changed the technical content.
10/8/2012	15.0	Major	Significantly changed the technical content.
2/11/2013	16.0	Major	Significantly changed the technical content.
7/26/2013	17.0	Major	Significantly changed the technical content.
11/18/2013	18.0	Major	Significantly changed the technical content.
2/10/2014	18.0	No Change	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	19.0	Major	Significantly changed the technical content.
7/31/2014	19.0	No Change	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	19.1	Minor	Clarified the meaning of the technical content.

Date	Revision History	Revision Class	Comments
3/16/2015	20.0	Major	Significantly changed the technical content.
5/26/2015	21.0	Major	Significantly changed the technical content.

Table of Contents

1   Introduction

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

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in [RFC2119]. Sections 1.5 and 1.9 are also normative but do not contain those terms. All other sections and examples in this specification are informative.

1.1   Glossary

The following terms are specific to this document:

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 2.2.1.1.1.

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 (1), 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 addressee on a **Message object** that is not visible to recipients of the Message object.

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

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

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].

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 (1). 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 context-dependent term commonly overloaded with three meanings. Note that much of the industry literature concerning RPC technologies uses this term interchangeably for any of the three meanings. Following are the three definitions: (*) The runtime environment providing remote procedure call facilities. The preferred usage for this meaning is "RPC runtime". (*) The pattern of request and response message exchange between two parties (typically, a client and a server). The preferred usage for this meaning is "RPC exchange". (*) A single message from an exchange as defined in the previous definition. The preferred usage for this term is "RPC message". For more information about RPC, 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 (2) 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: An object that is used to view properties for a collection of objects of a specific type, such as a **Message object** or a **Folder object**. A Table object is structured in a row and

column format with each row representing an object and each column representing a property of the object.

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 vector graphics format for Windows-compatible computers. Windows Metafile Format is used primarily as a clip-art format 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-OXCPERM] Microsoft Corporation, "Exchange Access and Operation Permissions Protocol".

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

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

[MS-OXCRPC] Microsoft Corporation, "Wire Format Protocol".

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

[MS-OXCTABL] Microsoft Corporation, "Table Object Protocol".

[MS-OXOABK] Microsoft Corporation, "Address Book 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-OXPROPS] Microsoft Corporation, "Exchange Server Protocols Master Property List".

[MS-OXRTFCP] Microsoft Corporation, "Rich Text Format (RTF) Compression Algorithm".

[MS-OXRTFEX] Microsoft Corporation, "Rich Text Format (RTF) Extensions Algorithm".

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

[MS-WMF] Microsoft Corporation, "Windows Metafile Format".

[RFC1521] Borenstein, N., and Freed, N., "MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies", RFC 1521, September, 1993, http://www.rfc-editor.org/rfc/rfc1521.txt

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

[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

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

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

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

1.2.2 Informative References

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

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

[MS-OXODOC] Microsoft Corporation, "Document Object Protocol".

[MS-OXOJRNL] Microsoft Corporation, "Journal Object Protocol".

[MS-OXONOTE] Microsoft Corporation, "Note Object Protocol".

[MS-OXOPOST] Microsoft Corporation, "Post Object Protocol".

[MS-OXORSS] Microsoft Corporation, "RSS Object Protocol".

[MS-OXOSMIME] Microsoft Corporation, "S/MIME Email Object Algorithm".

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

[MS-OXOTASK] Microsoft Corporation, "Task-Related Objects Protocol".

[MS-OXOUM] Microsoft Corporation, "Voice Mail and Fax Objects Protocol".

[MS-OXPROTO] Microsoft Corporation, "Exchange Server Protocols System Overview".

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-OXCPRPT], 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]

- Journal Object Protocol, as described in [MS-OXOJRNL]

- RSS Object Protocol, as described in [MS-OXORSS]

- Post Object Protocol, as described in [MS-OXOPOST]

- Short Message Service (SMS) and Multimedia Messaging Service (MMS) Object Protocol, as described in [MS-OXOSMMS]

- Document Object Protocol, as described in [MS-OXODOC]

- S/MIME Email Object Protocol, as described in [MS-OXOSMIME]

- Voice Mail and Fax Objects Protocol, as described in [MS-OXOUM]

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

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-OXCFOLD]. For more information on folder storage and organization, see [MS-OXCSTOR].

1.6   Applicability Statement

The Message and Attachment Object Protocol can be used as the basis for different types of personal information messages, 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 2.2.2.2)

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

The server computes this property from the **mfHasAttach** flag of the **PidTagMessageFlags** property ([MS-OXPROPS] section 2.780).

2.2.1.3 PidTagMessageClass Property

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

The **PidTagMessageClass** property ([MS-OXPROPS] section 2.776) denotes the specific type of the Message object. It determines the set of properties defined for the message, the kind of information the message conveys, and how to handle the message.

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

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 ("."). Each group specifies a type of object. A **message class** of "IPM.Note" denotes a standard Message object, and a message class of "Remote.IPM.Note" indicates a **header message object**.

2.2.1.4 PidTagMessageCodepage Property

Type: **PtypInteger32**, unsigned

The **PidTagMessageCodepage** property ([MS-OXPROPS] section 2.777) specifies the **code page** used to encode the **non-Unicode** string properties on this Message object. The Folder object code page is used if this property is set to 0x00000000.

2.2.1.5 PidTagMessageLocaleId Property

Type: **PtypInteger32**, unsigned

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

2.2.1.6 PidTagMessageFlags Property

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

The **PidTagMessageFlags** property ([MS-OXPROPS] section 2.780) specifies the status of the Message object. Set to zero or to a bitwise OR of one or more of the values from the following tables.

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

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

Flag name	Value	Description
mfResend	0x00000080	The message includes a request for a resend operation with a non-delivery report. For more details, see [MS-OXOMSG] section 3.2.4.5.

These flags are always read-only for the client.

Flag name	Value	Description
mfUnmodified	0x00000002	The message has not been modified since it was first saved (if unsent) or it was delivered (if sent).
mfSubmitted	0x00000004	The message is marked for sending as a result of a call to the **RopSubmitMessage** ROP
mfHasAttach	0x00000010	The message has at least one attachment. This flag corresponds to the message's **PidTagHasAttachments** property (section 2.2.1.2).
mfFromMe	0x00000020	The user receiving the message was also the user who sent the message.
mfFAI	0x00000040	The message is an FAI message.
mfNotifyRead	0x00000100	The user who sent the message has requested notification when a recipient (1) first reads it.
mfNotifyUnread	0x00000200	The user who sent the message has requested notification when a recipient (1) deletes it before reading or the Message object expires.
mfEverRead	0x00000400	The message has been read at least once. This flag is set or cleared by the server whenever the **mfRead** flag is set or cleared.<4> Clients SHOULD ignore this flag.
mfInternet	0x00002000	The incoming message arrived over the Internet and originated either outside the organization or from a source the gateway does not consider trusted.
mfUntrusted	0x00008000	The incoming message arrived over an external link other than X.400 or the Internet. It originated either outside the organization or from a source the gateway does not consider trusted.

The **PidTagMessageFlags** property 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.

Flag name	Value	Description
msRemoteDownload	0x00001000	The message has been marked for downloading from the remote message store to the local client.
msInConflict	0x00000800	This is a conflict resolve message as specified in [MS-OXCFXICS] section 3.1.5.6.2.1. This is a read-only value for the client.
msRemoteDelete	0x00002000	The message has been marked for deletion at the remote message store without downloading to the local client.

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 2.1021) 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 2.801) 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 2.727) 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.

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

2.2.1.12 PidTagPriority Property

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

The **PidTagPriority** property ([MS-OXPROPS] section 2.860) 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.

Value	Description
0x00000001	Urgent
0x00000000	Normal
0xFFFFFFFF	Not urgent

2.2.1.13 PidTagSensitivity Property

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

The **PidTagSensitivity** property ([MS-OXPROPS] section 2.999) indicates the sender's assessment of the sensitivity of the Message object. The value of this property is one of the following.

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

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.

Flag name	Value	Description
seOpenToDelete	0x00000001	The client opens the Message object when deleting.
seNoFrame	0x00000008	No UI is associated with the Message object.
seCoerceToInbox	0x00000010	The client moves the Message object to the **Inbox folder** when moving or copying to a Folder object with the **PidTagContainerClass** property ([MS-OXOCAL] section 2.2.11.1) set to "IPF.Note". For more details about the **PidTagContainerClass** property, see [MS-OXOSFLD] section 2.2.8.
seOpenToCopy	0x00000020	The client opens the Message object when copying to another folder.
seOpenToMove	0x00000040	The client opens the Message object when moving to another folder.
seOpenForCtxMenu	0x00000100	The client opens the Message object when displaying context-sensitive commands, such as a context menu, to the end user.
seCannotUndoDelete	0x00000400	The client cannot undo a delete operation; this flag MUST NOT be set unless the **seOpenToDelete** flag is set.
seCannotUndoCopy	0x00000800	The client cannot undo a copy operation; this flag MUST NOT be set unless the **seOpenToCopy** flag is set.
seCannotUndoMove	0x00001000	The client cannot undo a move operation; this flag MUST NOT be set unless the **seOpenToMove** flag is set.
seHasScript	0x00002000	The Message object contains end-user script.
seOpenToPermDelete	0x00004000	The client opens the Message object to permanently delete it.

2.2.1.17   PidNameKeywords Property

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

The **PidNameKeywords** property ([MS-OXPROPS] section 2.444) 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 2.605) 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 2.604) 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 2.738) 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   PidLidInfoPathFormName 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 2.792) 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 2.1035) 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 2.505) 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 2.646) 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 2.754) 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 2.568) 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 2.920) 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 2.928) 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 2.708) 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 2.891) 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 2.406) 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 2.865) 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 2.1016) 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 2.1039) 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.

Value	Meaning
0x00000000	Message was not delivered through a trusted transport channel.
0x00000001	Message was delivered through a trusted transport channel.

2.2.1.46 PidTagSubject Property

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

The **PidTagSubject** property ([MS-OXPROPS] section 2.1021) 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 2.784) 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 2.407) 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 2.761) 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 2.408) 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 2.647) 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 2.774) 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 2.867) 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 2.888) 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 2.889) contains an EntryID that identifies the **Address Book object** associated with a recipient (2).

2.2.1.56 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.

2.2.1.56.1 PidTagBody Property

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

The **PidTagBody** property ([MS-OXPROPS] section 2.609) 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.

2.2.1.56.2 PidTagNativeBody Property

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

The **PidTagNativeBody** property ([MS-OXPROPS] section 2.794) 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.

Value	Meaning
0x00000000	**Undefined body**
0x00000001	Plain text body
0x00000002	**Rich Text Format (RTF)** compressed body

Value	Meaning
0x00000003	**HTML** body
0x00000004	**Clear-signed body**

2.2.1.56.3   PidTagBodyHtml Property

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

The **PidTagBodyHtml** property ([MS-OXPROPS] section 2.612) contains the HTML body as specified in [RFC2822] section 2.3.

2.2.1.56.4   PidTagRtfCompressed Property

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

The **PidTagRtfCompressed** property ([MS-OXPROPS] section 2.930) contains an RTF body compressed as specified in [MS-OXRTFCP].

2.2.1.56.5   PidTagRtfInSync Property

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

The **PidTagRtfInSync** property ([MS-OXPROPS] section 2.931) is set to "TRUE" (0x01) if the RTF body has been synchronized with the contents in the **PidTagBody** property (section 2.2.1.56.1).

2.2.1.56.6   PidTagInternetCodepage Property

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

The **PidTagInternetCodepage** property ([MS-OXPROPS] section 2.735) indicates the code page used for the **PidTagBody** property (section 2.2.1.56.1) or the **PidTagBodyHtml** property (section 2.2.1.56.3).

2.2.1.56.7   PidTagBodyContentId Property

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

The **PidTagBodyContentId** property ([MS-OXPROPS] section 2.610) 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.

2.2.1.56.8   PidTagBodyContentLocation Property

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

The **PidTagBodyContentLocation** property ([MS-OXPROPS] section 2.611) 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].

2.2.1.56.9 PidTagHtml Property

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

The **PidTagHtml** property ([MS-OXPROPS] section 2.722) contains the message body text in HTML format.

2.2.1.57 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**.

2.2.1.57.1 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 0	1	2	3	4	5	6	7	8	9	2 0	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.

2.2.1.57.2 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 2.2.1.57.1). This property can also include names not referenced in the **PidLidContactLinkEntry** property.

2.2.1.57.3 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 2.2.1.57.2), separated by a semicolon and a space ("; ").

2.2.1.57.4 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	1 0	1	2	3	4	5	6	7	8	9	2 0	1	2	3	4	5	6	7	8	9	3 0	1
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 may be a variable size and are null-terminated, as specified in [MS-OXOABK] section 2.2.3.5.

2.2.1.58 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.

2.2.1.58.1 PidTagArchiveTag Property

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

The **PidTagArchiveTag** property ([MS-OXPROPS] section 2.852) 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.

2.2.1.58.2 PidTagPolicyTag Property

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

The **PidTagPolicyTag** property ([MS-OXPROPS] section 2.852) 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.

2.2.1.58.3 PidTagRetentionPeriod Property

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

The **PidTagRetentionPeriod** property ([MS-OXPROPS] section 2.923) 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.

2.2.1.58.4   PidTagStartDateEtc Property

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

The **PidTagStartDateEtc** property ([MS-OXPROPS] section 2.1014) has the following structure.

Length in bytes	Meaning	Notes
4	**Default retention period**	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.)
8	**Start date**	The date, in UTC, from which the age of the Message object is calculated.

The **PidTagStartDateEtc** property can be present only on Message objects.

2.2.1.58.5   PidTagRetentionDate Property

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

The **PidTagRetentionDate** property ([MS-OXPROPS] section 2.921) specifies the date, in UTC, after which a Message object is expired by the server. The **PidTagRetentionDate** property can be present only on Message objects, not on folders. If the property is not present, the Message object never expires. The **PidTagRetentionDate** property can be set by both client and server.

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

▪   When the Message object has a specific retention policy:

PidTagRetentionDate = **PidTagMessageDeliveryTime** ([MS-OXOMSG] section 2.2.3.9) + **PidTagRetentionPeriod** (section 2.2.1.58.3). If the **PidTagMessageDeliveryTime** property does not exist, the **PidTagCreationTime** property (section 2.2.2.3) is used.

▪   When the Message object has the default retention policy:

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

2.2.1.58.6   PidTagRetentionFlags Property

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

The **PidTagRetentionFlags** property ([MS-OXPROPS] section 2.922) 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.

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

2.2.1.58.7   PidTagArchivePeriod Property

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

The **PidTagArchivePeriod** property ([MS-OXPROPS] section 2.571) 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.

2.2.1.58.8   PidTagArchiveDate Property

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

The **PidTagArchiveDate** property ([MS-OXPROPS] section 2.570) 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:

PidTagArchiveDate = start date + **PidTagArchivePeriod** (section 2.2.1.58.7)

The start date is obtained from the last eight bytes of the **PidTagStartDateEtc** property (section 2.2.1.58.4).

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 2.753) 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 2.667) 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 2.599) 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 2.594) 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 2.580) 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 2.581) 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 2.592) represents the way the contents of an attachment are accessed. This property is set to one of the following values.

Flag name	Value	Description
afNone	0x00000000	The attachment has just been created.
afByValue	0x00000001	The **PidTagAttachDataBinary** property (section 2.2.2.7) contains the attachment data.
afByReference	0x00000002	The **PidTagAttachLongPathname** property (section 2.2.2.13) contains a fully qualified path identifying the attachment **To recipients** with access to a common file server.
afByReferenceOnly	0x00000004	The **PidTagAttachLongPathname** property contains a fully qualified path identifying the attachment.
afEmbeddedMessage	0x00000005	The attachment is an embedded message that is accessed via the **RopOpenEmbeddedMessage** ROP ([MS-OXCROPS] section 2.2.6.16).
afStorage	0x00000006	The **PidTagAttachDataObject** property (section 2.2.2.8) contains data in an application-specific format.

2.2.2.10   PidTagAttachLongFilename Property

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

The **PidTagAttachLongFilename** property ([MS-OXPROPS] section 2.586) 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 2.584) 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 2.583) 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 2.587) 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 2.595) 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 2.600) 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.

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

2.2.2.16   PidTagRenderingPosition Property

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

The **PidTagRenderingPosition** property ([MS-OXPROPS] section 2.903) 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 2.598) 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 2.585) 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.

Flag name	Value	Description
attInvisibleInHtml	0x00000001	The Attachment object is not available to be rendered in HTML.
attInvisibleInRtf	0x00000002	The Attachment object is not available to be rendered in RTF.
attRenderedInBody	0x00000004	The Attachment object is referenced and rendered within the HTML body of the associated Message object. More details are specified in the **PidTagBodyHtml** property (section 2.2.1.56.3).

2.2.2.19   PidTagAttachTransportName Property

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

The **PidTagAttachTransportName** property ([MS-OXPROPS] section 2.601) 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 2.582) 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<10> correctly detect MacBinary I, MacBinaryII, and MacBinary III formats.

2.2.2.21   PidTagAttachAdditionalInformation Property

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

The **PidTagAttachAdditionalInformation** property ([MS-OXPROPS] section 2.576) 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 2.591) 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 2.589) 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 2.590) 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 2.1032) 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 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.

Type	Property name	Content
PtypString	**PidTagAttachMimeTag** ([MS-OXPROPS] section 2.593)	The **Content-Type** header.
PtypString	**PidTagAttachContentId** ([MS-OXPROPS] section 2.578)	A content identifier unique to this Message object that matches a corresponding "cid:" URI scheme reference in the HTML body of the Message object.
PtypString	**PidTagAttachContentLocation** ([MS-OXPROPS] section 2.579)	A relative or full URI that matches a corresponding reference in the HTML body of the Message object.
PtypString	**PidTagAttachContentBase** ([MS-OXPROPS] section 2.577)	The base of a relative URI. MUST be set if the **PidTagAttachContentLocation** property contains a relative URI.
PtypString	**PidTagAttachPayloadClass** ([MS-OXPROPS] section 2.596)	The class name of an object that can display the contents of the message.

Type	Property name	Content
PtypString	**PidTagAttachPayloadProviderGuidString** ([MS-OXPROPS] section 2.597)	The GUID of the software application that can display the contents of the message.
PtypString	**PidNameAttachmentMacContentType** ([MS-OXPROPS] section 2.368)	The **Content-Type** header of the Macintosh attachment.
PtypBinary	**PidNameAttachmentMacInfo** ([MS-OXPROPS] section 2.369)	The headers and resource fork data associated with the Macintosh attachment.

2.2.3 Message Object ROPs

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

2.2.3.1 RopOpenMessage 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.<11> 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.

Flag name	Value	Description
ReadOnly	0x00	Message will be opened as read-only.
ReadWrite	0x01	Message will be opened for both reading and writing.
BestAccess	0x03	Open for read/write if the user has write **permissions** for the folder, read-only if

Flag name	Value	Description
		not.<12>
OpenSoftDeleted	0x04	Open a **soft deleted** Message object if available.

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.

Value	Meaning
0x00	No named properties are defined for this Message object.
Nonzero	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.

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.

Value	Meaning
0x01	**Primary recipient**
0x02	**Carbon copy (Cc) recipient**
0x03	**Blind carbon copy (Bcc) recipient**

Valid flags are as follows.

Value	Meaning
0x10	When resending a previous failure, this flag indicates that this recipient (1) did not successfully receive the message on the previous attempt.
0x80	When resending a previous failure, this flag indicates that this recipient (1) did successfully receive the message on the previous attempt.

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.

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

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.

Value	Meaning
0x00	This is the last byte in the buffer.

Value	Meaning
Nonzero	The **MessageId** field follows beginning with the next byte in the buffer.

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 one of the flags specified in the following table. Other flags are not valid and are ignored by the server.

Flag name	Value	Description
KeepOpenReadOnly<13>	0x01	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-OXCPRPT] section 2.2.1.2.
KeepOpenReadWrite	0x02	The client requests that the server commit the changes. The server either returns an error and leaves the Message object open with unchanged access level or returns a success code and keeps the Message object open with read/write access.
ForceSave	0x04<14>	The client requests that the server commit the changes. The server either returns an error and leaves the Message object open with unchanged access level or returns a success code and keeps the Message object open with read/write access. The **ecObjectModified** error code is not valid when this flag is set; the server overwrites any changes instead.

2.2.3.3.2 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-OXCPRPT] 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 2.787) 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-OXCDATA] 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.

Flag name	Value	Description
rfDefault	0x00	The server sets the read flag and sends the receipt.
rfSuppressReceipt	0x01	The user requests that any pending read receipt be canceled; the server sets the **mfRead** bit.
rfReserved	0x0A	Ignored by the server.
rfClearReadFlag	0x04	Server clears the **mfRead** bit; the client MUST include the **rfSuppressReceipt** bit with this flag.
rfGenerateReceiptOnly	0x10	The server sends a read receipt if one is pending, but does not change the **mfRead** bit.
rfClearNotifyRead	0x20	The server clears the **mfNotifyRead** bit but does not send a read receipt.
rfClearNotifyUnread	0x40	The server clears the **mfNotifyUnread** bit but does not send a nonread receipt.

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.

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

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.

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

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<16> 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.

Value name	Value	Description
ReadOnly	0x00	Open the message as read-only.<17>
ReadWrite	0x01	Open the message for both reading and writing.

Value name	Value	Description
Create	0x02	Create the attachment if it does not already exist and open the message for both reading and writing.

2.2.3.16.2 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**, **ColumnCount**, **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.

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

2.2.3.17.2 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 request 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-OXCMAPIHTTP] 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**:

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

3.1.1.3 Per Message Object

A Message object is represented by the **MessageObject** ADM 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 2.2.1.57.

- **PidLidContactLinkEntry** (section 2.2.1.57.1)

- **PidLidContactLinkName** (section 2.2.1.57.3)

- **PidLidContactLinkSearchKey** (section 2.2.1.57.4)

- **PidLidContacts** (section 2.2.1.57.2)

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.

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

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 2.2.3.3.1.

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 2.787) 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 **RopSetProperties** 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:

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

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 2.2.1.56. 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 2.2.1.56.3) and the client requests the **PidTagBody** property (section 2.2.1.56.1), 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.

Error code name	Value	Meaning
ecNotFound	0x8004010F	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 ROP request buffer does not contain a message with the entered MID. The message is soft deleted and the client has not specified the **OpenSoftDeleted** flag as part of the **OpenModeFlag** field.

Error code name	Value	Meaning
ecNullObject	0x000004b9	The value of the **InputHandleIndex** field on which this ROP was called does not refer to a Folder object or a Store object.
ecAccessDenied	0x80070005	The user does not have rights to open the message.

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<21> initialize the following properties before responding.

Property name	Initial data
PidTagImportance (section 2.2.1.11)	0x00000001
PidTagMessageClass (section 2.2.1.3)	"IPM.Note"
PidTagSensitivity (section 2.2.1.13)	0x00000000
PidTagDisplayBcc ([MS-OXOMSG] section 2.2.1.7)	""
PidTagDisplayCc ([MS-OXOMSG] section 2.2.1.8)	""
PidTagDisplayTo ([MS-OXOMSG] section 2.2.1.9)	""
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<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]	Server generated search key

Property name	Initial data
section 2.2.1.9)	
PidTagMessageLocaleId (section 2.2.1.5)	The Logon object LocaleID.
PidTagCreatorName ([MS-OXPROPS] section 2.647)	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
PidTagLocaleId ([MS-OXPROPS] section 2.763)	Same as the **PidTagMessageLocaleId** property

The following specific error code applies to this ROP.

Error code name	Value	Meaning
ecAccessDenied	0x80070005	The user does not have permissions to create this message.<24>

3.2.5.3 Receiving a RopSaveChangesMessage ROP Request

After processing the **RopSaveChangesMessage** ROP ([MS-OXCROPS] 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.

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

3.2.5.4 Receiving a RopRemoveAllRecipients ROP Request

When processing the **RopRemoveAllRecipients** ROP ([MS-OXCROPS] 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.

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

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.

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

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.

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

3.2.5.7 Receiving a RopReloadCachedInformation ROP Request

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

Error code name	Value	Meaning
ecNullObject	0x000004b9	The value of the **InputHandleIndex** field on which this ROP was called does not refer to a Message object.

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 name	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 name	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 2.2.1.1.1, 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.

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

3.2.5.11 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.

Error code name	Value	Meaning
ecNullObject	0x00004B9	The value of the **InputHandleIndex** field on which this ROP was called does not refer to a Message object.

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.

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

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.

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

The following specific error codes apply to this ROP.

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

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.

Error code name	Value	Meaning
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.

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

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.

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

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.

Error code name	Value	Meaning
ecNullObject	0x000004B9	The value of the **InputHandleIndex** field on which this ROP was called does not refer to a Message object.
ecBusy	0x80040108	The server is too busy to complete the request.

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

4 Protocol Examples

A user creates a new HTML-format e-mail, sets its subject to "abc123sample" and its body to "This is a sample body text". The user also adds two attachments, an HTML embedded image and a text file, adds a recipient (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-OXCPRPT] section 2.2.12.

4.3 Get Attachment Table

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

4.3.1 RopGetAttachmentTable Request Buffer

0000:21 00 00 01 00

RopId: 0x21

LogonId: 0x00

InputHandleIndex: 0x00

OutputHandleIndex: 0x01

TableFlags: 0x00 (Standard)

4.3.2 RopGetAttachmentTable Response Buffer

0000:21 01 00 00 00 00

RopId: 0x21

OutputHandleIndex: 0x01

ReturnValue: 0x00000000

4.4 Insert HTML Embedded Image

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

4.4.1 RopCreateAttachment Request Buffer

0000: 23 00 00 01

RopId: 0x23

LogonId: 0x00

InputHandleIndex: 0x00

OutputHandleIndex: 0x01

4.4.2 RopCreateAttachment Response Buffer

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

RopId: 0x23

OutputHandleIndex: 0x01

ReturnValue: 0x00000000

AttachmentID: 0x00000000

4.4.3 Setting Properties

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

Property tag	Property name	Data
0x37050003	**PidTagAttachMethod** (section 2.2.2.9)	0x00000001

Property tag	Property name	Data
0x370B0003	**PidTagRenderingPosition** (section 2.2.2.16)	0xFFFFFFFF
0x7FFD0003	**PidTagAttachmentFlags** (section 2.2.2.23)	0x00000000
0x3001001F	**PidTagDisplayName** (section 2.2.2.4)	"image001.PNG"
0x3712001F	**PidTagAttachContentId** (section 2.2.2.26)	"image001.PNG@01C86E1C.F1954390"
0x370E001F	**PidTagAttachMimeTag** (section 2.2.2.26)	"image/PNG"
0x7FFA0003	**PidTagAttachmentLinkId** (section 2.2.2.22)	0x00000000
0x37140003	**PidTagAttachFlags** (section 2.2.2.18)	"0x00000004"
0x7FFE000b	**PidTagAttachmentHidden** (section 2.2.2.24)	"0x01"
0x3707001F	**PidTagAttachLongFilename** (section 2.2.2.10)	"image001.PNG"
0x3704001F	**PidTagAttachFilename** (section 2.2.2.11)	"image001.PNG"
0x3703001F	**PidTagAttachExtension** (section 2.2.2.12)	".PNG"

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

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

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

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

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

4.4.4 RopSaveChangesAttachment Request Buffer

0000: 25 00 01 00 0A

RopId: 0x25

LogonId: 0x00

ResponseHandleIndex: 0x01

InputHandleIndex: 0x00

SaveFlags: 0x0A (**KeepOpenReadWrite**)

4.4.5 RopSaveChangesAttachment Response Buffer

0000: 25 01 00 00 00 00

RopId: 0x25

ResponseHandleIndex: 0x01

ReturnValue: 0x00000000

4.4.6 Releasing Attachment Object

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

4.5 Attach Text File

The client first creates the Attachment object on the Message object 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 00 01 00-00 00

RopId: 0x23

OutputHandleIndex: 0x03

ReturnValue: 0x00000000

AttachmentID: 0x00000001

4.5.3 Setting Properties

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

Property tag	Property name	Data
0x37050003	**PidTagAttachMethod** (section 2.2.2.9)	"0x00000001"
0x370B0003	**PidTagRenderingPosition** (section 2.2.2.16)	"0xFFFFFFFF"
0x7FFD0003	**PidTagAttachmentFlags** (section 2.2.2.23)	"0x00000000"
0x3001001F	**PidTagDisplayName** (section 2.2.2.4)	"test.txt"
0x7FFA0003	**PidTagAttachmentLinkId** (section 2.2.2.22)	"0x00000000"
0x37140003	**PidTagAttachFlags** (section	"0x00000000"

77 / 94

Property tag	Property name	Data
	2.2.2.18)	
0x7FFE000B	**PidTagAttachmentHidden** (section 2.2.2.24)	"0x00"
0x3707001F	**PidTagAttachLongFilename** (section 2.2.2.10)	"test.txt"
0x3704001F	**PidTagAttachFilename** (section 2.2.2.11)	"test.txt"
0x3703001F	**PidTagAttachExtension** (section 2.2.2.12)	".txt"
0x30070040	**PidTagCreationTime** (section 2.2.2.3)	"2008/02/1222:28:34.636"
0x30080040	**PidTagLastModificationTime** (section 2.2.2.2)	"2008/02/1222:28:50.112"
0x37090102	**PidTagAttachRendering** (section 2.2.2.17)	3,512 bytes representing a Windows Metafile Format (WMF) file. For more information on WMF, see [MS-WMF].

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 2.2.9.6) 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 2.2.1.56.3), is the following:

```
<html>
<head>
<meta http-equiv=Content-Type content="text/html; charset=us-ascii">
</head>
<body lang=EN-US link=blue vlink=purple>
<p>This is a sample body text<o:p></o:p></p>
<p ><img width=174 height=152 id="Picture_x0020_2"
src="cid:image001.png@01C86E1C.F1954390"
alt="cid:image001.png@01C86E1C.F1954390"><o:p></o:p></p>
</div>
</body>
</html>
```

4.7 Adding Recipients

4.7.1 RopModifyRecipients Request Buffer

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

```
0010:39 1f 00 fe 39 03 00 71-3a 03 00 05 39 1f 00 f6
0020:5f 03 00 fd 5f 03 00 ff-5f 03 00 de 5f 03 00 df
0030:5f 02 01 f7 5f 01 00 00-00 00 00 01 27 01 51 06
0040:5a 00 55 73 65 72 32 00-75 00 73 00 65 00 72 00
0050:32 00 00 00 75 00 73 00-65 00 72 00 32 00 00 00
0060:0c 00 00 06 00 00 00 00-00 00 00 75 00 73 00 65
0070:00 72 00 32 00 00 00 75-00 73 00 65 00 72 00 32
0080:00 40 00 73 00 7a 00 66-00 6b 00 75 00 6b 00 2d
0090:00 64 00 6f 00 6d 00 2e-00 65 00 78 00 74 00 65
00a0:00 73 00 74 00 2e 00 6d-00 69 00 63 00 72 00 6f
00b0:00 73 00 6f 00 66 00 74-00 2e 00 63 00 6f 00 6d
00c0:00 00 00 00 00 00 00 00-00 00 40 75 00 73 00 65
00d0:00 72 00 32 00 00 00 01-00 00 00 00 00 00 00 00
00e0:00 00 00 00 00 00 7c-00 00 00 00 00 dc a7 40
00f0:c8 c0 42 10 1a b4 b9 08-00 2b 2f e1 82 01 00 00
0100:00 00 00 00 00 2f 6f 3d-46 69 72 73 74 20 4f 72
0110:67 61 6e 69 7a 61 74 69-6f 6e 2f 6f 75 3d 45 78
0120:63 68 61 6e 67 65 20 41-64 6d 69 6e 69 73 74 72
0130:61 74 69 76 65 20 47 72-6f 75 70 20 28 46 59 44
0140:49 42 4f 48 46 32 33 53-50 44 4c 54 29 2f 63 6e
0150:3d 52 65 63 69 70 69 65-6e 74 73 2f 63 6e 3d 75
0160:73 65 72 32 00
```

RopId: 0x0E

LogonId: 0x00

InputHandleIndex: 0x08

ColumnCount: 0x000C (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 2.888): 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 2.891): 0x5FDF0003

PidTagRecipientEntryId ([MS-OXPROPS] section 2.889): 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 00 01 01-00 00 00 00 f0 86 39

RopId: 0x0C

ResponseHandleIndex: 0x00

ReturnValue: 0x00000000

InputHandleIndex: 0x01

MessageId: 01 00 00 00 00 F0 86 39

4.9   Releasing Message Object

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

5 Security

5.1 Security Considerations for Implementers

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

5.2 Index of Security Parameters

None.

6 Appendix A: Product Behavior

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

- Microsoft Exchange Server 2003

- Microsoft Exchange Server 2007

- Microsoft Exchange Server 2010

- Microsoft Exchange Server 2013

- Microsoft Exchange Server 2016 Preview

- Microsoft Office Outlook 2003

- Microsoft Office Outlook 2007

- Microsoft Outlook 2010

- Microsoft Outlook 2013

- Microsoft Outlook 2016 Preview

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

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

<1> Section 2.2: Exchange 2007 changes properties **PidTagAccess** and **PidTagLastModificationTime**. Exchange 2010, Exchange 2013, and Exchange 2016 Preview 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, and Exchange 2016 Preview do not support the **PidTagObjectType** property

<3> Section 2.2.1.1: Exchange 2010, Exchange 2013, and Exchange 2016 Preview 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, and Exchange 2016 Preview.

<6> Section 2.2.1.56.2: Exchange 2003 and Exchange 2007 do not support the **PidTagNativeBody** property.

<7> Section 2.2.1.58.6: Exchange 2003, Exchange 2007, and the initial release version of Exchange 2010 do not support the **NeedsRescan** flag.

<8> Section 2.2.1.58.6: 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, and Exchange 2016 Preview 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, and Exchange 2016 Preview 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, and Exchange 2016 Preview return read/write messages when the user does not have write permissions.

<13> Section 2.2.3.3.1: Exchange 2010, Exchange 2013, and Exchange 2016 Preview 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: ex14 and ex15 return a zero value.

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

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

<18> Section 2.2.3.18: Exchange 2010, Exchange 2013, and Exchange 2016 Preview 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 and Exchange 2016 Preview do not initialize the following properties: **PidTagCreatorName** ([MS-OXPROPS] section 2.647), **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 2.873).

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

<23> Section 3.2.5.2: Exchange 2013 and Exchange 2016 Preview 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 0x000004FF.

<25> Section 3.2.5.3: Exchange 2010, Exchange 2013, and Exchange 2016 Preview 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.

<26> Section 3.2.5.3: Exchange 2010, Exchange 2013, and Exchange 2016 Preview return a GeneralFailure error if pending changes include changes to read-only properties **PidTagMessageSize**, **PidTagAccess**, **PidTagAccessLevel**, **PidTagObjectType**, **PidTagRecordKey**, **PidTagMessageStatus**, and **PidTagHasAttachments**.

<27> Section 3.2.5.3: Exchange 2010, Exchange 2013, and Exchange 2016 Preview return this value. Exchange 2007 returns Success.

<28> Section 3.2.5.10: Exchange 2003, Exchange 2007, Exchange 2010, Office Outlook 2003, Office Outlook 2007, and Outlook 2010 do not support this behavior. Microsoft Exchange Server 2010 Service Pack 1 (SP1), Exchange 2013, Exchange 2016 Preview, Microsoft Outlook 2010 Service Pack 1 (SP1), Outlook 2013, and Outlook 2016 Preview support this behavior.

<29> Section 3.2.5.10: Exchange 2003, Exchange 2007, Exchange 2010, Office Outlook 2003, Office Outlook 2007, and Outlook 2010 do not support this behavior. Exchange 2010 SP1, Exchange 2013, Exchange 2016 Preview, Outlook 2010 SP1, Outlook 2013, and Outlook 2016 Preview support this behavior.

<30> Section 3.2.5.13: Exchange 2013 and Exchange 2016 Preview set the **PidTagAttachSize** property (section 2.2.2.5) to 0x00000000.

<31> Section 3.2.5.13: Exchange 2013 and Exchange 2016 Preview set the **PidTagAccessLevel** property ([MS-OXCPRPT] section 2.2.1.2) to 0x00000000.

<32> Section 3.2.5.13: Exchange 2013 and Exchange 2016 Preview 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 New, Major, Minor, Editorial, or No change.

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

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

- A document revision that incorporates changes to interoperability requirements or functionality.

- The removal of a document from the documentation set.

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

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

The revision class **No change** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the technical content of the document is identical to the last released version.

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

- New content added.

- Content updated.

- Content removed.

- New product behavior note added.

- Product behavior note updated.

- Product behavior note removed.

- New protocol syntax added.

- Protocol syntax updated.

- Protocol syntax removed.

- New content added due to protocol revision.

- Content updated due to protocol revision.

- Content removed due to protocol revision.

- New protocol syntax added due to protocol revision.

- Protocol syntax updated due to protocol revision.

- Protocol syntax removed due to protocol revision.

- Obsolete document removed.

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

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

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.

- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
2 Messages	Updated product behavior notes for the "Messages" section to include behavior of Exchange 2016.	Y	Product behavior note updated.
3 Protocol Details	Updated product behavior notes for the "Protocol Details" section to include behavior of Exchange 2016 and Outlook 2016.	Y	Product behavior note updated.
3.2.5.2 Receiving a RopCreateMessage ROP Request	Specified that Exchange 2007 and Exchange 2010 return error code ecNoCreateRight with value 0x000004FF.	Y	New product behavior note added.
6 Appendix A: Product Behavior	Added Exchange 2016 and Outlook 2016 to the list of applicable products.	Y	Content update.

8 Index

A

Abstract data model
  client 56
  server 63
Abstract data model types - client
  global 56
  per mailbox 56
  per Message object 56
Abstract data model types - server
  global 63
  per mailbox 63
  per Message object 63
Add recipients example
  RopModifyRecipients request buffer 79
  RopModifyRecipients response buffer 81
Applicability 18
Attach text file example
  overview 77
  releasing Attachment object 79
  RopCreateAttachment request buffer 77
  RopCreateAttachment response buffer 77
  RopSaveChangesAttachment request buffer 78
  RopSaveChangesAttachment response buffer 78
  setting properties 77
Attachment object properties
  general properties 37
  MIME properties 41
  PidTagAttachAdditionalInformation property 40
  PidTagAttachDataBinary property 38
  PidTagAttachDataObject property 38
  PidTagAttachEncoding property 40
  PidTagAttachExtension property 39
  PidTagAttachFilename property 38
  PidTagAttachFlags property 40
  PidTagAttachLongFilename property 38
  PidTagAttachLongPathname property 39
  PidTagAttachmentFlags property 41
  PidTagAttachmentHidden property 41
  PidTagAttachmentLinkId property 41
  PidTagAttachMethod property 38
  PidTagAttachNumber property 37
  PidTagAttachPathname property 39
  PidTagAttachRendering property 40
  PidTagAttachSize property 37
  PidTagAttachTag property 39
  PidTagAttachTransportName property 40
  PidTagCreationTime property 37
  PidTagDisplayName property 37
  PidTagLastModificationTime property 37
  PidTagRenderingPosition property 39
  PidTagTextAttachmentCharset property 41

B

Body Message object properties 31

C

Capability negotiation 18

Change tracking 87
Client
  abstract data model 56
  initialization 57
  other local events 62
  timer events 62
  timers 56
Client - abstract data model types
  global 56
  per mailbox 56
  per Message object 56
Client - higher-layer triggered events
  accessing the attachments table 58
  adding, deleting, or modifying a recipient 57
  creating a Message object 57
  creating an attachment 58
  creating an embedded message 58
  deleting an attachment 58
  getting message status 57
  linking a Contact object 59
  opening a Message object 57
  opening an attachment 58
  opening an Embedded Message object 58
  reading recipients 57
  reload Message object header info 57
  removing all recipients 57
  saving an embedded message 58
  saving changes on a Message object 57
  saving changes on an Attachment object 58
  setting Attachment object content 58
  setting Message object read status 58
  setting message status 57
Client - message processing
  sending a RopCreateAttachment ROP request 61
  sending a RopCreateMessage ROP request 59
  sending a RopGetAttachmentTable ROP request 62
  sending a RopGetMessageStatus ROP request 61
  sending a RopGetPropertiesSpecific ROP request 62
  sending a RopModifyRecipients ROP request 60
  sending a RopOpenAttachment ROP request 61
  sending a RopOpenEmbeddedMessage ROP request 62
  sending a RopOpenMessage ROP request 59
  sending a RopReadRecipients ROP request 60
  sending a RopRemoveAllRecipients ROP request 60
  sending a RopSaveChangesAttachment ROP request 62
  sending a RopSaveChangesMessage ROP request 59
  sending a RopSetMessageStatus ROP request 60
  sending a RopSetProperties ROP request 61
  sending a RopSetReadFlags ROP request 61
Client - sequencing rules
  sending a RopCreateAttachment ROP request 61
  sending a RopCreateMessage ROP request 59
  sending a RopGetAttachmentTable ROP request 62
  sending a RopGetMessageStatus ROP request 61
  sending a RopGetPropertiesSpecific ROP request 62